

Aplikasi Email Client pada Ponsel untuk Memproses Informasi Perkuliahan

Moh. Firomas AN¹, Kodrat IS², Adian FR²

1. Mahasiswa Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang

2. Dosen Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang

Abstract

Perkembangan teknologi ponsel sangat pesat, ponsel sekarang tidak hanya dapat digunakan untuk telepon atau sms saja tetapi sudah dapat berinteraksi dengan internet. Dengan kelebihanannya itu maka ponsel dapat digunakan untuk browsing mencari informasi di internet. Salah satu media yang sangat penting di internet adalah email. Dengan menggunakan email kita dapat mengirim data sangat banyak, jika kita bandingkan dengan SMS (Short messaging Service). Dengan email, kita juga dapat mengirim data kantor, mengirim informasi perkuliahan, tugas kuliah, pesan, dll. Penelitian ini mengimplementasikan Aplikasi Email client Pada Ponsel Untuk Memproses Informasi Perkuliahan. Aplikasi ini ditujukan kepada mahasiswa untuk memperoleh informasi perkuliahan seperti jadwal kuliah, jadwal ujian, jadwal seminar, informasi registrasi ulang, informasi nilai ujian, dll. Dengan adanya aplikasi ini maka kebutuhan akan informasi perkuliahan secara cepat, murah dan aktual dapat terwujud. Khusus informasi jadwal, secara otomatis akan disimpan pada basisdata event ponsel, sehingga akan mengaktifkan alarm sesuai dengan informasi waktu yang ada pada jadwal. Aplikasi ini dikembangkan menggunakan bahasa Pemrograman Java. Penggunaan bahasa pemrograman java karena perkembangannya sangat pesat dan semakin luas digunakan oleh masyarakat serta memiliki kemampuan yang dapat diandalkan.

Keyword : Ponsel, SMS, Email, Java, Email client, informasi Perkuliahan.

I. PENDAHULUAN

1.1 Latar Belakang

Kebutuhan akan informasi perkuliahan secara cepat, mudah, akurat dan murah sangat dibutuhkan untuk menunjang kegiatan perkuliahan, Seperti informasi jadwal ujian, jadwal seminar, jadwal perkuliahan, perubahan jadwal perkuliahan, informasi awal perkuliahan, informasi dosen, informasi nilai, dll. Penggunaan teknologi untuk pengiriman/penerimaan informasi perkuliahan yang umum digunakan adalah *SMS Gateway*. Kendala dalam penggunaan teknologi *SMS Gateway* ini adalah biaya membeli pulsa untuk pengiriman informasi menjadi semakin banyak ketika jumlah mahasiswa dan informasi yang dikirim juga semakin banyak. Pemanfaatan *email* untuk mengirim informasi perkuliahan masih sangat sedikit, padahal penggunaan *email* ini lebih murah jika dibandingkan dengan *SMS* apalagi banyak kampus yang sudah memiliki jaringan internet, sehingga dengan menggunakan *email* ini biaya yang dikeluarkan sudah menjadi satu dengan biaya internet.

Informasi perkuliahan yang dikirim melalui *email* ini harus diterima secara cepat seperti halnya *SMS* agar informasinya benar-benar aktual, untuk itu penggunaan ponsel lebih sesuai daripada menggunakan laptop untuk menerima informasi ini. Untuk saat ini ada satu jenis ponsel yang sudah memiliki fitur untuk menerima *email* secara cepat, yaitu Blackberry. Fitur *push email* yang dimiliki Blackberry menjadikan ponsel ini cukup populer karena fitur ini memungkinkan menerima *email* secara cepat. Harga ponsel Blackberry ini masih cukup mahal sehingga pengguna ponsel ini masih terbatas. Untuk mengatasi permasalahan ini maka diperlukan aplikasi yang dapat digunakan pada ponsel secara umum.

Aplikasi untuk mengambil *email* dari *mail server* lebih dikenal sebagai Aplikasi *email client*. Aplikasi ini sebenarnya sudah banyak digunakan pada komputer *desktop*, seperti *Outlook Express*, *Pegasus Mail*, *Windows Live Mail*, *Eudora*, *Alpine*, *Blitz Mail* dll. Kekurangan dari aplikasi tersebut adalah tertanam pada komputer *desktop* sehingga kurang mendukung untuk seseorang yang memiliki mobilitas tinggi. Dengan aplikasi *email client* pada ponsel untuk memproses informasi perkuliahan diharapkan mahasiswa yang menggunakannya mendapatkan pemberitahuan adanya informasi perkuliahan yang baru secara cepat seperti *push email* yang dimiliki blackberry.

Aplikasi *email client* pada ponsel untuk menerima informasi perkuliahan ini dikembangkan menggunakan bahasa pemrograman *java*. Penggunaan bahasa *java* karena ponsel yang ada sekarang ini kebanyakan sudah mendukung bahasa pemrograman *java* ini. Selain itu kelebihan-kelebihan yang dimilikinya seperti *portable*, *robust*, berorientasi objek, *multithreading*, dinamis, terdistribusi, aman, netral secara arsitektur dan masih banyak lagi kelebihan yang dimiliki oleh bahasa pemrograman *java* ini.

Khusus informasi jadwal secara otomatis akan disimpan pada basis data *event* ponsel, sehingga akan mengaktifkan *alarm* sesuai dengan informasi waktu yang ada pada jadwal. Fasilitas ini sangat bermanfaat sekali untuk mengingatkan mahasiswa tentang kegiatan yang akan diikutinya, seperti kegiatan perkuliahan, seminar, ujian, dll.

1.2 Tujuan

Tujuan Penelitian ini adalah membangun sebuah aplikasi *email client* pada ponsel untuk memproses informasi Perkuliahan, sehingga mempermudah kegiatan perkuliahan

karena informasi perkuliahan disampaikan secara cepat, murah dan aktual menggunakan *email*.

1.3 Batasan Masalah

Pada Penelitian ini pembahasan akan dibatasi pada hal-hal berikut ini:

1. Penelitian ini dirancang dengan menggunakan metode analisis dan perancangan berorientasi objek (*Object-Oriented Analysis and Design*) dan menggunakan bahasa pemodelan UML
2. Penggunaan aplikasi ini pada ponsel yang sudah mendukung pemrograman Java dengan minimal versi CLDC-1.1 dan MIDP-2.0.
3. Tidak membahas *setting* koneksi ponsel ke internet karena untuk *setting* tersebut sangat tergantung dari jenis ponsel dan operator seluler yang digunakan.
4. Aplikasi akan dikembangkan menggunakan JME (Java Micro Edition) SDK 3.0 dan LWUIT (*Lightweight User Interface Toolkit*) untuk tampilan grafisnya.

II. DASAR TEORI

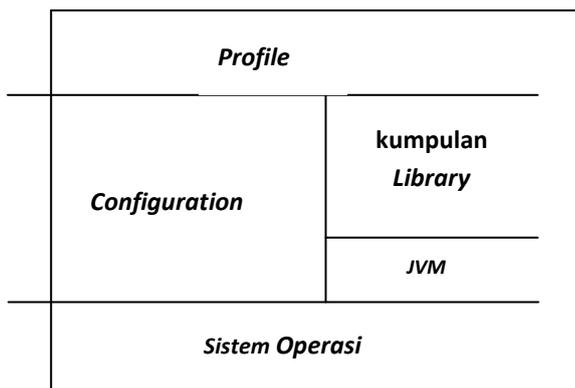
2.1 Email

Email / Elektronik mail adalah media komunikasi yang memungkinkan pertukaran informasi berbasis text untuk dipertukarkan secara elektronik. Seiring dengan perkembangan teknologi, email kemudian memungkinkan pertukaran file digital melalui lampiran (*attachment*). *Email* merupakan sebuah layanan pengiriman surat elektronik yang dikirim melalui internet. *Email* dikirim dari suatu alamat *email* yang terdapat pada sebuah *mail server* kepada alamat *email* yang lainnya yang terdapat pada *mail server* yang sama maupun pada *mail server* yang berbeda.

Email dapat dianalogikan dengan kotak surat yang ada di kantor POS sedangkan *server email* dapat diibaratkan sebagai kantor POS. Dengan analogi ini sebuah *mail server* dapat memiliki banyak *account email* yang ada didalamnya.

2.2 JavaME (Java Micro Edition)

Java Micro Edition atau biasa disebut JME adalah lingkungan pengembangan yang didesain untuk meletakkan perangkat lunak java pada perangkat elektronik. Pada JME, jika perangkat lunak berfungsi baik pada sebuah perangkat maka belum tentu berfungsi baik pada perangkat yang lainnya. JME biasanya digunakan pada telepon seluler, pager, personal digital assistants (PDA's) dan sejenisnya. JME adalah bagian dari JSE, karena itu tidak semua *library* yang ada pada JSE dapat digunakan pada JME. Tetapi JME mempunyai beberapa *library* khusus yang tidak dimiliki JSE. Arsitektur JME dapat dilihat pada gambar 2.1.



Gambar 2.1 Arsitektur JME

Teknologi JME juga memiliki beberapa keterbatasan, terutama jika diaplikasikan pada ponsel. JME sangat tergantung pada perangkat (*divice*) yang digunakan. Bisa dari segi merk ponsel, maupun kemampuan ponsel, dan dukungannya terhadap teknologi JME. Misalnya, jika sebuah ponsel tidak memiliki kamera maka jelas JME pada ponsel tersebut tidak dapat mengakses kamera.

Configuration merupakan java *library* minimum dan kapabilitas yang dimiliki oleh JME. Ada 2 buah *configuration* pada JME, yaitu:

1. CLDC (*Connected Limited Device Configuration*) adalah perangkat dasar dari JME, spesifikasi dasar yang berupa *library* dan API yang diimplementasikan pada JME, seperti yang digunakan pada telepon, pager, dan PDA. Perangkat tersebut dibatasi dengan keterbatasan memori, sumber daya, dan kemampuan memproses. Spesifikasi CLDC pada JME adalah spesifikasi minimal dari *package*, kelas, dan sebagian fungsi *Java Virtual Machine* yang dikurangi agar dapat diimplementasikan dengan keterbatasan sumber daya pada alat-alat tersebut, JVM yang digunakan disebut KVM (*Kilobyte Virtual Machine*).
2. CDC (*Connected Device Configuration*) digunakan pada perangkat yang lebih besar daripada perangkat yang digunakan oleh CLDC. Berikut adalah tabel perbandingan antara CLDC dan CDC:

Tabel 2.1 Perbandingan CLDC dan CDC

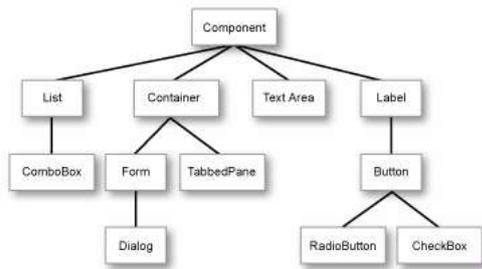
CLDC	CDC
Mengimplementasikan sebagian dari JSE	Mengimplementasikan seluruh fitur JSE
JVM yang digunakan adalah KVM	JVM yang digunakan adalah CVM
Digunakan pada perangkat genggam (handphone, PDA, twoway pager) dengan memori terbatas (160 -512 KB)	Digunakan pada perangkat genggam (Internet TV, Nokia Communicator, car TV) dengan memori minimal 2 MB
Prosesor: 16/32 bit	Prosesor: 32 bit

Profile berbeda dengan *configuration*, *profile* membahas sesuatu yang spesifik untuk sebuah perangkat. Sebagai contoh misalnya, sebuah telepon seluler dengan merk tertentu tentu mempunyai ciri spesifik dengan telepon seluler lainnya. Dalam JME terdapat 2 buah *profile* yaitu, MIDP dan *Foundation Profile*. MIDP atau *Mobile Information Device Profile* adalah spesifikasi untuk sebuah profil JME. MIDP memiliki lapisan di atas CLDC.

2.3 LWUIT

Lightweight User Interface Toolkit (LWUIT) adalah *lightweight widget library* dari SUN *Microsystem* yang terinspirasi oleh *Swing* namun didesain untuk perangkat dengan kapasitas terbatas seperti *mobile Phone* dan PDA. LWUIT mendukung kemampuan *pluggable theme*, hirarki komponen dan kontainer serta abstraksi dibawah GUI *toolkit*.

Interface internal dan kelas abstrak menyediakan *abstraction* dari *interface* dan API pada *profile* sehingga memudahkan *portability* dan migrasi untuk *device* dan *profile* di masa mendatang. LWUIT *menghindari lowest common denominator* yakni mengimplementasikan beberapa fitur pada *platform low-end* dan membawa hasil yang lebih baik pada *platform high-end*. Struktur komponen dari LWUIT ditunjukkan pada gambar 2.3.



Gambar 2.3 Struktur Komponen LWUIT

LWUIT merupakan sebuah *library* untuk *widget* UI dan tidak mengatur tentang bagaimana sistem pada *device* seperti *networking* atau *storage*. Hal yang berkaitan dengan sistem sudah diatur oleh Java ME *profile* sebab LWUIT pada dasarnya adalah sebuah UI *framework* yang menawarkan antar muka yang lebih baik, teratur dan mudah untuk diimplementasikan oleh *developer*.

Untuk membuat menjadi *portable*, LWUIT mengimplementasikan *layer* tipis diatas *native system* dari kanvas dan menyediakan sebuah *widget abstraction*. *Abstraction* tersebut digunakan dengan memanfaatkan kelas-kelas yang ada di dalam sistem seperti *Graphic*, *Image*, dan *Font*.

III. PERANCANGAN SISTEM

3.1 Kebutuhan Sistem

Informasi perkuliahan sangat bermanfaat untuk menunjang kegiatan perkuliahan, seperti informasi jadwal ujian, jadwal seminar, jadwal perkuliahan, perubahan jadwal perkuliahan, informasi awal perkuliahan, informasi dosen, informasi nilai, dll. Kebanyakan dalam penyampaian informasi perkuliahan sementara ini masih menggunakan kertas yang dipasang di papan pengumuman. Penyampaian informasi menggunakan papan pengumuman ini hanya akan dibaca oleh sebagian mahasiswa dan dimungkinkan ketika informasi itu dibaca sudah tidak aktual lagi, untuk itu diperlukan aplikasi yang dapat menerima informasi perkuliahan secara cepat dan aktual.

Agar informasi yang disampaikan bisa diterima secara cepat dan aktual maka dibutuhkan perangkat yang bersifat *portable*. Penggunaan ponsel lebih tepat mengingat ponsel lebih *portable* dan lebih terjangkau harganya dibandingkan dengan perangkat lainnya seperti PC dan laptop. Media penyampain dan penerimaan informasi perkuliahan ini menggunakan *email*.

3.2 Perancangan Perangkat Lunak

Aplikasi ini beroperasi pada ponsel yang memiliki keterbatasan memori, sumber daya, dan kemampuan memproses. Penggunaan *Java Micro Edition* (JME) untuk mengembangkan aplikasi *email client* untuk memproses informasi perkuliahan karena JME didesain untuk meletakkan perangkat lunak java pada perangkat elektronik, seperti ponsel. Selain alasan tersebut penggunaan JME karena hampir semua ponsel yang ada sekarang sudah mendukung penggunaan bahasa pemrograman java ini. Pada pengembangan aplikasi ini *Configuration* yang digunakan adalah CLDC (*Connected Limited Device Configuration*) versi 1.0 dan *profile* yang digunakan MIDP (*Mobile Information Device Profile*) versi 2.0. Untuk *widget* UI menggunakan LWUT (*Lightweight User Interface Toolkit*).

3.2.1 Perancangan Perangkat Lunak Aplikasi Email Client untuk Memproses Informasi Perkuliahan

Agar perangkat lunak dapat bekerja sesuai dengan yang diinginkan, maka sebelumnya perlu dilakukan analisis kebutuhan terhadap perangkat lunak yang akan dirancang. Perangkat lunak diharapkan dapat memberikan fasilitas sebagai berikut:

1. Pengguna dapat menambahkan, mengubah dan menghapus *profile* aplikasi.
2. Pengguna dapat mengubah data *setting* aplikasi.
3. Aplikasi dapat terhubung dengan *mail server* sesuai dengan waktu yang telah ditentukan sebelumnya dan sesuai dengan data *setting* yang dimasukkan untuk mengambil *email*.
4. Aplikasi dapat melakukan validasi *email*, apakah sesuai dengan format data yang telah ditentukan jika sesuai akan disimpan pada *inbox* yang sesuai dengan jenis informasinya.
5. Nada dering hp akan berbunyi jika ada *email* baru yang masuk untuk memberitahu ke pengguna.
6. Pengguna dapat mengubah nada pada *Setting* aplikasi untuk masing-masing jenis informasi.
7. Pengguna dapat menambahkan dan menghapus nada.
8. Aplikasi dapat menyimpan dan menghapus data *email* yang telah divalidasi sebelumnya.
9. Khusus informasi yang berhubungan dengan jadwal akan disimpan pada RMS dan basis data *event* ponsel agar dapat mengaktifkan alarm ponsel sesuai dengan waktu yang ada.

Berdasarkan analisis kebutuhan di atas, maka dapat disusun algoritma untuk perangkat lunak tersebut. Berikut adalah algoritmanya.

1. Menyimpan, mengubah dan menghapus data *profile* aplikasi. Data *profile* ini akan disimpan pada ponsel menggunakan RMS (*Record Management System*). Data *profile* dapat lebih dari satu. Walaupun data *profile* dapat lebih dari satu tetapi hanya ada satu data *profile* yang akan digunakan sebagai *profile* aplikasi setelah ditentukan sebagai *default profile*. *Default profile* dapat diganti-ganti sesuai dengan keinginan pengguna aplikasi pada bagian *setting* aplikasi.
2. Mengubah *setting* aplikasi. Pada menu *setting* ini Pengguna dapat menentukan *default profile* yang akan digunakan, mengisi data nama, NIM, waktu akses, nada jadwal, nada nilai, dan nada umum.
3. Membangun koneksi dengan *mail server* menggunakan protokol SMTP dan POP3. Untuk terhubung dengan *mail server* dibutuhkan data *profile* seperti alamat *mail server* yang dituju, *user* dan *password* untuk *login* ke *mail server*.
4. Setelah koneksi dengan *mail server* berhasil maka bisa dilakukan pengambilan *email* yang ada.
5. *Email* yang diambil akan divalidasi terlebih dahulu oleh aplikasi. Dalam validasi *email* ini ada beberapa tahap, yaitu:
 - a. Melakukan pengecekan format *subject email* apakah sesuai dengan format yang ditentukan, jika tidak sesuai *email* tidak disimpan dan *email* dihapus dari *mail server*.
 - b. Jika *subject email* sesuai dengan format yang ditentukan maka selanjutnya dilakukan pengecekan format pesan *email*, jika tidak sesuai *email* tidak disimpan dan *email* dihapus dari *mail server*.

- c. Jika format *subject* dan format pesan *email* sesuai maka *email* akan disimpan pada ponsel sesuai dengan jenis informasi *email* kemudian email dihapus dari *mail server*.
6. Data *email* yang sudah divalidasi akan disimpan pada ponsel dan dikelompokkan berdasarkan jenis informasi, ada 3 jenis informasi perkuliahan:
 - a. Informasi nilai yaitu informasi yang berhubungan dengan nilai matakuliah. Misalnya nilai ujian semester, nilai ujian semester pendek, dll
 - b. Informasi jadwal yaitu informasi yang didalamnya terdapat informasi waktu. Informasi waktu ini akan digunakan untuk mengaktifkan alarm ponsel sesuai dengan data waktu. Misalnya jadwal ujian, jadwal perkuliahan, dll.
 - c. Informasi perkuliahan umum adalah informasi yang tidak tercakup pada jenis informasi nilai dan informasi jadwal. Misalnya informasi dosen, informasi lomba, informasi perubahan kurikulum perkuliahan, dll.
7. Ketika *email* berhasil disimpan pada ponsel maka nada dering akan aktif..

IV. PENGUJIAN SISTEM

4.1 Implementasi

Dari perancangan yang telah dibuat, diperoleh hasil perancangannya untuk membahas implementasi dan pengujian pada aplikasi yang telah dibuat. Kemudian dianalisis agar sesuai dengan tujuan yang akan dicapai.

4.1.1 Halaman Utama

Halaman utama adalah halaman yang pertama kali ditampilkan ketika aplikasi ini dijalankan. Berikut adalah gambar 4.1 yang menunjukkan halaman utama.



Gambar 4.1 Tampilan Halaman Utama

Ada beberapa tombol yang ditampilkan pada halaman utama yaitu: *Profile*, *Setting*, *Inbox*, *Tones* dan *Help*.

4.1.2 Halaman Profile

Profile digunakan untuk menyimpan *profile* aplikasi yang berhubungan dengan *login* ke *akun email*, seperti alamat *email* pengguna, *password email* pengguna, *inbox type*, *inbox host*, *inbox port*, *SMTP autentikasi*. Aplikasi akan melakukan *login* ke *email* sesuai dengan data yang dimasukkan pada *profile*. Pengguna dapat menyimpan beberapa *profile* sesuai dengan keinginan tetapi hanya ada satu yang akan digunakan oleh aplikasi, yaitu *profile* yang telah ditetapkan sebagai *profile default*. Gambar 4.2 menunjukkan tampilan halaman *profile*.



Gambar 4.2 Tampilan halaman *profile*

4.1.3 Halaman Setting

Pada halaman *setting* kita dapat menyimpan data *setting* aplikasi seperti, *default profile* yang digunakan, nama mahasiswa, NIM (Nomer Induk Mahasiswa), waktu akses, nada jadwal, nada nilai, nada umum. Gambar 4.3 menunjukkan tampilan halaman *Setting*.



Gambar 4.3 Tampilan Halaman Setting

4.1.4 Halaman Inbox

Inbox digunakan untuk menyimpan informasi email. Gambar 4.4 menunjukkan tampilan halaman *inbox*.



Gambar 4.4 Tampilan Halaman *Inbox*

4.2 Pengujian

Pengujian dilakukan untuk mengetahui kekurangan yang ada pada aplikasi, untuk mengetahui aplikasi berjalan sesuai dengan yang diinginkan dan untuk memperkirakan kebiasaan pengguna yang dapat membuat aplikasi menjadi error, seperti mengisi data *profile* yang tidak lengkap. Dengan memperkirakan kesalahan-kesalahan yang umum dilakukan oleh pengguna, maka dapat dilakukan pencegahan

dengan memberikan peringatan kepada pengguna melalui pesan yang muncul secara otomatis jika ada kesalahan yang dilakukan.

4.2.1 Pengujian Login Akun Email

Ketika aplikasi pertama kali dibuka, aplikasi secara otomatis akan melakukan koneksi *internet* untuk *login* ke *akun email* sesuai dengan data *default profile*. Jika terjadi kesalahan, aplikasi akan mengeluarkan pemberitahuan kepada pengguna. Gambar 4.5 menunjukkan pemberitahuan telah terjadi kegagalan *login akun email*.



Gambar 4.5 Menunjukkan Kegagalan *login akun email*

4.2.2 Pengujian Menu Setting

Ada 7 Data pada menu setting yang dapat kita ubah isinya. Pengujian dapat dilakukan pada masing-masing data setting yang diubah. Berikut ini adalah data-data yang ada pada menu setting:

1. **Current Profile**
Kegunaan dari *current profile* adalah untuk *setting profile* aplikasi yang digunakan pada saat aplikasi jalan (*running*). Pengguna dapat mengganti *current profile* sesuai dengan kebutuhannya. Untuk menambah *profile* pengguna dapat menambahkan melalui *menu Add profile*
2. **Nama**
Data nama digunakan untuk menampilkan nama pengguna. Data nama ini akan dipakai ketika ada informasi nilai yang dikirimkan oleh server.
3. **NIM (Nomer Induk Mahasiswa)**
NIM diisi sesuai dengan NIM pengguna. Data NIM digunakan untuk mengambil data informasi nilai mahasiswa sesuai dengan NIM. Gambar 4.6 Menunjukkan tampilan nama dan NIM pada informasi nilai sesuai dengan data yang diisikan pada Setting.



Gambar 4.6 Informasi Nilai

1. **Waktu akses**
Waktu akses digunakan untuk setting akses aplikasi ke akun email, misalnya waktu akses disetting 10 menit, maka aplikasi akan melakukan akses ke akun email setiap 10 menit sekali. Waktu akses tidak boleh dibawah 5 menit, jika pengguna setting dibawah 5 menit maka aplikasi akan mengeluarkan peringatan. Gambar 4.7 menunjukkan peringatan yang keluar jika pengguna melakukan setting di bawah 5 menit.



Gambar 4.7 Peringatan Waktu Akses

2. **Nada Jadwal**
Digunakan untuk penanda jika ada informasi Jadwal yang masuk ke aplikasi. Nada jadwal dapat diubah sesuai dengan keinginan pengguna.
3. **Nada Nilai**
Digunakan untuk penanda jika ada informasi nilai yang masuk ke aplikasi. Nada nilai dapat diubah sesuai dengan keinginan pengguna.
4. **Nada Umum**
Digunakan untuk penanda jika ada informasi umum yang masuk ke aplikasi. Nada umum dapat diubah sesuai dengan keinginan pengguna.

V. PENUTUP

5.1 Kesimpulan

1. Aplikasi dapat memproses informasi perkuliahan yang dibagi menjadi 3 jenis informasi yaitu; informasi perkuliahan nilai, informasi perkuliahan jadwal, dan informasi perkuliahan umum.
2. Khusus informasi jadwal aplikasi akan menyimpan informasi jadwal tersebut ke basis data *event* yang tertanam pada ponsel.
3. Aplikasi dapat ditanam pada ponsel yang sudah mendukung aplikasi java yang minimal memiliki CLDC (*Connected Limited Device Configuration*) versi 1.0 dan MIDP (*Mobile Information Device Profile*) versi 2.0. Media yang digunakan untuk penyampaian informasi menggunakan *email*.
4. *Setting* akun *email* dalam aplikasi hanya ditujukan untuk satu akun *email* saja. namun demikian aplikasi dapat menyimpan beberapa *profile* akun *email* yang dapat digunakan untuk merubah *setting* akun *email* tersebut.
5. *Email* diperoleh dengan cara menarik (*pull*) *email* dari *mail server* dengan interval waktu yang telah ditentukan.

5.2 Saran

Disarankan untuk melakukan penyempurnaan sistem sehingga sistem dapat beroperasi dengan lebih optimal yaitu :

1. Untuk saat ini aplikasi hanya bisa memproses satu akun *email* dalam satu waktu. Diharapkan dalam pengembangan aplikasi ini kedepannya bisa memproses beberapa akun *email* sekaligus dalam satu waktu.
2. Pengambilan email dengan cara menarik (*pull email*) dapat dikembangkan dengan cara mendorong (*push email*) sehingga waktu untuk memperoleh *email* tidak tergantung pada interval waktu tertentu.
3. Pengiriman *email* ke akun *email* pengguna aplikasi *email client* ini masih dilakukan secara *manual*, sehingga dibutuhkan aplikasi *server* untuk mengatur pengiriman *email* tersebut.

Daftar Pustaka

- [1] Shalahuddin, M., *Pemrograman J2ME Belajar Cepat Pemrograman Perangkat Telekomunikasi Mobile*, Informatika Bandung, Bandung, 2006
- [2] Wicaksono, A., *Pemrograman Internet dan XML pada ponsel dengan MIDlet Java*, PT Elex Media Komputindo, Jakarta, 2003
- [3] Wicaksono, A., *Pemrograman Aplikasi Wireless dengan Java*, PT Elex Media Komputindo, Jakarta, 2002
- [4] Yuan, M.Y., *Enterprise J2ME: Developing Mobile Java Applications*, Prentice Hall PTR, 2003
- [5] ---, *Java ME*, <http://www.oracle.com/technetwork/java/javame/overview/index.html>, Juli, 2010.