

BAB IV

ANALISIS SISTEM

A. Program Utama

1. Tampilan program

Untuk mempermudah pengguna menggunakan perangkat lunak maka disusun tampilan antarmuka yang sederhana dan mudah digunakan. Tampilan program utama disusun dengan file *reminder_list.xml*.

Secara umum terdapat dua tampilan antarmuka yaitu program utama sebagai halaman depan sekaligus menampilkan daftar jadwal pengingat yang sedang berjalan. Yang kedua adalah tampilan subprogram yang berfungsi untuk menampilkan form untuk menambahkan jadwal baru.

a. Tampilan program utama

Program utama dibuat untuk menghubungkan antara pengguna dengan subprogram yang akan mengirimkan SMS pengingat minum obat otomatis. Program utama terdiri atas tampilan list yang akan menampilkan daftar jadwal pengingat yang sedang berjalan.



Gambar 7. Tampilan antarmuka program utama

Kelas *ListActivity* diimplementasikan pada tampilan program utama. Kelas ini yang akan melakukan display daftar jadwal pengingat otomatis yang sedang berjalan. Daftar didapat dari data yang ada di *SQLite* database Android dengan bantuan kelas *database adapter*. Kumpulan data ini disebut *query string*.

b. Tampilan Menu

Untuk menghubungkan antara program utama dengan subprogram dibuat menu yang apabila ditekan akan memunculkan tampilan subprogram. Dalam hal ini penulis menggunakan fitur *Options Menu* yang bisa diakses dengan menekan tombol menu (Felker, 2010).

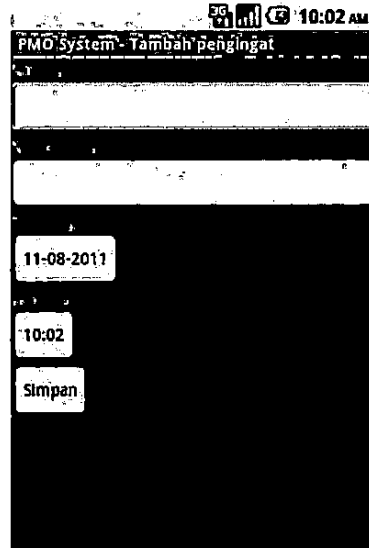


Gambar 8. Tampilan antarmuka program utama setelah menekan tombol menu

c. Tampilan Subprogram

Pada subprogram akan menampilkan form untuk mengisi data pasien. Selain itu terdapat tombol untuk mengatur tanggal dan waktu penjadwalan. Setelah semua kebutuhan data terpenuhi, kemudian

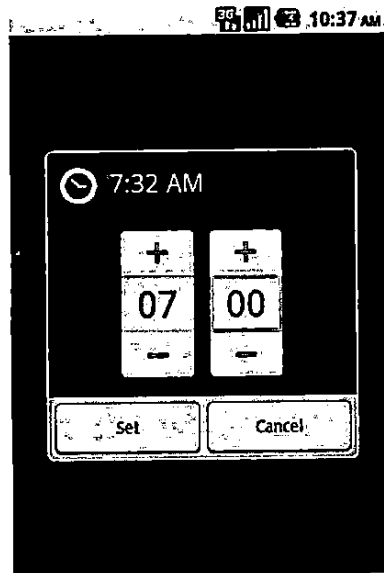
menekan tombol “Simpan” untuk melakukan registrasi jadwal baru.



Gambar 9. Tampilan Subprogram Tambah Pengingat

Pada subprogram, pengguna perlu memasukkan beberapa data sebelum dapat menjalankan pengingat. Ada 4 data input yang harus dilakukan, yang pertama memasukkan nama pasien, data ini diperlukan sebagai identitas jadwal pengingat. Kedua memasukkan nomor tujuan pengiriman SMS pengingat, dalam hal ini nomor telepon genggam pasien. Nomor ini yang akan digunakan sebagai tujuan pengiriman SMS pengingat.

Data ketiga adalah tanggal mulai pengiriman SMS pengingat. Untuk mengatur tanggal perlu menekan tombol “Tanggal pengingat” yang akan memunculkan dialog untuk memilih tanggal. Kemudian tekan tombol “Set” untuk meregistrasi input tanggal. Format tanggal menggunakan format “tanggal-bulan-tahun” (DD-MM-YYYY).



Gambar 11. Dialog untuk mengatur waktu

Setelah semua data dimasukkan kemudian pengguna menekan tombol “Simpan” untuk meregistrasi semua data ke dalam database. Proses registrasi akan menyimpan data dalam satu baris database. Selanjutnya satu baris tersebut akan ditampilkan di halaman program utama.



Gambar 12. Tampilan program utama setelah registrasi

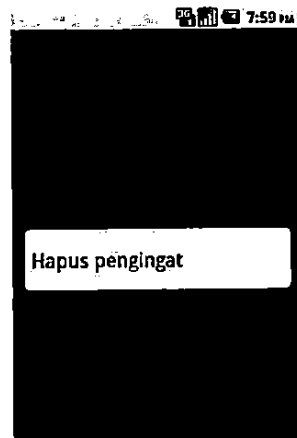
2. *Background Service*

Dalam perangkat lunak pengingat minum obat melibatkan proses. Beberapa proses diantaranya proses menyimpan data yang sudah diregister, menjalankan penjadwalan pengiriman SMS, dan pengiriman SMS itu sendiri.

a. Database

Sebagai media penyimpanan data penulis menggunakan fitur *SQLite database* yang sudah tertanam di dalam telepon genggam dengan platform *Android*. Dalam perangkat lunak ini database ditangani oleh kelas *RemindersDbAdapter*.

Pada dasarnya kelas ini menangani fungsi dasar database yaitu *CRUD (Create, Read, Update, Delete)*. Fungsi-fungsi ini akan dilakukan oleh activity yang berbeda. Fungsi *Read* dan *Delete* akan diimplementasikan pada program utama yang akan menampilkan daftar jadwal pengingat. Ketika salah satu jadwal ditekan lebih dari 2 detik, akan muncul opsi untuk menghapus jadwal.



Gambar 13. Opsi hapus data

Fungsi *Create*, *Read*, dan *Update* diimplementasikan pada kelas *ReminderEditActivity*. Kelas ini yang mengontrol subprogram tambah jadwal pengingat yang selanjutnya dapat melayani fasilitas penambahan jadwal pengingat baru, membaca data jadwal kemudian melakukan modifikasi di dalamnya.

id	nama	Nomor	reminder_date_time
1	Pak Sukijo	085729671587	15-07-2011 07:00
2	Bu Wagirah	0817279126	20-07-2011 06:00

Tabel 1. Ilustrasi tabel database

b. *Alarm Manager*

Untuk dapat melakukan suatu aktifitas yang terjadwal, maka perlu mengimplementasikan fitur *Alarm Manager*. *Alarm Manager* dijalankan dalam bentuk service sehingga dapat berjalan walaupun perangkat lunak sudah ditutup. Ada empat kelas yang akan bertanggung jawab terhadap jalannya jadwal, yaitu kelas *ReminderManager*, *OnAlarmReceiver*, *WakeReminderIntentService*, yang terakhir kelas *ReminderService*. Pertama kelas *ReminderManager* yang akan mengatur mulainya jadwal dengan fitur *Alarm Manager*.

Selanjutnya untuk menentukan aktifitas tertentu ketika mencapai waktu yang terjadwal akan ditangani oleh kelas *OnAlarmReceiver*. Kelas ini menjadi *entry-point* aktifnya Alarm Manager yang sudah diatur oleh kelas *ReminderManager*. Kemudian kelas selanjutnya yaitu *WakeReminderIntentService* akan membuat perangkat lunak tetap berjalan walaupun sudah ditutup. Selain itu juga

melakukan sinkronisasi dengan waktu telepon genggam. Kelas ini juga akan berhubungan dengan kelas *ReminderService* untuk mengaktifkan kelas tersebut.

Pada saat tertentu telepon genggam bisa mati ketika kehabisan tenaga batere, maka supaya waktu jadwal berjalan terus perlu diimplementasikan kelas *onBootReceiver*. Kelas ini menangani masalah telepon genggam ketika harus mati. Ketika telepon genggam hidup kembali, kelas ini akan menjalankan kembali waktu jadwal dan juga menyesuaikan dengan waktu telepon genggam ketika hidup kembali.

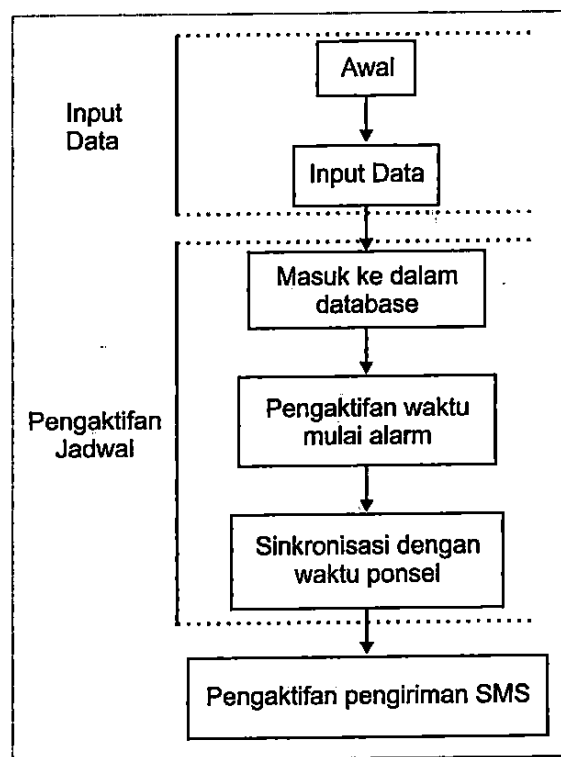
c. *ReminderService*

Setelah serangkaian proses penyesuaian waktu dan pengaktifan alarm, maka perangkat lunak perlu menentukan aktifitas apa untuk mengingatkan pasien. *SmsManager* merupakan fitur Android untuk berkomunikasi dengan fasilitas SMS. Fitur ini diimplementasikan pada kelas *ReminderService*. Untuk mengaktifkan pengiriman SMS, kelas ini membutuhkan informasi nomor tujuan pengiriman dan isi pesan. Informasi ini ditarik dari database dengan kelas *Cursor* dalam bentuk string.

2. **Fungsionalitas dan Integritas Sistem**

Pengujian integritas sistem dilakukan untuk menguji keutuhan dan keterkaitan antara program yang telah dibuat. Hal ini dilakukan karena di dalam program tersebut terdapat kelas-kelas yang saling berkaitan yang

menghasilkan suksesnya pengiriman SMS secara otomatis. Program diuji pada dua lingkungan yang berbeda, pada emulator (*ad hoc*) dan pada telepon genggam. Hubungan antara program dengan *background service* dapat dilihat di diagram 15. Hasil pengujian fungsional program adalah sebagai berikut.



Gambar 14. Diagram keseluruhan program PMO System v1.0

Untuk mengecek apakah tombol dan fungsi-fungsi perangkat lunak berjalan dengan baik maka diuji beberapa percobaan berikut. Hasilnya disajikan oleh tabel 2 berikut.

No.	Proses	Algoritma program	
		Berhasil	Gagal
1.	Membuka subprogram input tambah pengingat baru	√	
2.	Registrasi jadwal	√	

3.	Menghapus jadwal	√	
4.	Mengirimkan SMS sesuai jadwal	√	
5.	Mengirimkan SMS secara berkala	Tidak dapat diuji	

Tabel 2. Uji fungsional dan integritas program pada emulator

No.	Proses	Algoritma program	
		Berhasil	Gagal
1.	Membuka subprogram input tambah pengingat baru	√	
2.	Registrasi jadwal	√	
3.	Menghapus jadwal	√	
4.	Mengirimkan SMS sesuai jadwal	√	
5.	Mengirimkan SMS secara berkala	√	

Tabel 3. Uji fungsional dan integritas program pada telepon genggam

Syarat tiap poin di atas dianggap berhasil apabila dapat berjalan baik pada emulator maupun pada telepon genggam. Pada pengujian membuka subprogram berfungsi dengan baik. Proses pengisian data termasuk pengaturan tanggal dan waktu dapat berjalan. Registrasi data berjalan dengan baik dengan bukti tertera nama pasien pada halaman program utama.