

APLIKASI *SCHEDULER* DAN *REMINDER* BERBASIS WEB UNTUK KELOMPOK PROFESIONAL (STUDI KASUS DI PT. SARANA PERMATA CONTAINER SEMARANG)

Novrido Charibaldi, Denyu Budi Wijayanto
Jurusan Teknik Informatika UPN "Veteran" Yogyakarta
novrido_ch@walla.com

Abstract

Application of Scheduler and of Reminder For Professional Group can look for some right time alternatives to be performed a meeting on H which have been determined by group leader. This application will deliver message of SMS for notification will be had a meeting after group leader determine meeting schedule and as schedule reminder of each group member where the delivery of can be arranged the time of before the schedule fulfilled. The methodology of software engineering who was used to develop this application was GRAPPLE. The segment of GRAPPLE (Gidelines for Rappid Application Engineering) methodology are Requirement Gathering, Analysis, Design, Development, and Deployment. Programming language who was used in development segmen was Java and JSP (Java Server Pages)

Keyword : Application, Scheduler, Reminder, SMS, GRAPPLE

Aplikasi *Scheduler* dan *Reminder* Untuk Kelompok Profesional dapat mencari beberapa alternatif waktu yang tepat untuk diadakan rapat pada hari H yang telah ditentukan oleh pimpinan kelompok. Aplikasi ini akan mengirimkan pesan SMS untuk pemberitahuan akan diadakan rapat setelah pimpinan kelompok menentukan jadwal rapat dan sebagai *reminder* jadwal masing-masing anggota kelompok dimana pengirimannya dapat diatur waktunya sebelum jadwal tersebut terpenuhi. Metodologi rekayasa perangkat lunak yang digunakan untuk membangun aplikasi ini adalah GRAPPLE yang meliputi *Requirement Gathering, Analysis, Design, Development, dan Deployment*. Bahasa pemrograman yang digunakan pada segmen *Development* adalah Java dan JSP (*Java Server Pages*).

Kata Kunci : Aplikasi, Scheduler, Reminder, SMS, GRAPPLE.

1. Pendahuluan

Penentuan jadwal rapat menjadi suatu permasalahan yang sulit apabila anggota rapat memiliki jadwal yang sangat padat. Sehingga penyelesaiannya menempatkan jadwal rapat diluar jam kerja dengan demikian akan mengurangi efektifitas dan efisiensi kerja. Untuk itu akan dibuat suatu sistem aplikasi *scheduler* dan *reminder* berbasis web untuk kelompok profesional.

Tujuan penelitian ini adalah menghasilkan aplikasi *scheduler* dan *reminder* berbasis web yang mampu memberikan alternatif waktu untuk diadakan rapat serta memberitahukan jadwal rapat serta *reminder* melalui SMS.

Dengan adanya aplikasi ini diharapkan dapat membantu pimpinan kelompok untuk menentukan jadwal rapat dengan cepat serta akurat.

2. Tinjauan Pustaka

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategis dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. (Jogiyanto, 2000). Jaringan GSM merupakan suatu *Public Switched Telephone Network* (PSTN) yang tersusun atas beberapa komponen pendukung (Mulyanta, 2003). Aplikasi SMS pada ponsel menyediakan layanan untuk mengirim dan menerima pesan pendek berupa huruf dan angka. SMS hanya terbatas pada pengiriman dan penerimaan data berupa teks dengan panjang pesan antara 120-160 huruf (Sutedjo, 2003).

Metode dalam rekayasa perangkat lunak ini menggunakan *Guidelines for Rappid Application Engineering* (GRAPPLE) yang terdiri dari lima segmen yaitu *Requirement Gathering, Analysis, Design, Development, dan Deployment*.

Unified Modeling Language

Unified Modelling Language (UML) merupakan sistem arsitektur yang bekerja dalam OOAD dengan suatu bahasa yang konsisten untuk menentukan visualisasi, mengkonstruksi dan mendokumentasikan *artifact* yang terdapat dalam sistem perangkat lunak (Suhendar dkk, 2003).

Java

Java adalah sebuah bahasa pemrograman yang juga merupakan suatu *platform*, yang dikeluarkan oleh *Sun Microsystem*. Salah satu teknologi java adalah "*write once run everywhere*" yang menawarkan kehandalan *java* untuk dapat berjalan secara lintas sistem operasi. *Java Virtual Machine* (JVM) yang dapat menjembatani antara program dengan keunikan serta kompleksitas sistem operasi dimana program itu berjalan.

Java Server Pages

JSP (*Java Server Pages*) merupakan teknologi yang didasarkan pada bahasa Java, yang dapat digunakan untuk membentuk halaman-halaman Web yang bersifat dinamis (Kadir, A, 2004). Teknologi ini dikembangkan oleh *Sun Microsystems*.

3. Analisis dan Perancangan Sistem

Dalam analisis dan perancangan ini akan dibahas tiga segmen dalam metodologi GRAPPLE (*Guidelines for Rapid APPLication Engineering*) yaitu segmen perencanaan kebutuhan, analisis dan perancangan. Setiap segmen terdiri dari beberapa aksi dan setiap aksi menghasilkan produk berupa diagram UML.

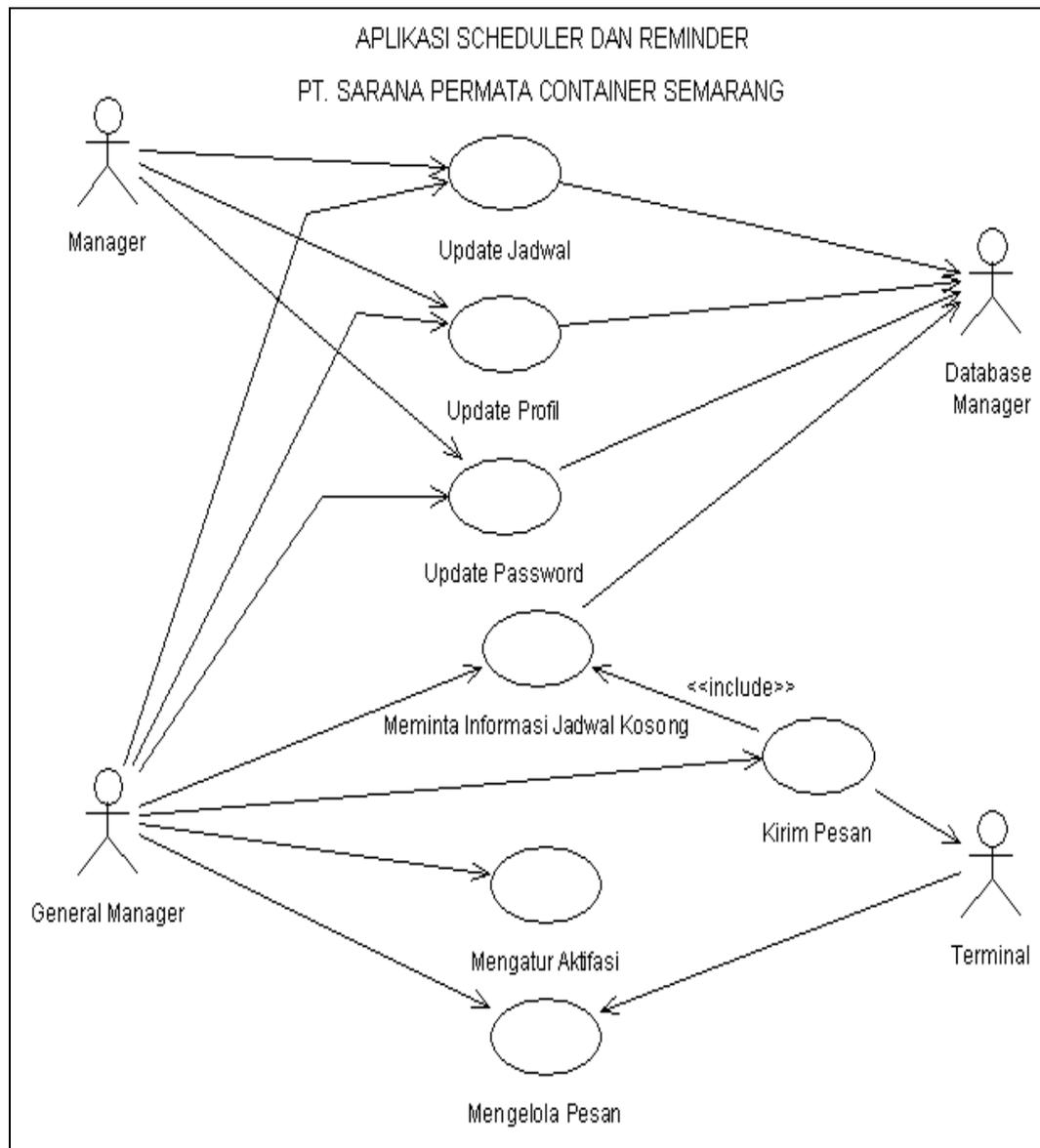
Perencanaan Kebutuhan

Perencanaan kebutuhan digunakan mengetahui kebutuhan pengguna terhadap sistem yang baru. Hal ini disebabkan apabila kita tidak dapat memahami kebutuhan dari pengguna, maka kita tidak dapat membuat suatu sistem dengan benar. Adapun kebutuhan-kebutuhan tersebut adalah sebagai berikut:

1. Anggota kelompok membutuhkan suatu sistem yang dapat meningkatkan efisiensi dan efektifitas kerja dan mengingatkan adanya jadwal rapat dan jadwal lain yang harus dilaksanakan.
2. Pimpinan kelompok menginginkan suatu sistem yang mampu mencari alternative waktu yang dapat digunakan untuk jadwal rapat dan memberitahukan kepada anggota kelompok apabila akan diadakan rapat.

Identifikasi Kebutuhan Sistem

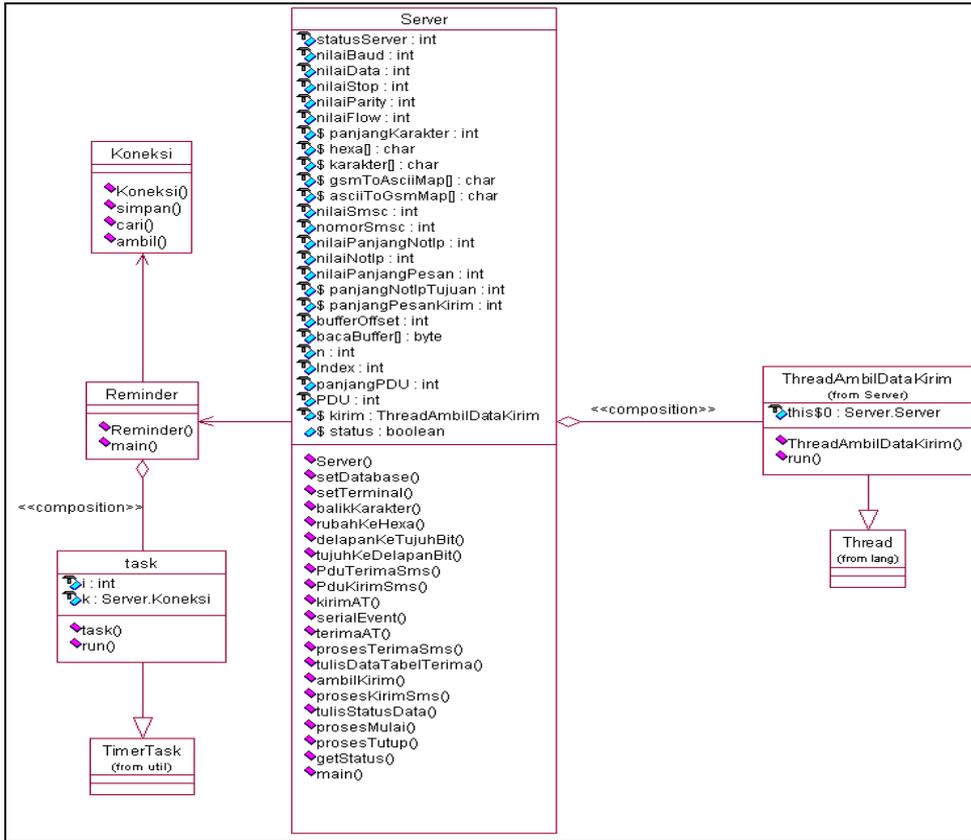
Diagram *Use Case* pada gambar 1 menjelaskan tentang sistem aplikasi *scheduler* dan *reminder* serta aktor yang berinteraksi dengan sistem. Dalam diagram *use case* terdapat empat aktor yang akan berinteraksi dengan sistem antara lain *General Manager, Manager, Terminal, Database Manager*. Dimana aktor ini akan berinteraksi dengan tujuh *use case*. Dari interaksi tersebut terlihat bahwa aktor *General Manager* yang bertindak sebagai *admin* memiliki akses lebih luas dari pada aktor *Manager* yang aksesnya terbatas. Terdapat hubungan *stereotype* bertipe *include* antara *use case* kirim pesan dengan *use case* meminta informasi jadwal kosong.



Gambar 1. Diagram Use Case

Memperinci Diagram Kelas

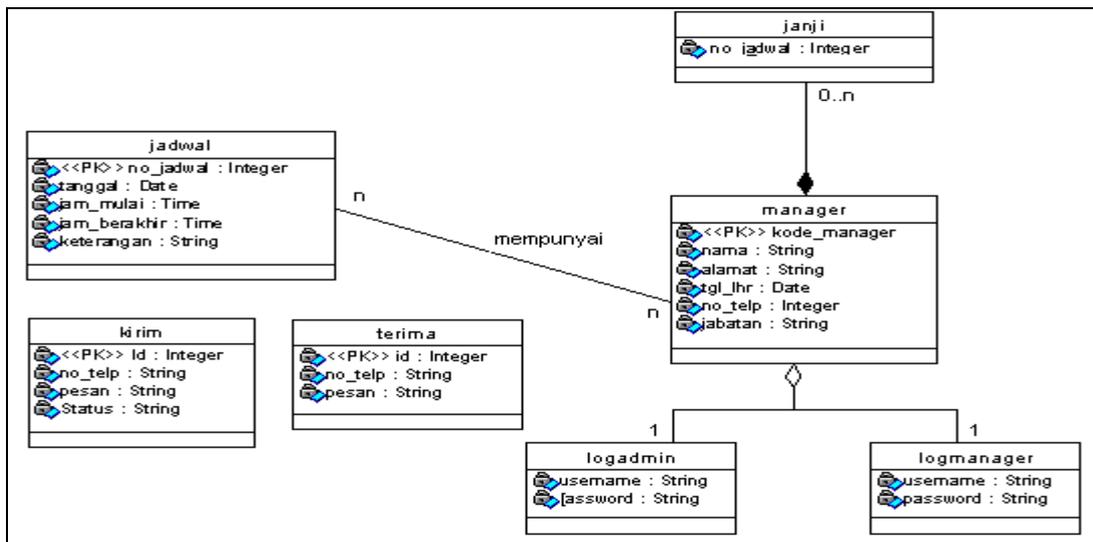
Diagram kelas menampilkan hubungan antar kelas. Kelas-kelas pada aplikasi ini yaitu *Server*, *Koneksi*, *Reminder*, *task*, *TimerTask*, *Thread*, *ThreadAmbilDataKirim*. Terdapat tiga relasi yang menghubungkan kelas-kelas tersebut yaitu *uni-directional association*, *composite*, dan *generalization*. *Uni-directional association* maksudnya adalah akhir dari arah panah mengindikasikan yang menerima komunikasi. *Composite* maksudnya *agregasi* yang kuat dimana bagian dari obyek tergantung pada keseluruhan obyek. *Generalization* menunjukkan hubungan antar elemen yang lebih umum ke elemen yang lebih khusus atau spesifik. Dengan *generalization*, kelas yang lebih spesifik (*subclass*) akan menurunkan atribut dan operasi dari kelas yang lebih umum (*superclass*).



Gambar 2. Diagram kelas

Perancangan Database

Perancangan *database* pada aplikasi *scheduler* dan *reminder* adalah obyek *manager* dan obyek janji menunjukkan relasi *composition*. Sehingga setiap janji secara fisik akan melekat pada satu obyek *manager*. Asosiasi yang digunakan adalah mempunyai antara obyek *manager* dan obyek jadwal. Logadmin dan logmanager ke *manager* adalah bentuk *agregasi*. Setiap obyek pada logadmin atau logmanager secara logika terkandung pada satu obyek *manager*.



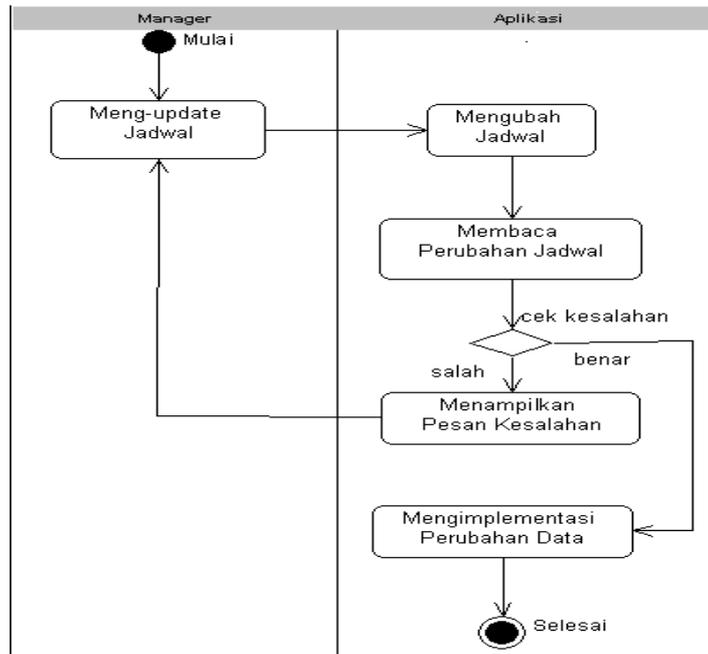
Gambar 3. Koneksi

Mengembangkan dan Memperinci Diagram Obyek

Diagram *Activity* merupakan produk dari aksi mengembangkan dan memperinci diagram obyek dalam tahap perancangan. Diagram *activity* dapat dihasilkan dari setiap *use case* yang ada. Dalam penelitian ini terdapat lima *use case* yang terdapat dalam sistem, dengan demikian dalam penelitian ini juga akan dibentuk tujuh diagram *activity* yang bersesuaian dengan *use case* yang ada.

3.5.1. Diagram *Activity Update Jadwal*

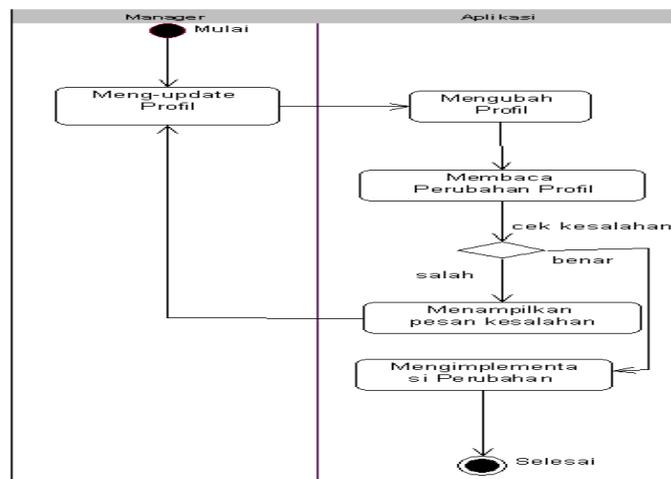
Pada diagram *Activity* ini memperlihatkan bahwa *manager* melakukan peng-update-an data terhadap data jadwal yang masing-masing *manager* miliki.



Gambar 4.4. Diagram *Activity Update Jadwal*

3.5.2. Diagram *Activity Update Profil*

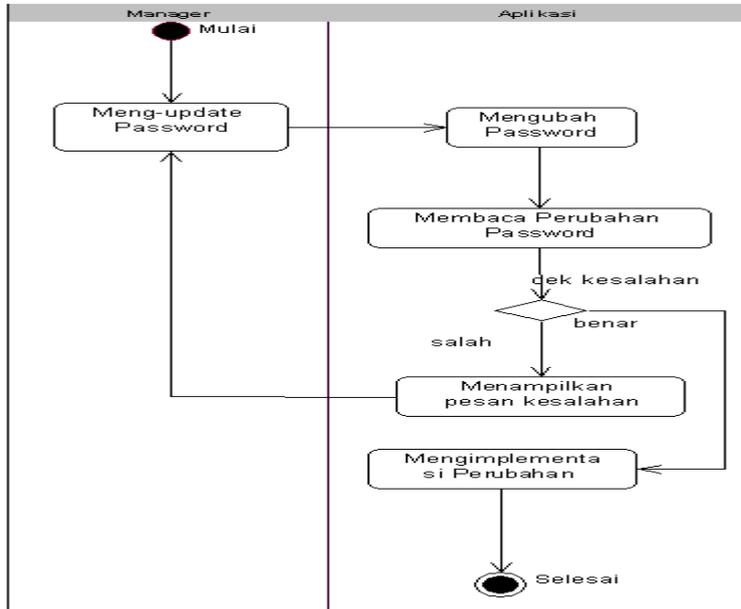
Pada diagram *Activity* ini memperlihatkan bahwa *manager* melakukan peng-update-an data terhadap data profil yang masing-masing *manager* miliki.



Gambar 5. Diagram *Activity Update Profil*

3.5.3. Diagram Activity Update Password

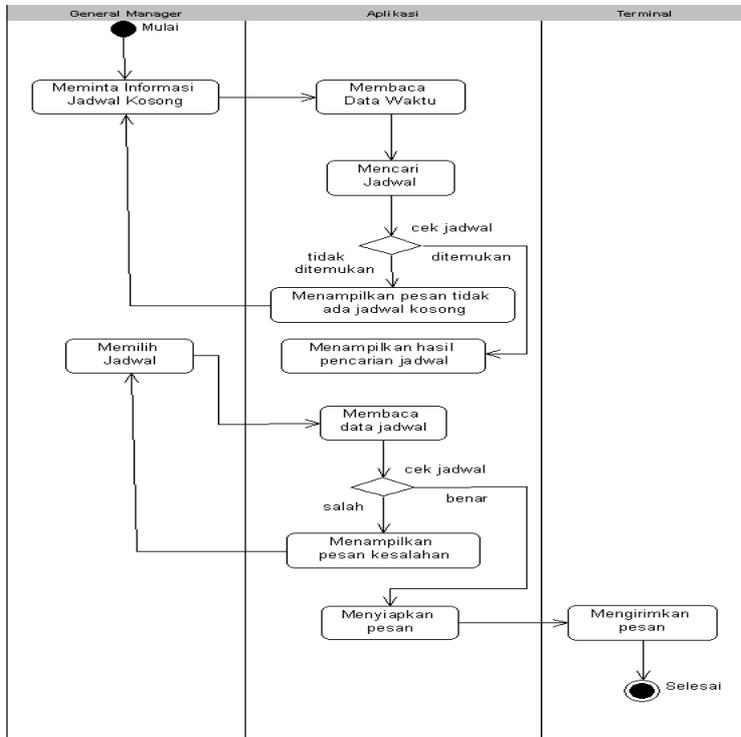
Pada diagram *Activity* ini memperlihatkan bahwa *manager* melakukan peng-update-an data terhadap data *password* yang masing-masing *manager* miliki.



Gambar 4.6. Diagram Activity Update Password

5.3.4. Diagram Activity Meminta Informasi Jadwal Kosong

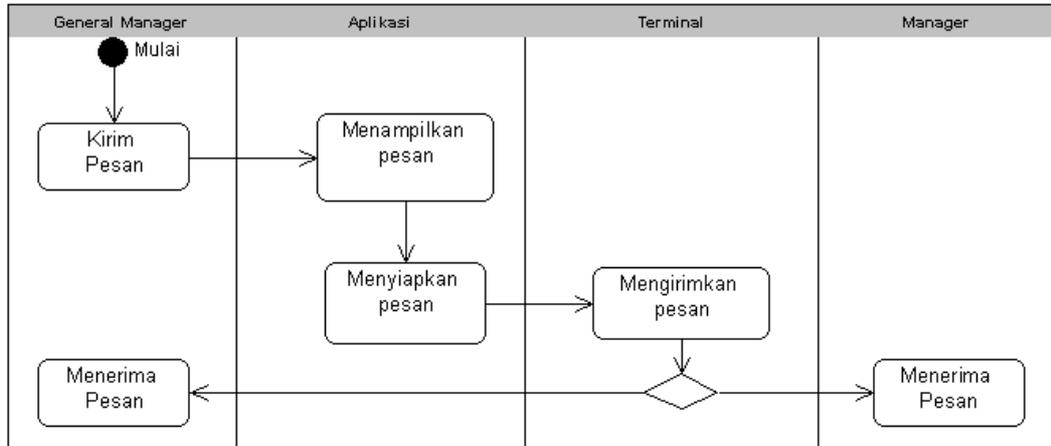
Pada diagram *Activity* ini memperlihatkan bahwa *general manager* meminta informasi jadwal kosong kepada sistem untuk dapat menentukan jadwal rapat.



Gambar 7. Diagram Activity Meminta Informasi Jadwal Kosong

3.5.5. Diagram Activity Kirim Pesan

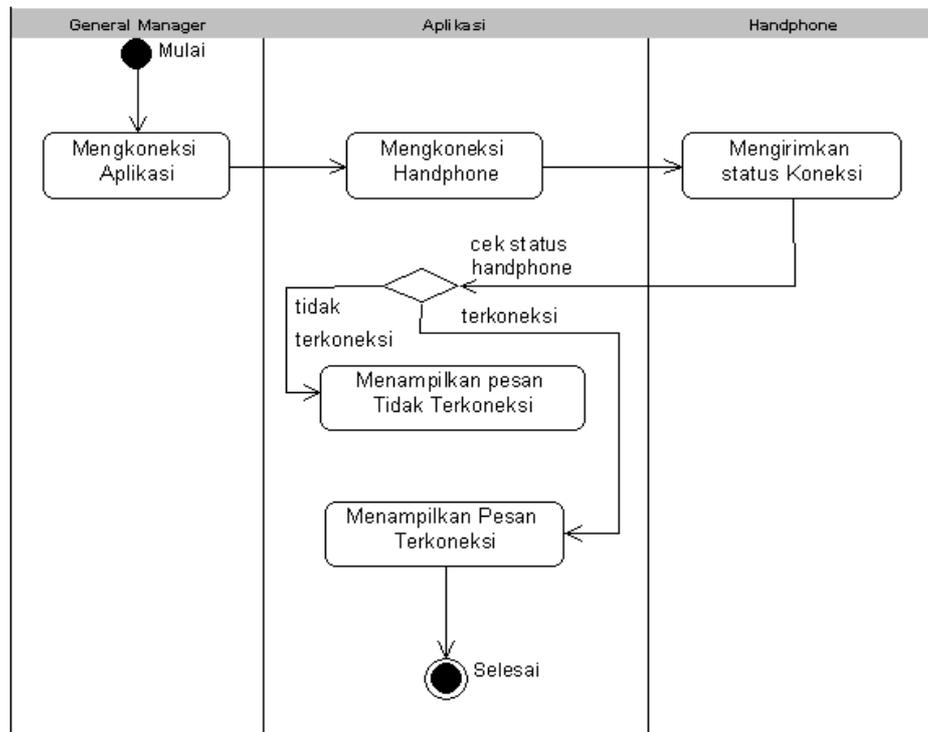
Pada diagram *Activity* ini memperlihatkan bahwa *general manager* melakukan pengiriman pesan SMS kepada *manager* dimana dalam aktivitas ini melibatkan terminal dalam proses pengiriman pesan SMS.



Gambar 8. Diagram Activity Kirim Pesan

3.5.6. Diagram Activity Mengatur Aktifasi

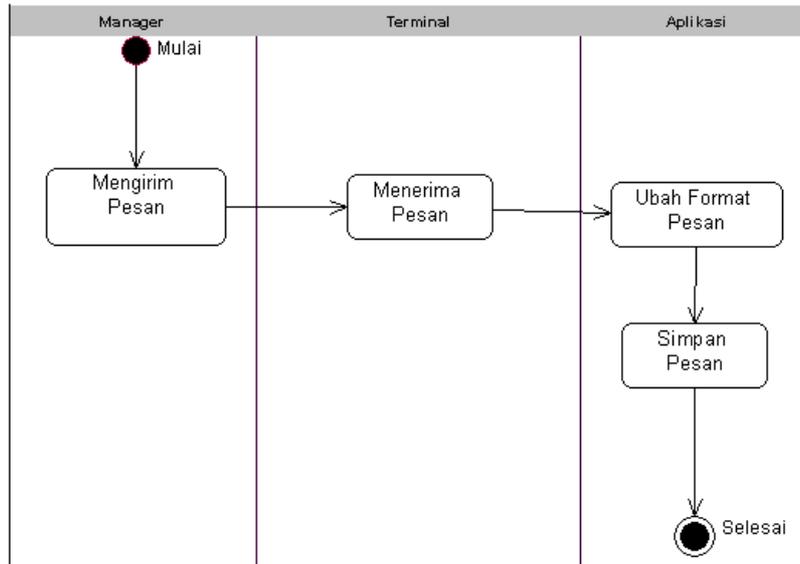
Pada diagram *Activity* ini memperlihatkan bahwa *general manager* melakukan aktifasi dari *Server*.



Gambar 9. Diagram Activity Mengatur Aktifasi

3.5.7. Diagram Activity Mengelola Pesan

Pada diagram *Activity* ini memperlihatkan bahwa sistem akan mengelola pesan yang dikirim oleh *manager* untuk disimpan dalam *database*. Dimana dalam aktivitas ini melibatkan terminal sebagai penerima pesan.

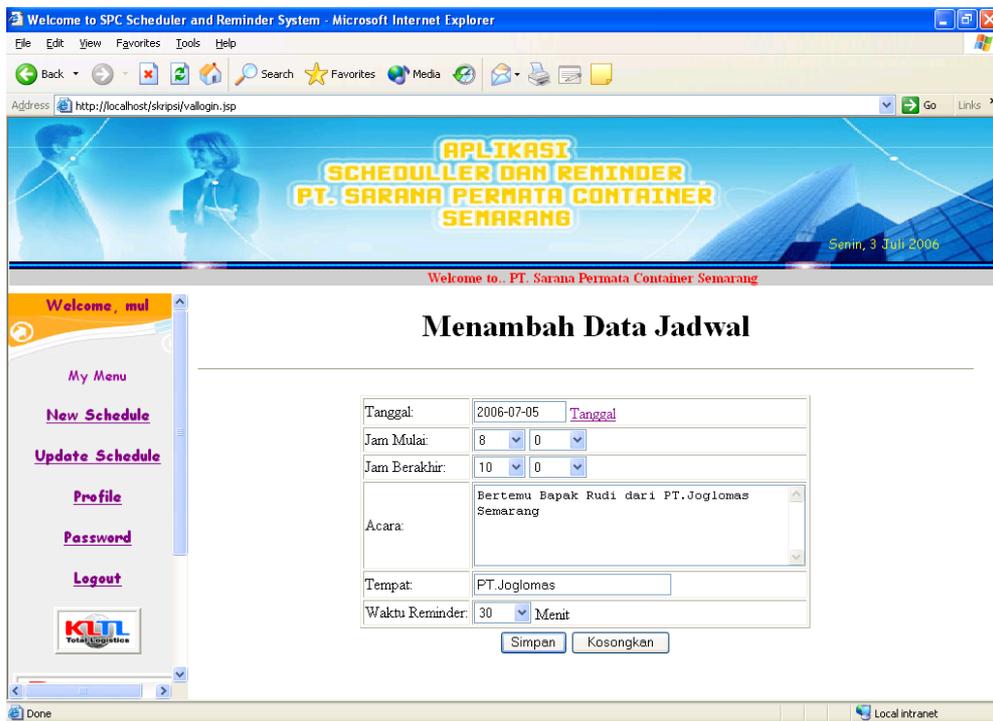


Gambar 10. Diagram Activity Mengelola Pesan

4. Hasil dan Pembahasan

Tampilan Antarmuka *Input Jadwal User*

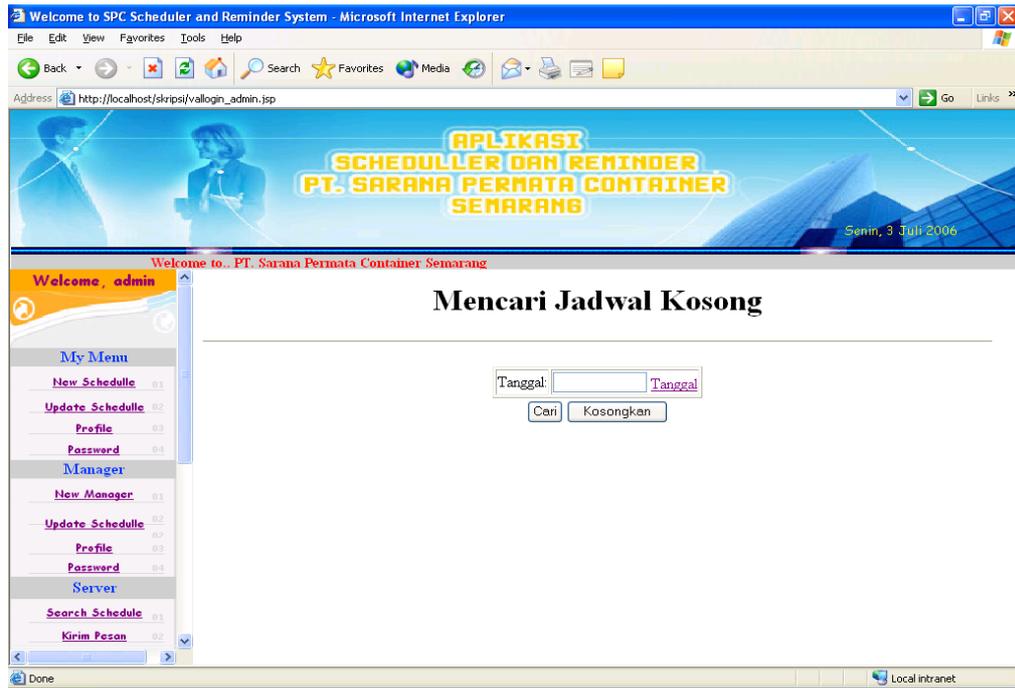
Pada antarmuka *input* jadwal *user* ditampilkan halaman dimana *user* dapat melakukan *input* jadwal. Tampilan antarmuka *input* jadwal *user* dapat dilihat pada gambar 11 berikut:



Gambar 11. Antarmuka *Input* Jadwal *User*

Tampilan Antarmuka Cari Waktu Kosong

Pada antarmuka cari waktu kosong ditampilkan halaman yang digunakan oleh *admin* untuk memasukkan tanggal yang akan dicari waktu kosongnya. Tampilan antarmuka cari waktu kosong dapat dilihat pada gambar 12 berikut:



Gambar 12. Antarmuka Cari Waktu Kosong

Tampilan Antarmuka Hasil Pencarian Waktu Kosong

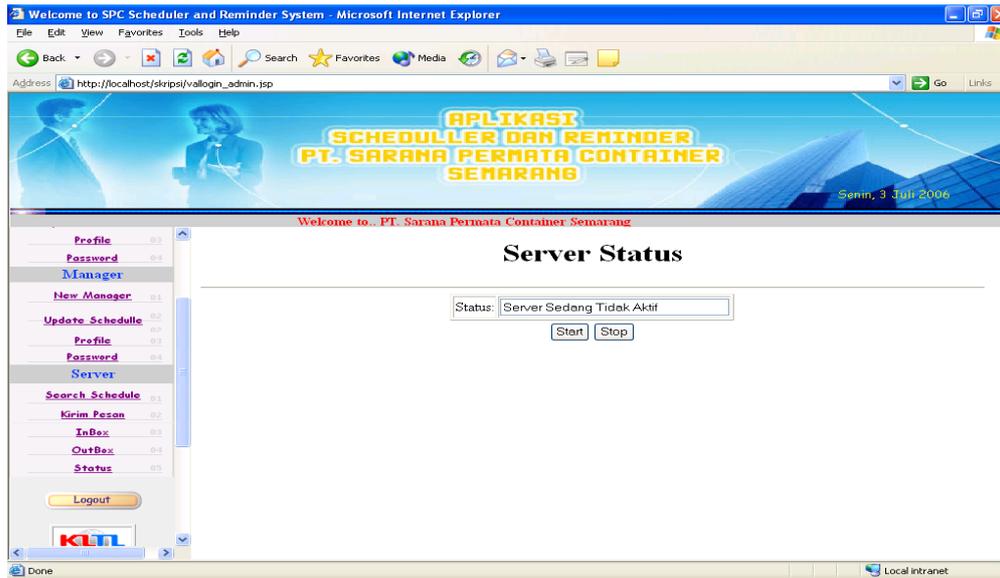
Pada antarmuka hasil pencarian waktu kosong ditampilkan halaman yang digunakan oleh *admin* untuk melihat waktu kosong dalam satu hari sesuai tanggal yang di-*inputkan* dan melakukan *input* jadwal rapat setelah melihat alternatif waktu yang ada. Tampilan antarmuka hasil pencarian waktu kosong dapat dilihat pada gambar 5.16 berikut:



Gambar 5.16. Antarmuka Hasil Pencarian Waktu Kosong

Tampilan Antarmuka Status Server

Pada antarmuka status server ditampilkan halaman yang digunakan oleh *admin* untuk melihat serta merubah status dari server. Tampilan antarmuka status server dapat dilihat pada gambar 5.20 berikut:



Gambar 5.20. Antarmuka Status Server

5. Kesimpulan

Telah berhasil dibangun sebuah aplikasi *scheduler* dan *reminder* berbasis web untuk kelompok profesional, dengan kemampuan (a). mencari beberapa alternatif waktu yang tepat untuk diadakan rapat pada hari yang telah ditentukan oleh pimpinan kelompok. (b). Pengiriman pesan SMS ini digunakan untuk pemberitahuan akan diadakannya jadwal rapat dan digunakan untuk reminder jadwal dari masing-masing *manager* dimana waktu pengirimannya disesuaikan dengan waktu dari jadwal tersebut.

6. Daftar Pustaka

- Jogiyanto, Hartono, 2000, *Pengenalan Komputer*, Penerbit Andi, Yogyakarta.
- Hermanan, Benny, 2004, *Menguasai Java 2 dan Object Oriented Programming*, Penerbit Andi, Yogyakarta
- Kadir, Abdul, 2004, *Dasar Pemrograman Web Dinamis dengan JSP*, Penerbit Andi, Yogyakarta.
- Mehrota, Asha, 1997, *GSM System Engineering*, Artech House.Inc, Boston London.
- Mulyanta, Edi, 2003, *Kupas Tuntas Telepon Selular*, Penerbit Andi, Yogyakarta.
- Munawar, 2005, *Pemodelan Visual Dengan UML*, Graha Ilmu, Yogyakarta
- Schmuller, Joseph, 1999, *Teach Yourself UML in 24 Hours*, Sams Publishing, Indianapolis
- Suhendar, dkk, 2002, *Visual Modelling Menggunakan UML dan Rational Rose* Penerbit Informatika Bandung, Bandung.
- Sutedjo, Budi, 2003, *Teleakses Database Pendidikan Berbasis Ponsel*, Penerbit Andi, Yogyakarta
- Wicaksono, Ady, 2002, *Pemrograman Aplikasi Wireless dengan Java*, Penerbit PT Elexmedia Komputindo, Jakarta.
- (www.modem.com/glossary/), 25 November 2005 .
- (<http://student.ugm.ac.id/~mastris/>), 25 November 2005