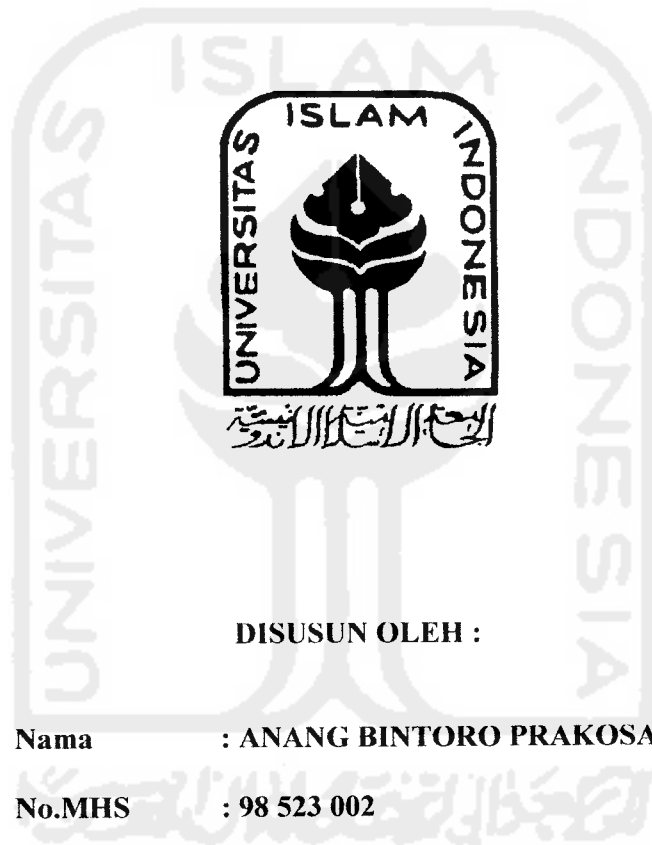


**EMAIL CLIENT MENGGUNAKAN
JAVA 2 MICRO EDITION**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika**

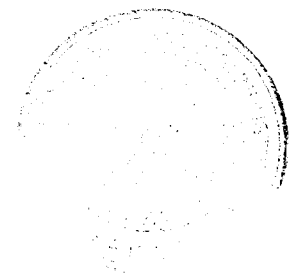


DISUSUN OLEH :

Nama : ANANG BINTORO PRAKOSA

No.MHS : 98 523 002

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2007**



LEMBAR PENGESAHAN PENGUJI

**EMAIL CLIENT MENGGUNAKAN
JAVA 2 MICRO EDITION**

Tugas Akhir

Disusun Oleh

**Nama : ANANG BINTORO PRAKOSA
No.Mhs : 98 523 002**

**Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia**

Yogyakarta, 26 Juli 2007

Tim Penguji

Tanda Tangan

Hendrik, ST



Taufiq Hidayat, ST, M.CS



Yudi Prayudi, S.Si, M.Kom



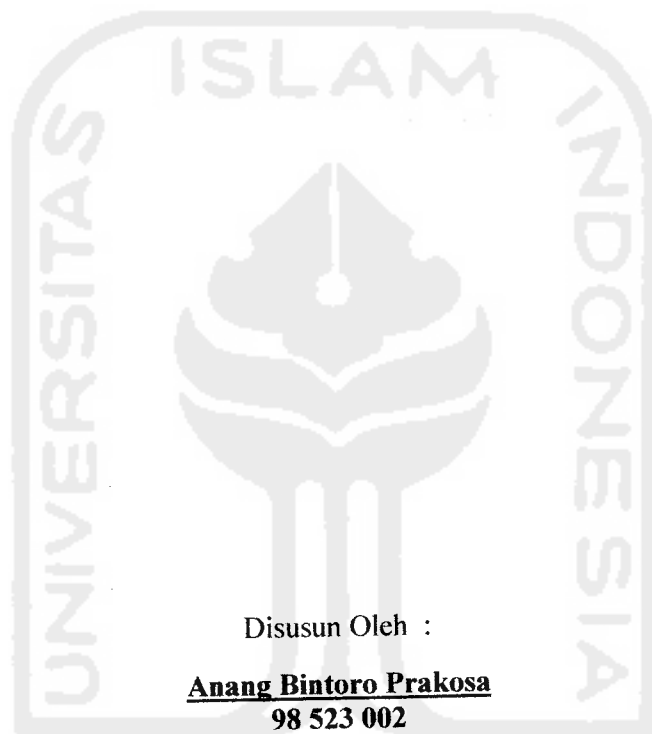
**Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Teknologi Industri
Universitas Islam Indonesia**



(Yudi Prayudi, S.Si, M.Kom)

LEMBAR PENGESAHAN PEMBIMBING
EMAIL CLIENT MENGGUNAKAN
JAVA 2 MICRO EDITION

TUGAS AKHIR

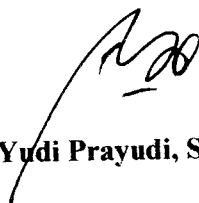


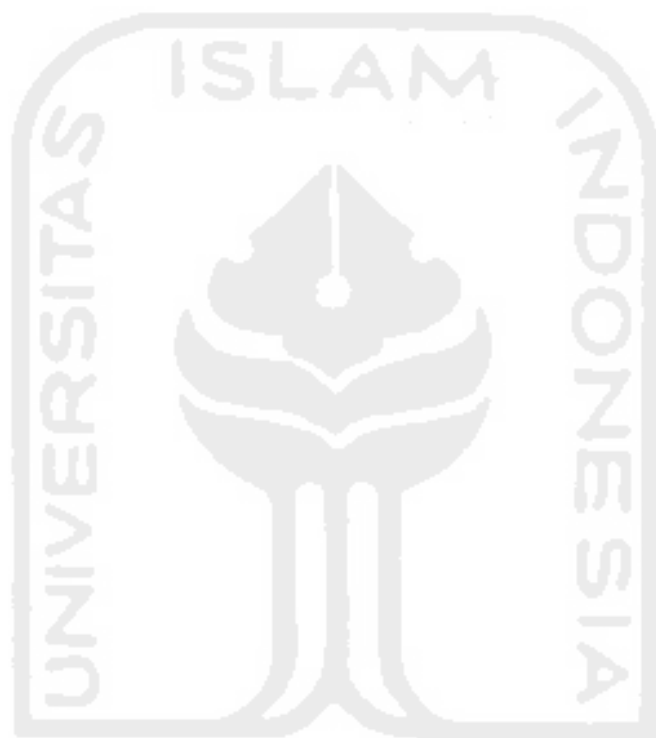
Disusun Oleh :

Anang Bintoro Prakosa
98 523 002

Yogyakarta, 16 Juli 2007

Dosen Pembimbing Tugas Akhir


(Yudi Prayudi, S.Si, M.Kom.)



Persembahkan

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Tugas akhir ini dipersembahkan kepada

1. Teristimewa Buat Kedua Orang Tua-ku Bapak Budi Pramono dan Ibu Mudji Wartini yang telah memberiku materi, moril serta doanya siang malam.
2. Buat Kakakku Sundoro Adjie P dan Pramawati, Amd beserta suaminya Suhartadi, SH yang telah memberikan bantuan materi, moril serta doanya, terima kasih.
3. Buat Keluarga Besar Ayahku, Ibuku dan semua saudara sepupuku, terima kasih atas bantuannya.
4. Buat Bapak Yudi Prayudi, S.Si., M.Kom. yang telah membimbing dalam pembuatan laporan dan lainnya, terima kasih.
5. Buat Keluarga Besar MAPALA UNISI yang telah memberikan pengalaman-pengalaman yang sangat berharga.
6. Buat keluarga besar IMPUS Cell (Tatok, Susilo, Adjie, Damarso) dan IMPUS PALASEKU terima kasih atas dukungan dan pengalamannya.
7. Buat Panji Bawono terima kasih atas bantuannya selama ini (terima kasih tidak akan cukup untuk mu)
8. Beni, Rio, Novri, Bedjo, dan teman-teman INFORMATIKA '98 yang tidak tersebut, terima kasih banyak atas dukungannya.
9. Irawati, V. Ratri Wulandari, Yani, Yuli Astuti terima kasih atas kenangan yang sangat indah dan atas waktunya walaupun cuma sebentar,
10. Buat Heni ku, terima kasih dan jangan pernah bosan menemani aku.

MOTTO

“Ketahuilah bahwa Allah SWT tidak pernah meninggalkan seorangpun umat-Nya sendirian”

“Lingkungan itu sangat mempengaruhi tapi keputusan terakhir ada pada diri sendiri”

“Masa depan adalah apa yang kamu perbuat saat ini”

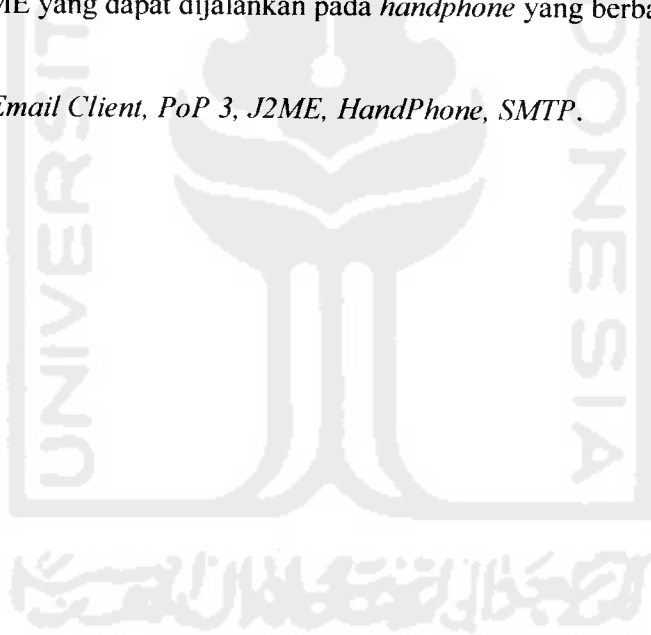
“Build your own future”

ABSTRAKSI

Handphone merupakan pilihan komunikasi dengan ruang gerak luas, tetapi memerlukan biaya yang mahal. Adapun layanan yang dapat digunakan pada *handphone* untuk melakukan komunikasi dengan biaya yang relatif murah adalah SMS (*Short Message Service*). Tetapi SMS memiliki keterbatasan dalam jumlah karakter isi untuk pengiriman yaitu sebanyak 160 karakter. Dan untuk mengirimkan SMS antar operator telepon apalagi ke luar negeri memerlukan biaya yang lebih mahal.

Sejalan dengan keperluannya maka perkembangan teknologi pada *handphone* terus bertambah, salah satu contohnya adalah penerapan teknologi J2ME pada *handphone*. Hal ini memungkinkan penerapan aplikasi yang ditulis dengan bahasa pemrograman Java pada *handphone*. Dengan adanya fungsi tersebut maka berbagai macam aplikasi dapat dibuat dan dijalankan pada *handphone*. Aplikasi pengiriman maupun membaca email (*Email Client*) melalui *handphone* dapat dibuat untuk mengatasi masalah diatas dengan memanfaatkan SMTP dan POP 3 Server. Hasil akhir penelitian ini berupa aplikasi Email Client berbasis J2ME yang dapat dijalankan pada *handphone* yang berbasis java.

Keywords: Email Client, PoP 3, J2ME, HandPhone, SMTP.



TAKARIR

<i>Email</i>	=	Surat elektronik
<i>Email Client</i>	=	Pengguna surat elektronik
<i>Handphone</i>	=	Telepon genggam
<i>Software</i>	=	Perangkat lunak
<i>Hardware</i>	=	Perangkat keras
<i>SMS</i>	=	Pesan singkat
<i>Server</i>	=	Pusat layanan
<i>Record</i>	=	Merekam
<i>Store</i>	=	Persediaan
<i>Inbox</i>	=	Kotak masuk
<i>Forward</i>	=	Melanjutkan
<i>About</i>	=	Tentang
<i>Mobile</i>	=	Mudah bergerak
<i>Device</i>	=	Alat
<i>Handled device</i>	=	Alat yang dapat dipegang
<i>Interface</i>	=	Antar muka
<i>Manufacture</i>	=	Membuat
<i>Field</i>	=	Bidang
<i>User name</i>	=	Nama pengguna
<i>Password</i>	=	Kata kunci
<i>Download</i>	=	Mengunduh
<i>Splash Screen</i>	=	Tampilan awal

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN PEMBIMBING	ii
LEMBAR PENGESAHAN PENGUJI	iii
HALAMAN PERSEMBAHAN	iv
HALAMAN MOTTO	vii
KATA PENGANTAR	viii
ABSTRAKSI	x
TAKARIR	xi
DAFTAR ISI	xii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Metodologi Penelitian	4
1.6.1 Teknik Pengumpulan Data	4
1.6.2 Sumber Data	4
1.6.3 Sistematika Penulisan	4
BAB II LANDASAN TEORI	7
2.1 Java	7
2.2 J2ME	9
2.2.1 J2ME Configuration	10
2.2.2 J2ME Profile	11
2.2.3 Netbeans	13
2.3 HTTP Connection	15
2.4 Hypertekst Transfer Protocol (HTTP)	18
2.5 Record Management System (RMS)	20
2.6 Handphone	21
2.7 Multipurpose Internet Mail Extentions (MIME)	23
2.8 Electronic Mail (Email)	25

BAB III	ANALISIS KEBUTUHAN PERANGKAT LUNAK.....	28
	3.1 Metode Analisis.....	28
	3.2 Hasil Analisis	29
	3.2.1 Analisis Data Masukan.....	29
	3.2.2 Analisis Proses.....	29
	3.2.3 AnalisisData Keluaran.....	30
	3.2.4 Analisis Kebutuhan Antarmuka.....	30
	3.2.5 Kebutuhan Sistem	31
	3.2.6 Kebutuhan Bahasa Pemrograman	31
BAB IV	PERANCANGAN PERANGKAT LUNAK	33
	4.1 Metode Perancangan.....	33
	4.2 Hasil Perancangan	34
	4.2.1 Prinsip Kerja Sistem.....	34
	4.2.2 Algoritma Sistem	34
	4.2.2.1 Prinsip Kerja Aplikasil.....	35
	4.2.2.2 Pengolahan Data Pada RMS.....	36
	4.2.2.3 Pengiriman Email.....	37
	4.2.2.4 Cek Email.....	37
	4.2.3 Diagram Arus Data (DFD)	38
	4.2.4 Perancangan Program.....	42
	4.2.5 Perancangan Bagian Antarmuka.....	42
	4.2.5.1 Perancangan Tampilan Antarmuka.....	42
BAB V	IMPLEMENTASI PERANGKAT LUNAK	50
	5.1 Batasan Implementasi	50
	5.1.1 Perangkat Keras	50
	5.1.2 Perangkat Lunak	51
	5.2 Implementasi Perangkat Lunak.....	51
	5.2.1 Tampilan Awal.....	51
	5.2.2 Tampilan Menu Utama.....	52
	5.2.3 Form Inbox.....	53
	5.2.4 Form Menu Kirim Email.....	55
	5.2.5 Form Konfigurasi.....	57
	5.2.6 Form Daftar Email.....	57
	5.2.7 Form Data Email.....	58
	5.2.8 Menu About.....	59
BAB VI	ANALISIS KINERJA PERANGKAT LUNAK	60
	6.1 Pengujian Program	60
	6.2 Pengujian dan Analisis.....	60
	6.2.1 Pengujian Normal	60
	6.2.1.1 Pemasukan Data.....	61
	6.3 Kelebihan Dan Kekurangan Aplikasi.....	63
	6.3.1 Kelebihan.....	64

6.3.2 Kekurangan.....	64
BAB VII PENUTUP	65
7.1 Kesimpulan	65
7.2 Saran	65

DAFTAR PUSTAKA



DAFTAR GAMBAR

Gambar 2.1	Java 2 Edition dan targetnya	9
Gambar 2.2	IDE Netbeans	14
Gambar 2.3	Hirarki <i>Interface Connection</i>	17
Gambar 2.4	Koneksi Jaringan HTTP.....	18
Gambar 2.5	Struktur <i>Record Store</i>	20
Gambar 2.6	Alur Kerja Email	26
Gambar 4.1	Alur Kerja Program	34
Gambar 4.2	Koneksi Antara <i>client</i> dan <i>server</i>	34
Gambar 4.3	Diagram Konteks Pengaksesan Email melalui <i>Hand Phone</i>	39
Gambar 4.4	Diagram arus data (DFD) level 1 Pengaksesan Email Melalui <i>Hand Phone</i>	40
Gambar 4.5	Diagram arus data (DFD) level 2 Proses Daftar Email	41
Gambar 4.6	Diagram Arus Data (DFD) level 2 Proses Konfigurasi	41
Gambar 4.7	Form Tampilan awal.....	43
Gambar 4.8	Form Menu Utama.....	43
Gambar 4.9	Form Menu Inbox.....	44
Gambar 4.10	Form Baca Email.....	44
Gambar 4.11	Form Balas Email.....	45
Gambar 4.12	Form Forward.....	46
Gambar 4.13	Form Kirim Email	46
Gambar 4.14	Form Konfigurasi	47
Gambar 4.15	Form Daftar Email.....	48
Gambar 4.16	Form Data Email	48
Gambar 4.17	Form Tambah Data.....	49
Gambar 4.18	Form About	49
Gambar 5.1	Tampilan Awal	51
Gambar 5.2	Tampilan Splash Screen.....	51
Gambar 5.3	Form Menu Utama.....	52
Gambar 5.4	Form Menu Inbox.....	53
Gambar 5.5	Form Menu Baca Email.....	55
Gambar 5.6	Form Menu Kirim Email	56
Gambar 5.7	Form Menu Konfigurasi	57
Gambar 5.8	Form Menu Daftar Nama.....	58
Gambar 5.9	Form Menu Data Email	59
Gambar 5.10	Form Menu About	59
Gambar 6.1	Pesan Data Berhasil Disimpan	61
Gambar 6.2	Pesan Data Kurang Lengkap.....	61
Gambar 6.3	Pesan Email Terkirim	62
Gambar 6.4	Pesan Tujuan Alamat Email Harus Diisi	63
Gambar 6.5	Pesan Data Email Harus Lengkap	63

DAFTAR TABEL

Tabel 2.1	Tabel Perbandingan CDC dan CLDC.....	11
Tabel 2.2	Tabel Jenis-jenis Protocol dan Tipe Koneksinya	17
Tabel 2.3	Tabel <i>Header</i> pada MIME	24
Tabel 2.4	Tabel Tipe dan Subtipe pada <i>header</i> MIME.....	24
Tabel 2.5	Tabel <i>Message header</i> pada <i>message transfer system</i>	27
Tabel 4.6	Tabel <i>Message header</i> pada UA/ <i>User</i>	27



BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi di bidang komunikasi serta komputer yang pesat makin memudahkan kita dalam berkomunikasi. Dengan adanya dunia internet yang menghadirkan berbagai pilihan cara untuk berkomunikasi secara langsung maupun tidak langsung seperti *email*, *chatting*, *ftp*, *telnet* dan lain sebagainya dapat dilakukan dengan biaya murah, tetapi untuk melakukan komunikasi tersebut pengguna harus menggunakan komputer yang ruang geraknya terbatas.

Handphone merupakan pilihan komunikasi dengan ruang gerak luas, tetapi memerlukan biaya yang mahal. Adapun layanan yang dapat digunakan pada *handphone* untuk melakukan komunikasi dengan biaya yang relatif murah adalah SMS (*Short Message Service*). Tetapi SMS memiliki keterbatasan dalam jumlah karakter isi untuk pengiriman yaitu sebanyak 160 karakter. Dan untuk mengirimkan SMS antar operator telepon apalagi ke luar negeri memerlukan biaya yang lebih mahal.

Sejalan dengan keperluannya maka perkembangan teknologi pada *handphone* terus bertambah, salah satu contohnya adalah penerapan teknologi Java pada *handphone*. Hal ini memungkinkan penerapan aplikasi yang ditulis dengan bahasa pemrograman Java pada *handphone*. Dengan adanya fungsi tersebut maka berbagai macam aplikasi dapat dibuat dan dijalankan pada

handphone. Aplikasi pengiriman maupun membaca *email* melalui *handphone* dapat dibuat untuk mengatasi masalah diatas.

1.2 Rumusan Masalah

Dari latar belakang yang diuraikan diatas dapat diambil rumusan yang akan menjadi pembahasan penelitian ini adalah **“Bagaimana membangun *Email Client* menggunakan Java 2 Microedition yang dapat diakses menggunakan *handphone*”**

1.3 Batasan Masalah

Pembatasan masalah pada pembuatan aplikasi ini bukan saja untuk menyederhanakan persoalan yang dihadapi, tetapi juga untuk menyederhanakan persoalan tersebut agar tidak menyimpang dari yang diinginkan. Batasan-batasan penelitian ini adalah:

1. Wujud Perancangan adalah *software* aplikasi berupa *Email Client* berbasis java menggunakan Java 2 MicroEdition.
2. *Handphone* yang digunakan adalah *handphone* yang berbasis java, seperti Motorola i 85 s, Siemens M50, Simens M55, Sony Erricson T 610 dan mendukung koneksi internet GPRS/Edge maupun 3G.
3. *Email* yang digunakan harus mendukung POP 3 dan didukung oleh operator GSM.

4. *Email Client* hanya dapat menerima dan mengirim *email* berupa teks dan tidak dapat menerima *attachment* dikarenakan kapasitas penyimpanan (*memory*) *handphone* yang terbatas.

1.4 Tujuan penelitian

Penelitian ini dilakukan dengan beberapa tujuan yang dapat dijelaskan sebagai berikut:

1. Membangun perangkat lunak *Email Client* menggunakan Java 2 Micro Edition yang dapat diakses secara *on line* menggunakan *handphone*
2. Mengetahui unjuk kerja *aplikasi java* sebagai bahasa pemrograman dengan studi kasus membangun *Email Client* menggunakan Java 2 Micro Edition.

1.5 Manfaat Penelitian

Dengan adanya penelitian yang dilakukan ini diharapkan dapat memberi manfaat antara lain:

1. Memberikan kemudahan bagi pengguna *Handphone* untuk mengakses *email*
2. Mengoptimalkan fungsi *hanphone* sebagai alat komunikasi yang serba praktis
3. Menambah wawasan dalam hal pemrograman dan desain program aplikasi.

4. Menerapkan pengetahuan dalam pembuatan model sistem berbasis bahasa pemrograman dengan bahasa Java.

1.6 Metodologi Penelitian

Teknik dan metode dalam pengumpulan data dan pengolahan data yang diperoleh adalah sebagai berikut:

1.6.1 Teknik pengumpulan data

Dalam teknik pengumpulan data dilakukan 2 metode yaitu :

1. Metode langsung, dilakukan dengan melakukan pengamatan langsung terhadap data yang perlu dimasukkan kedalam model sistem.
2. Metode Tidak langsung, dengan cara studi pustaka, yaitu Mengumpulkan data dari buku-buku ilmiah sebagai acuan dan artikel yang ada di internet.

1.6.2 Sumber data

Data yang dijadikan sumber data adalah data sekunder yaitu data yang diperoleh dari tinjauan pustaka maupun data studi literatur, mencakup buku-buku, artikel, dan tutorial melalui internet.

1.6.3 Sistematika Penulisan

BAB I PENDAHULUAN

Bab ini berisikan latar belakang pembuatan, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini menguraikan tentang teori-teori yang relevan dengan aspek objek penelitian yang digunakan sebagai dasar untuk pembahasan masalah. Teori tersebut antara lain bahasa pemrograman java, J2ME, HTTP *Connection*, HTTP, *Record Management System (RMS)*, *Multipurpose Internet Mail Extensions (MIME)*, *handphone*, dan *electronic mail*.

BAB III ANALISIS KEBUTUHAN PERANGKAT LUNAK

Bab ini menguraikan semua perangkat lunak yang digunakan sebagai input yang dimasukkan dalam pembuatan model sistem dan perangkat keras untuk mengoperasikan perangkat lunak yang dibuat. Proses yang dijalankan dan output yang diharapkan.

BAB IV PERANCANGAN PERANGKAT LUNAK

Bab ini memuat tentang konsep perancangan perangkat lunak, metode perancangan perangkat lunak yang meliputi struktur perancangan, *Data Flow Diagram (DFD)* dan perancangan antar muka.

BAB V IMPLEMENTASI PERANGKAT LUNAK

Dari hasil perancangan pada bab sebelumnya bisa dijadikan landasan untuk mengimplementasikan kedalam program serta memberikan batasan implementasi baru kemudian mengimplementasikan rancangan.

BAB VI ANALISIS KINERJA PERANGKAT LUNAK

Bab ini berisi tentang pengujian perangkat lunak yang telah dihasilkan dibandingkan kesesuaiannya dengan kebutuhan perangkat lunak.

BAB VII PENUTUP

Pada bagian ini ditarik suatu kesimpulan atas apa yang dibuat, serta saran perbaikan perangkat lunak untuk masa yang akan datang.



BAB II

LANDASAN TEORI

2.1 Java

Java merupakan bahasa pemrograman yang dikembangkan oleh Sun Microsystems. Dari pertama diluncurkannya, Java memiliki semboyan *write one run everywhere*. Java mempunyai beberapa keunggulan, antara lain [WIC02]:

1. Sederhana

Bahasa pemrograman Java mudah dipahami dan dipelajari.

2. Berorientasi Obyek

Java berbasis pada pemrograman berorientasi obyek

3. Kuat dan aman

Java banyak menekankan pada pengecekan awal untuk kemungkinan terjadinya masalah, pengecekan pada saat *runtime* dan mengurangi timbulnya kesalahan. Java memiliki pengaturan yang baik khususnya pada masalah *buffer overflow* yang umumnya menjadi lubang keamanan.

4. Arsitektur Netral dan *Portable*

Java dirancang untuk dapat dijalankan pada semua *platform*. *Compiler* Java yang digunakan untuk mengkompilasi kode program Java dirancang untuk menghasilkan kode yang netral terhadap semua arsitektur perangkat keras yang disebut dengan Java *bytecode*.

Bahasa Java merupakan bahasa yang cocok untuk *wireless device*. Hal ini disebabkan adanya keuntungan yang disediakan oleh Java platform antara lain [GIG 01]:

1. *Dynamic delivery of content* adalah aplikasi dan layanan baru dapat di-*download* sewaktu waktu melalui jaringan yang berbeda
2. *Security* adalah verifikasi *file Class* dan *interface* program aplikasi yang terdefinisi menyebabkan *third party application* tahan uji dan tidak membahayakan *device* ataupun jaringan
3. *Cross platform compatibility* adalah aplikasi yang dapat ditransfer secara fleksibel antara *device* yang berbeda
4. *Offline acces* adalah aplikasi tetap yang dapat digunakan walau tanpa koneksi. Hal ini menurunkan biaya dan mengurangi kesalahan jaringan.
5. *Large developer community*, pada saat ini diperkirakan terdapat lebih dari 2.5 juta pengembang *software* Java di dunia.

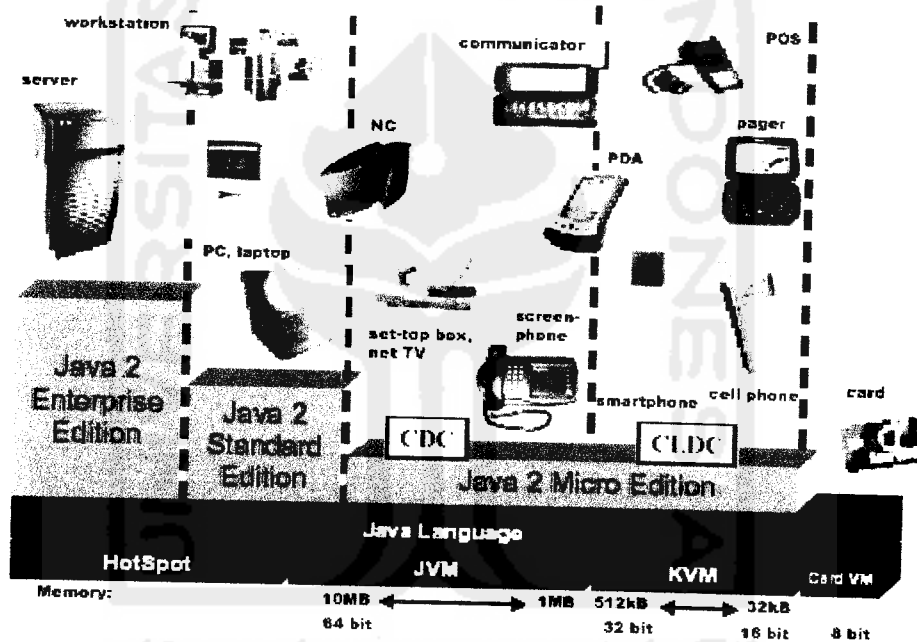
Pada perkembangan selanjutnya Sun Microsystem memperkenalkan Java versi 1.2 atau lebih dikenal dengan Java 2. Java 2 terdiri atas JDK dan JRE. Java 2 ini dibagi menjadi 3 kategori yaitu:

1. *J2SE (Java 2 Standard Edition)*
Java kategori ini digunakan untuk menjalankan dan mengembangkan aplikasi-aplikasi java pada level PC (*Personal Computer*)
2. *J2EE (Java 2 Enterprise Edition)*
Kategori ini digunakan untuk menjalankan dan mengembangkan aplikasi-aplikasi Java pada lingkungan *enterprise*, dengan menambah

fungsionalitas-fungsionalitas Java semacam EJB (*Enterprise Java Bean*), Java CORBA, Servlet dan JSP, serta Java XML (*Extensible Markup Language*)

3. J2ME (*Java 2 Micro Edition*)

Pada kategori ini Java digunakan untuk menjalankan dan mengembangkan aplikasi-aplikasi Java pada *mobile device* semacam *handphone*, Palm, PDA, dan Pocket PC



Gambar 2.1 *Java 2 Edition* dan targetnya

2.2 J2ME

Java 2 Micro Edition (J2ME) merupakan salah satu bagian dari Java yang dikembangkan untuk pembuatan program yang dapat dijalankan pada perangkat-perangkat *mobile* seperti Palm, Pocket PC, *Handphone* yang mendukung Java.

Pada J2ME dibagi menjadi 2 bagian yaitu *J2ME Configuration* dan *J2ME Profile* [WIC02].

2.2.1 *J2ME Configuration*

Karena banyaknya jenis perangkat *mobile* dan memiliki fitur-fitur yang berbeda-beda, maka *J2ME Configuration* menyediakan fungsi standar yang mengimplementasikan fitur standar yang dimiliki oleh sebuah perangkat *mobile*. *J2ME Configuration* memastikan terdapat keseragaman karakteristik *Java development environment* yang terdapat pada semua jenis *Java enabled devices* yang memiliki kesamaan dalam spesifikasi sistem, meliputi kemampuan prosesor dan kapasitas *memory*.

Ada 2 kategori pada *J2ME Configuration* berdasarkan dari kapasitas *memory* dan prosesor, yaitu:

1. *CDC (Connected Device Configuration)*

Kategori ini digunakan untuk aplikasi *Java* pada perangkat – perangkat *handheld devices* yang memiliki kapasitas *memory* minimal 2 *Megabytes*. Contohnya *Internet TV*, *Car Television* dan *Nokia Communicator*.

2. *CLDC (Connected Limited Device Configuration)*

Kategori ini digunakan untuk aplikasi *Java* pada *PDA* dan *handphone* yang mendukung teknologi *Java*, seperti *Nokia*, *Siemens*, *Samsung*. Pada umumnya perangkat-perangkat tersebut memiliki kapasitas *memory* 160-512 *Kilobytes*.

Tabel 2.1 Perbandingan CDC dan CLDC

CDC	CLDC
Mengimplementasikan seluruh fitur dari J2SE	Mengimplementasikan subset dari J2SE
Digunakan pada perangkat <i>handled devices</i> yang memiliki <i>memory</i> minimal 2 Mbytes	Digunakan pada perangkat <i>handled devices</i> dengan ukuran <i>memory</i> 160 – 512 Kilobytes
Prosesor : 32 Bit	Prosesor 16 atau 32 Bit

2.2.2 J2ME Profile

J2ME *Profile* mendefinisikan tambahan kumpulan fungsi yang bersifat lebih spesifik dari sebuah perangkat *handled devices*. Sebagai contoh sebuah *handphone* memiliki kemampuan untuk menelpon ke nomor *handphone* lainnya, ini merupakan fungsi standar yang dimiliki oleh *handphone* yang diimplementasikan oleh J2ME *Configuration*. Selain memiliki fungsi-fungsi standar, *handphone* memiliki fungsi khusus yang hanya dimiliki oleh *handphone* tertentu, sebagai contoh Siemens SL45 dapat menjalankan *file* MP3. Nokia dengan *game snake* yang menjadi *trademark*-nya. Fitur-fitur tersebut tergantung pada jenis *handled devices* dan akan diimplementasikan oleh J2ME *Profile*.

Ada 5 kategori pada J2ME *profile*, yaitu

1. *Mobile Information Device Profile* (MIDP)

CLDC *profile* MIDP yang dirancang khusus untuk *wireless phone* dan *two way pager*.

MIDP menyediakan *library* Java:

a. Antarmuka (GUI)

Library ini menyediakan fungsi-fungsi yang berhubungan dengan antarmuka dari *handled device* seperti *Screen*, *Item*, *Canvas* dan lain lain.

b. Jaringan (*Networking*)

Library ini menyediakan fungsi jaringan yang berbasis HTTP.

c. *Record (Records Managament System)*

Library ini menyediakan fungsi *database* sederhana yang dapat melakukan penambahan data, penghapusan data serta perubahan data.

d. *Timer*

Library ini menyediakan fungsi waktu.

2. *Foundation Profile (FP)*

CDC *profile* yang dikhususkan untuk digunakan oleh *device* yang membutuhkan implementasi yang menyeluruh dari Java *virtual machine* dan termasuk didalamnya API dari J2SE.

3. *Personal Profile*

CDC *profile* yang dikembangkan oleh Sun's Personal Java *Environment*. *Profile* ini menyediakan kompatibilitas bagi perkembangan aplikasi untuk versi 1.1.x dan versi 1.2.x.

4. *RMI Profile*

CDC *profile* yang mendefinisikan API dari J2SE RMI.

5. *Personal Digital Assistance Profile*

CLDC *profile* yang didesain khusus untuk *handheld device* yang dikeluarkan oleh Palm OS seperti Palm Pilots dan Visor

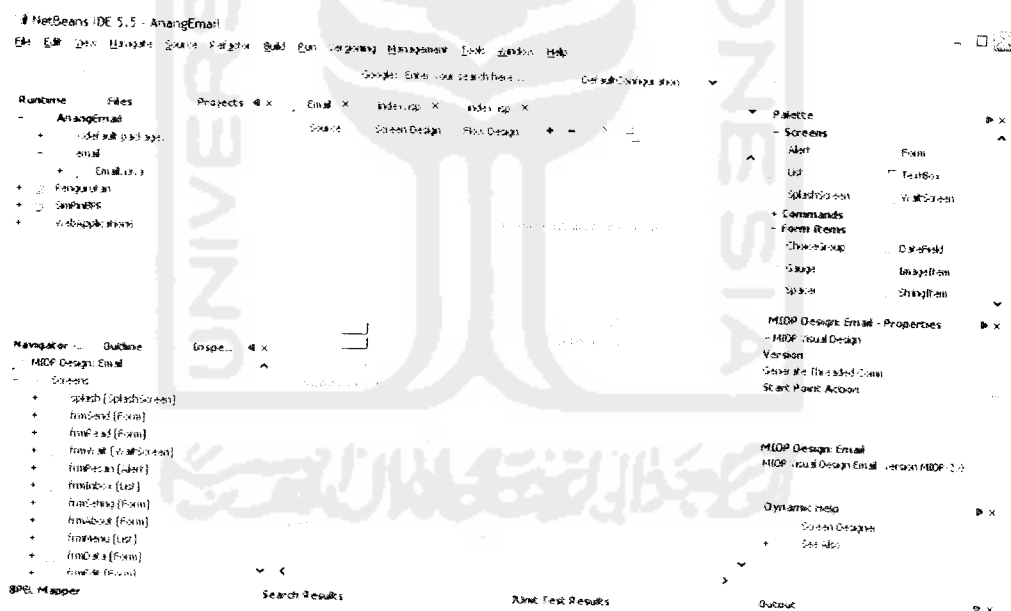
2.2.3 Netbeans

Sebelum adanya IDE (*Integrated Development Environmet*) dalam melakukan pemrograman bahasa java menggunakan teks *editor* (misalnya Vi, Notepad, Textpad) yang kemudian di-*compile* menggunakan java SDK . Menggunakan perintah `javac <nama file java>`. Hasil proses tersebut berupa *file java (class)* yang dapat dijalankan dengan menggunakan perintah `java <nama class java>`.

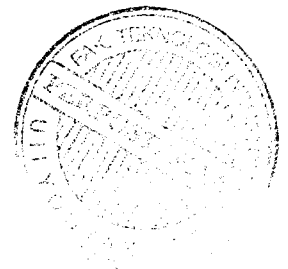
Sekarang terdapat banyak IDE sebagai alat bantu dalam melakukan pemrograman dengan memanfaatkan bahasa java. Ada yang berupa perangkat lunak berbayar (JBuilder) maupun *open source* (Eclipse, Netbeans). Salah satu yang merupakan perangkat lunak gratis adalah NetBeans (<http://www.netbeans.org>). Merupakan perangkat lunak IDE yang dikembangkan oleh SUN Microsistem yang sekarang bersifat terbuka untuk dikembangkan. NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra (dan terus bertambah!). Sun Microsystems mendirikan proyek kode terbuka NetBeans pada bulan Juni 2000 dan terus menjadi sponsor utama. Saat ini terdapat dua produk: NetBeans IDE dan NetBeans Platform.

NetBeans IDE adalah sebuah lingkungan pengembangan sebuah kaskas untuk *programmer* menulis, mengompilasi, mencari kesalahan dan menyebarkan

program. Netbeans IDE ditulis dalam Java, namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas Netbeans IDE. Netbeans IDE adalah sebuah produk bebas dengan tanpa batasan bagaimana digunakan. Tersedia juga *NetBeans Platform*; sebuah fondasi yang modular dan dapat diperluas yang dapat digunakan sebagai perangkat lunak dasar untuk membuat aplikasi desktop yang besar. Mitra ISV menyediakan *plug-in* bernilai tambah yang dapat dengan mudah diintegrasikan ke dalam *Platform* dan dapat juga digunakan untuk membuat kaskas dan solusi sendiri. Kedua produk adalah kode terbuka (*open source*) dan bebas (*free*) untuk penggunaan komersial dan non komersial. Kode sumber tersedia untuk guna ulang dengan lisensi *Common Development and Distribution License (CDDL)* [HAR06].



Gambar 2.2 IDE *Netbeans*



2.3 HTTP Connection

Untuk dapat melakukan komunikasi data berbasis jaringan diperlukan *library* Java.io dan Java.net yang terdapat pada J2EE/J2SE. Tetapi *library* tersebut tidak dapat diimplementasikan secara langsung pada MID, hal ini disebabkan karena kedua *library* tersebut terlalu besar untuk diimplementasikan pada MID yang memiliki spesifikasi CLDC dengan kapasitas *memory* 160-512 Kilobytes pada J2ME. *Generic Connection Framework* bertujuan menyediakan fungsionalitas serupa dengan yang disediakan paket `java.io` dan `java.net` pada J2SE / J2EE, secara lebih fleksibel, ekstensibel, dan sesuai dengan karakteristik teknologi komunikasi data pada MID.

Sesuai sifat spesifikasi CLDC, maka *Generic Connection Framework* mendefinisikan komunikasi data seumum mungkin sehingga dapat mengakomodasi berbagai perbedaan mekanisme komunikasi data, serta dapat dengan mudah menyesuaikan berbagai perkembangan mekanisme komunikasi data.

Definisi tersebut diwujudkan dengan mendefinisikan tujuh buah *interface* yang masing-masing mewakili tujuh jenis bentuk komunikasi data yang paling mendasar.

Tujuh jenis komunikasi data (*connection*) tersebut meliputi:

1. *Interface Connection*

Melambangkan bentuk komunikasi data paling sederhana dan mendasar.

2. *Interface InputConnection*

Subinterfaces dari *Connection* yang memungkinkan penggunaan *input stream*.

3. *Interface OutputConnection*

Subinterfaces dari *Connection* yang memungkinkan penggunaan *output stream*.

4. *Interface StreamConnection*

Subinterfaces dari *InputConnection* dan *OutputConnection*, menyediakan kemampuan untuk mengirim serta menerima data menggunakan soket tanpa fasilitas pengolahan data.

5. *Interface ContentConnection*

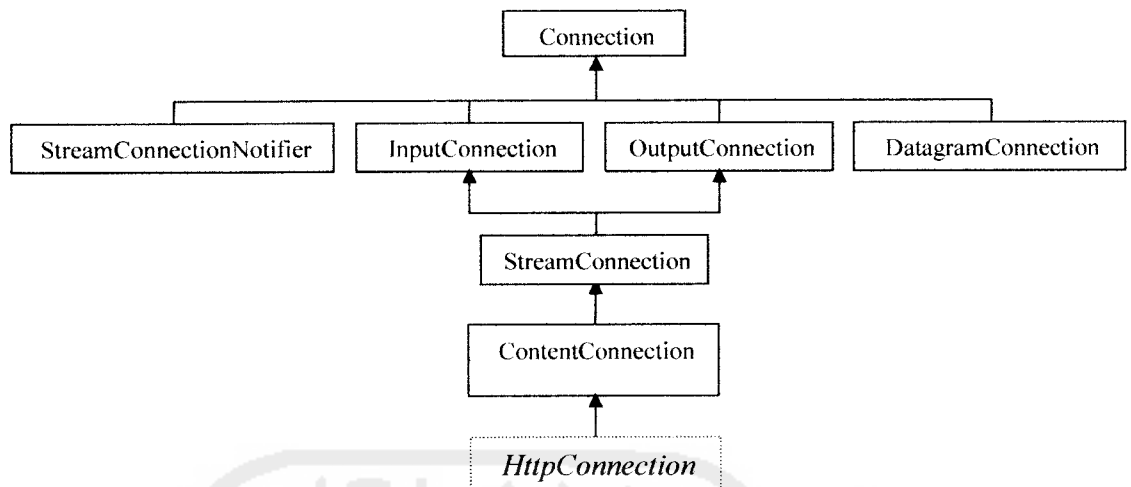
Subinterfaces dari *StreamConnection* dan menambahkan fasilitas pengolahan data. *Interface HttpConnection*, bagian dari spesifikasi MIDP, dikembangkan dari *interface* ini (*ContentConnection*) dengan menambahkan fasilitas pengolahan data yang mendukung penggunaan protokol HTTP.

6. *Interface DatagramConnection*

Subinterfaces dari *Connection* yang menyediakan komunikasi data menggunakan paket data (*datagram*).

7. *Interface StreamConnectionNotifier*

Subinterfaces dari *Connection* menyediakan fasilitas pembuatan *server-side socket*.

Gambar 2.3 Hirarki *Interface Connection*

Desain koneksi sebagai *interface* bertujuan memberi *device manufacturers* keleluasaan dalam memilih protokol komunikasi data yang akan diimplementasi, dengan tetap menyesuaikan ketentuan yang diatur dalam suatu *Profile*. Ada beberapa macam protokol yang dapat digunakan untuk melakukan komunikasi data, yaitu:

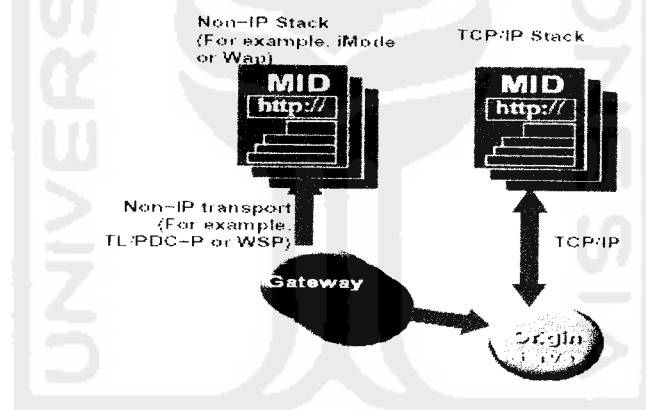
Tabel 2.2 Jenis-jenis protokol dan tipe koneksinya

Protokol	Tipe Koneksi
<i>File</i>	<i>File I/O</i>
Com.	Koneksi melalui port serial
Socket	Koneksi melalui soket TCP/IP
Datagram	Koneksi datagram
http	Koneksi dengan protokol HTTP

Dari berbagai macam protokol yang ada MIDP menggunakan HTTP sebagai protokolnya, karena HTTP dapat diimplementasikan dengan mudah pada

jaringan tanpa kabel. HTTP dapat mengambil keuntungan dari *server side* yang telah ditempatkan pada jaringan dengan kabel, pada waktu yang sama transmisi HTTP dapat merasakan berbagai permintaan yang datang dan mengirimkannya kembali.

Protokol HTTP tidak hanya dapat digunakan pada perangkat yang mendukung protokol IP. HTTP untuk MIDP dapat diimplementasikan pada perangkat yang mendukung protokol IP seperti TCP/IP atau bukan protokol IP (*wireless* protokol) seperti WAP. Tetapi pada perangkat yang menggunakan *wireless* protokol harus memiliki komputer khusus yang disebut dengan gateway yang berfungsi sebagai “jembatan” yang menghubungkan dengan *server* di internet.



Gambar 2.4 Koneksi Jaringan HTTP

2.4 *Hypertext Transport Protocol (HTTP)*

Protokol adalah sekumpulan aturan komunikasi atau standar digunakan dalam pertukaran informasi. *Hypertext Transport Protocol* adalah protokol level aplikasi (*application-level protocol*) yang mendasari keberadaan sistem informasi

yang terdistribusi, dapat saling bekerjasama dan mampu menangani informasi dalam berbagai media [FIE 99].

HTTP terdiri dari HTTP *Request* dan HTTP *Response*.

1. HTTP *Request*

Suatu HTTP *Request* terdiri dari tiga bagian, yaitu:

a. Bagian HTTP *Request* diikuti karakter CR LF

Bagian ini mendefinisikan *Request Method*, alamat (*path*) informasi yang hendak diakses, parameter tambahan (*query parameter*) dan versi protokol yang digunakan.

Spesifikasi HTTP 1.1 mendefinisikan tujuh *Request Method* (*OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE*) sementara spesifikasi MIDP 1.0 hanya mendefinisikan tiga *Request Method* (*GET, HEAD, POST*).

Contoh:

```
GET http://www.w3.org/pub/WWW/TheProject.html HTTP/1.1
GET /pub/WWW/TheProject.html HTTP/1.1
Host: www.w3.org
```

b. Bagian *Request Header* diikuti baris kosong, bersifat optional.

c. Bagian *Message Body* berisi data, bersifat optional.

2. HTTP *Response*

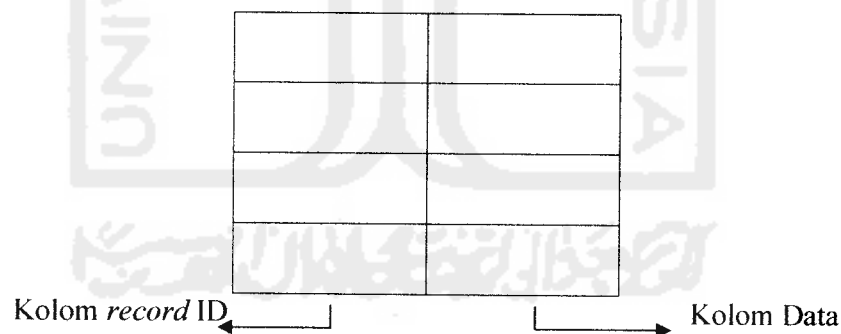
HTTP *Response* juga terdiri dari tiga bagian, yaitu:

a. Baris HTTP *Response* diikuti karakter CR LF.

- b. Bagian *Response Header*, masing-masing baris *Response Header* diikuti CR LF. Bagian ini diikuti oleh satu baris kosong untuk memisahkan bagian ini dengan bagian *Message Body*.
- c. Bagian *Message Body*, berisi data. Format data dijelaskan melalui *Content-Type Response Header*.

2.5 *Records Management System (RMS)*

Records management system mengimplementasikan *database* sederhana yang digunakan pada MIDP yang berorientasi *record*. RMS menyediakan operasi standar pada *database* seperti menambah, mengubah, menghapus serta manipulasi lainnya. *Database* RMS disebut dengan *recordstore* yang menyimpan informasi dalam bentuk *record*. Informasi yang disimpan berupa *record ID* dan data dalam bentuk byte. *Recordstore* memiliki nama yang unik dengan panjang karakter maksimal 32 karakter dan bersifat *case-sensitive*.



Gambar 2.5 Struktur *Record Store*

2.6 *Handphone*

Generasi pertama telepon selular memakai sistem analog yang dikenal dengan AMPS (*Advanced Mobile Phone Service*). Versi lainnya yang telah menggabungkan dengan teknologi digital adalah NAMPS (*Narrowband Advanced Mobile Phone Service*). AMPS dan NAMPS berkerja pada frekuensi 800MHz. Kualitas suara yang kurang bagus serta beberapa permasalahan teknis seperti sinyal analog tidak dapat dilakukan kompresi serta dimanipulasi. Hal-hal tersebut menjadi kendala untuk pengembangan sistem ini.

Generasi selanjutnya telepon selular memakai sistem digital, penggunaan sinyal digital dinilai lebih menguntungkan karena kemampuannya untuk mengkonversi dari suara menjadi informasi biner dan melakukan kompresi data. Salah satu teknologi yang menerapkan sistem digital adalah GSM (*Global System for Mobile Communication*) yang telah dipakai oleh hampir seluruh negara didunia. Teknologi ini dioperasikan pada 900MHz dan 1800MHz. Layanan yang diberikan lebih baik dari generasi sebelumnya. Suara yang dihasilkan lebih baik, akses internet secara *wireless* dan lain lain.

Perkembangan teknologi telepon selular telah mencapai pada generasi ke 3 yang sering disebut dengan 3-G, dimana generasi ini telah merambah akses secara permanen ke *web*, video interaktif serta kualitas suara yang sangat baik hingga teknologi kamera video yang diintegrasikan dalam telepon selular.

Salah satu dampak dari perkembangan teknologi telepon selular adalah masukan bahasa pemrograman Java yang aplikasinya dapat digunakan pada telepon selular. Dengan teknologi Java, pengguna ponsel diberikan pilihan untuk

men-download aplikasi pada *handphone* sesuai dengan kebutuhan. Aplikasi-aplikasi tersebut dapat berupa *game*, *organizer*, kalkulator dan lain lain.

Sun Microsystem telah membuat standarisasi *hardware* dan *software* untuk *handphone* yang dapat menjalankan aplikasi – aplikasi yang dibuat dengan bahasa pemrograman Java. *Handphone* yang bisa mendukung teknologi Java harus memiliki kriteria minimum *hardware* sebagai berikut [GIG01]:

1. *Memory*
 - a. 128 KB *memory non-volatile* untuk komponen MIDP
 - b. 32 KB *memory volatile* untuk *virtual machine runtime*
 - c. 8 KB *memory non-volatile* untuk data yang disimpan aplikasi.
2. Layar
 - a. Ukuran layar: 96 x 54
 - b. *Monochrome*
 - c. Bentuk *pixel* bujursangkar
3. Piranti Masukan (minimal satu)
 - a. *Keypad* satu tangan
 - b. *Keypad* dua tangan
 - c. Layar sentuh
4. Komunikasi Data

Dua arah, nirkabel, mungkin tidak tetap (*intermittent*), dengan *bandwidth* terbatas.

Untuk kebutuhan minimum *software* sebagai berikut [GIG01]:

1. Memiliki kernel untuk mengatur dasar perangkat keras dan dapat menjalankan Java Virtual Machine.
2. Memiliki kemampuan untuk membaca dan menulis non volatile memory untuk mendukung MIDP *persistence*.
3. Memiliki akses untuk jaringan nirkabel untuk mendukung MIDP *networking*.
4. Memiliki kemampuan untuk memanipulasi grafik pada layar.

2.7 *Multipurpose Internet Mail Extensions (MIME)*

MIME berfungsi untuk mempermudah pengguna mengirim bermacam-macam tipe *file* dengan menggunakan *message transfer system* yang sama dengan pengiriman teks. MIME memuat tata cara untuk pengkodean *file* biner, gambar, suara dan bentuk lainnya menjadi pesan berformat teks yang akan dikirim melalui keterbatasan 7 bit ASCII dari SMTP. Selain pengkodean, MIME juga menambahkan sejumlah *field* di *header* isi pesan untuk memperjelas isi dari pesan ke pengguna. Berikut contoh *header* MIME dalam pengiriman tabel [FIE96].

Tabel 2.3 Header pada MIME

<i>Header</i>	Deskripsi
<i>MIME-Version</i>	Mendefinisikan versi MIME yang digunakan
<i>Content-Description</i>	Diskripsi singkat dari isi pesan
<i>Content-ID</i>	Identitas unik
<i>Content-Type</i>	Mendefinisikan tipe dari isi pesan
<i>Content-Transfer-Encoding</i>	Sintaks pengiriman yang digunakan
<i>Content-Length</i>	Jumlah <i>bytes</i> pada isi pesan

Content-Type pada *header* MIME memiliki tipe dan subtype, sebagai berikut:

Tabel 2.4 Tipe dan subtype pada *header* MIME

Tipe	Subtipe	Diskripsi
<i>Text</i>	<i>Plain</i>	<i>Unformatted</i> ASCII
	<i>Richtext</i>	Formatted text based HTML
<i>Image</i>	<i>GIF</i>	Gambar digital GIF
	<i>JPEG</i>	Gambar digital JPEG
<i>Audio</i>	<i>Basic</i>	Suara digital
<i>Video</i>	<i>MPEG</i>	Video digital
<i>Application</i>	<i>Octet-stream</i>	<i>A string of bytes</i>
	<i>Postscript</i>	Dokumen pada <i>postscript</i>
<i>Message</i>	<i>Rfc822</i>	Pesan MIME lainnya
	<i>Partial</i>	Bagian dari pesan yang ada
	<i>External-body</i>	Penunjuk dimana isi pesan dapat diakses
<i>Multipart</i>	<i>Mixed</i>	Setiap bagian memiliki isi yang berbeda
	<i>Alternative</i>	Setiap bagian memiliki isi yang sama tetapi subtype berbeda
	<i>Pararel</i> <i>Digest</i>	Simulasi keluaran Banyak pesan

Setiap kategori terdiri dua bagian yaitu *type* dan *subtype*. Tipe informasi dalam ini isi pesan merupakan gabungan antara *type* dan *subtype* yang dihubungkan dengan tanda / . Contoh:

Content-Type: Text/Plain

Content-Type: Image/GIF

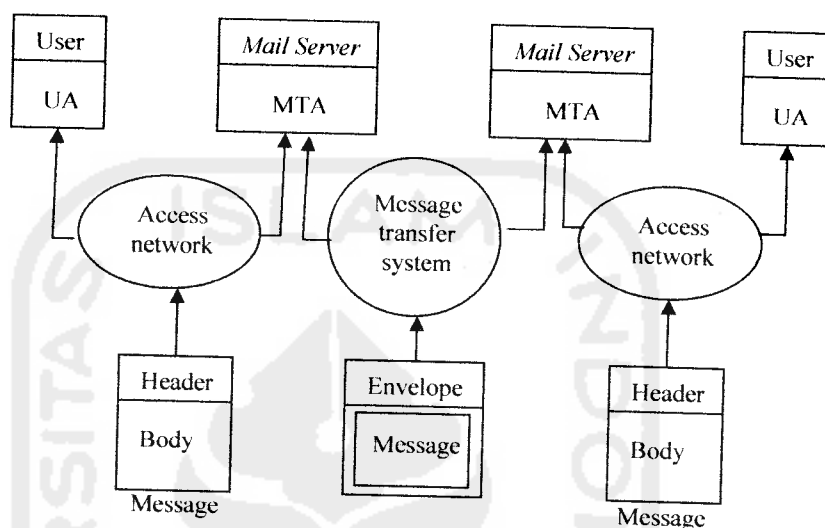
2.8 *Electronic Mail*

Email atau *electronic mail* merupakan cara pengiriman surat melalui internet. *Email* merupakan jenis layanan internet yang populer. Melalui *email* pengguna dapat mengirimkan berita, gambar atau data kepada rekan-rekannya atau menjawab surat-surat yang ditujukannya. Jika ingin menggunakan fasilitas yang ditawarkan oleh *email*, kita harus memiliki alamat *email*. Untuk mendapatkannya, pengguna harus menghubungi penyedia layanan *email* terlebih dahulu. Berbagai layanan *email* gratis telah disediakan oleh portal-portal *website* seperti www.yahoo.com, www.hotmail.com, www.bolohmail.com, dan lain – lain.

Email server adalah komputer *server* yang mengurus segala keperluan dalam kotak pos (*mailbox*) untuk semua pengguna/*clients* yang telah terdaftar. *Server* memiliki *software* yang dikenal dengan *UA server* untuk melakukan interaksi dengan *UA software* yang ada pada tiap *client* dan juga *software* untuk mengatur pertukaran *email* pada internet yang dinamakan MTA (*Message Transfer Agent*). MTA akan mengatur pengiriman dan penerimaan *email* dari *email server* lainnya. Protokol yang digunakan untuk menghubungkan antara dua MTA pada internet disebut SMTP (*Simple Mail Transfer Protocol*).

Dalam mengirimkan *email* terdapat dua prosedur, pertama mengisi data yang diperlukan seperti alamat *email* pengirim, alamat *email* penerima dan pada *head of the message*. Kedua pembungkusan kedalam “amplop” yang berisikan

data tersebut dan dikirimkan melalui jaringan dengan *message transfer system*. Alamat *email* yang terdapat pada *head of the message* memiliki standarisasi struktur maka akan dapat dibuka dan dikenali secara langsung oleh *message transfer system* [FIE96].



Gambar 2.6 Alur Kerja Email

Pada gambar di atas dapat dilihat *email* yang dikirim melalui jaringan, terdiri dari *envelope* dan *message*. *Envelope* berisi alamat *email* pengirim dan alamat *email* penerima. Alamat *email* memiliki format nama-pemakai@nama-mailserver dimana nama *mailserver* adalah nama DNS dari *mail server* dan nama-pemakai merupakan nama yang dipilih sebagai identitas pada waktu pemakai mendaftar di *mail server*. *Message* (pesan) berisikan *header* dan *body*, yang berisikan data – data yang dimasukkan oleh pengirim. *Header* terdiri dari beberapa kolom yang memiliki format standar dan dipisahkan oleh baris kosong dari isi pesan. Kolom yang terdapat pada *header* sebagai berikut [FIE99]:

Tabel 2.5 Message Header pada message transfer system

<i>From</i>	Alamat <i>email</i> yang membuat <i>email</i>
<i>To</i>	Alamat <i>email</i> yang dituju primer
<i>Cc</i>	Alamat <i>email</i> yang dituju sekunder
<i>Received</i>	Rute melewati <i>message transfer system</i>
<i>Return-Path</i>	Nama MTA terakhir

Tabel 2.6 *Message Header* pada UA / User

<i>Sender</i>	Alamat <i>email</i> pengirim
<i>Date</i>	Tanggal dan waktu <i>email</i> dikirim oleh UA
<i>Message Id</i>	Tanda unik untuk <i>email</i> oleh UA
<i>Reply - To</i>	Alamat <i>email</i> balasan
<i>Subject</i>	Judul dari <i>email</i>

Isi pesan hanya dapat berisi teks ASCII dengan panjang maksimum 1000 karakter. Karena penggunaan internet yang makin meluas maka permintaan akan isi message bertambah, seperti pengiriman pesan yang berisi gambar, *audio* dan video serta dalam bahasa yang berbeda. Untuk mengatasi hal tersebut maka digunakan MIME dalam pengiriman *email*.

BAB III

ANALISIS KEBUTUHAN PERANGKAT LUNAK

3.1 Metode Analisis

Analisis suatu sistem merupakan salah satu proses yang harus dilakukan dalam perancangan dan implementasi perangkat lunak, untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diinginkan sehingga dapat diusulkan perbaikan-perbaikannya. Tahap analisis merupakan tahap yang paling penting karena kesalahan didalam tahapan ini akan menyebabkan kesalahan pada tahap selanjutnya. Oleh sebab itu dibutuhkan suatu metode yang dapat digunakan sebagai pedoman dalam pengembangan sistem informasi.

Dalam pengembangan sebuah sistem yang akan dibuat, sistem yang telah ada sebelumnya dapat dijadikan sebagai dasar pengembangan sistem selanjutnya. Metode analisis yang digunakan adalah metode analisis secara terstruktur dengan membuat diagram arus data yang mendeskripsikan *input*, proses, dan output berdasarkan aliran informasi yang telah ada pada sistem sebelumnya. Metode ini akan menghasilkan sistem yang lebih terstruktur karena memberikan gambaran secara terperinci mengenai komponen-komponen yang digunakan oleh sistem sehingga akan memberikan sejumlah pandangan dari sudut yang berbeda terhadap sistem.

3.2 Hasil Analisis

3.2.1 Analisis Data Masukan

Berdasarkan hasil analisis kerja yang dilakukan, data masukan yang dibutuhkan untuk sistem pengaksesan *email* melalui *handphone* ini berupa data alamat *email* yang dituju, konfigurasi alamat *server*, berita atau surat yang akan disampaikan atau dikirim.

3.2.2 Analisis Proses

Ada 2 proses utama dalam perangkat lunak email client, yaitu :

1. Proses baca email
 - a. Membuka koneksi dengan *server*
 - b. Mengirimkan data yang ada, yaitu nama dan *password* ke *email server* untuk melakukan autentikasi. Jika benar proses cek *email* dilanjutkan, jika salah, *server* akan mengirimkan pesan kesalahan ke *client*
 - c. *Server* mengecek data yang diterima, yaitu alamat *email server*, *user name* dan *password*.
 - d. Jika autentikasi benar, proses selanjutnya mengambil data *email* pada *email server*. Data yang di ambil berupa judul, pengirim, tanggal, ID *email*.
2. Proses kirim *email*
 - a. Membuka koneksi dengan *server*.

- b. Mengirimkan data yang ada, yaitu nama dan *password* ke *email server* untuk melakukan autentikasi. Jika benar proses dilanjutkan, jika salah, *server* akan mengirimkan pesan kesalahan ke *client*.
- c. *Server* mengecek data yang diterima, yaitu alamat *email server*, *user name* dan *password*.
- d. Jika autentikasi benar, proses selanjutnya mengirimkan *email* ke alamat *email* tujuan.

3.2.3 Analisis Data Keluaran

Dalam proses baca email, Sistem dapat menampilkan data-data header *email* berupa subjek yang telah di-*download* dari *server*. Isi *email* dapat dilihat setelah memilih subyek *email* yang akan dibaca. Untuk proses kirim email system dapat mengirimkan email ke alamat email yang dituju.

3.2.4 Analisis Kebutuhan Antar muka

Kebutuhan terhadap antar muka (*interface*) yang akan dibuat bersifat *user friendly*, dengan tujuan agar program yang telah dibangun dapat digunakan dengan mudah oleh pengguna dan dapat memudahkan pengguna dalam mengaksesnya. Karena program yang akan dibangun ini dimaksudkan untuk diakses menggunakan *handphone*, maka antar muka yang digunakan sama persis dengan antar muka yang ada pada *handphone*.

3.2.5 Kebutuhan Sistem

Persyaratan standar yang harus dipenuhi untuk menjalankan sistem aplikasi tersebut, adalah sebagai berikut:

1. Sistem operasi Microsoft Windows 9x, NT, XP, Unix, Linux, Mac Os.
2. Prosesor Intel, AMD.
3. Ram 512 Mb.
4. Monitor VGA
5. Netbeans berbasis J2SDK 1.5.
6. *Handphone* yang berbasis java, seperti Nokia 9210 Communicator, Motorola i 85 s, Siemens M50, Simens M55, Sony Ericsson T 610.
7. *Data Cable/Memory Card Reader* untuk *handphone* yang digunakan

3.2.6 Kebutuhan Bahasa Pemrograman.

Dalam membangun sistem Pengaksesasn *email* melalui *handphone* ini menggunakan aplikasi Java, karena Aplikasi Java adalah program Java mandiri yang bisa dijalankan oleh interpreter Java melalui baris perintah (*command line*).

Java adalah sebuah bahasa berorientasi objek sederhana yang memiliki banyak kesamaan komponen dengan C dan C++, tetapi telah membuang bagian yang selama ini menyulitkan *programmer*. Java memiliki kelebihan yang ada di sebuah bahasa intepreter serta kinerja dari suatu kode yang terkompilasi. Dalam waktu yang relatif singkat, para pengguna bahasa pembuat program aplikasi dan *script*, seperti Visual Basic, Delphi, Perl, HyperCard, atau AppleScript akan segera dapat memulai

pemrograman Java. Sementara itu, pemrogram C dan C++ yang sudah lebih berpengalaman kan dapat dengan mudah mentransfer kemampuan mereka ke dalam bahasa Java ini. Sebagian kode besar kode serta algoritma yang sudah mereka buat pun tetap dapat digunakan. Java sangat cocok untuk tugas-tugas yang berkaitan dengan internet. Sun membuat Java ditujukan untuk digunakan dalam berbagai aplikasi berbasis *web*.



BAB IV

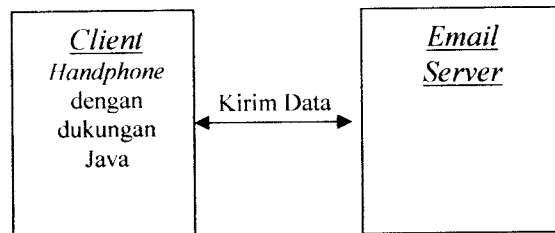
PERANCANGAN PERANGKAT LUNAK

4.1. Metode Perancangan

Metode perancangan yang digunakan untuk membangun aplikasi ini adalah metode perancangan terstruktur (*Structure Design Method*) dengan menggunakan diagram arus data (*Data Flow Diagram*) dengan metode pendekatan atas bawah (*top down approach*). Perancangan dimulai dari perancangan sistem secara umum (*general approach*) disebut juga desain konseptual (*conceptual design*) atau desain logika (*logical design*), bertujuan untuk membuat sistem yang baru secara logis dan konseptual. Hasil dari tahap ini adalah suatu bentuk esensial model, yaitu apa yang harus dilakukan oleh sistem untuk memenuhi kebutuhan pemakai dan menggambarkan bagaimana sistem akan diimplementasikan, hasil dari tahap ini masih dalam bentuk global. Kemudian perancangan sistem masih dilanjutkan ke perancangan sistem perancangan terinci atau detil disebut juga dengan perancangan sistem secara fisik (*physical system design*) atau desain internal (*internal design*), tahap ini merupakan tahap dari kelanjutan sistem global, dimana model perancangan sistem baru tidak hanya sebatas konseptual saja, melainkan sudah dijabarkan kedalam bentuk-bentuk yang lebih detil dan mendekati tahap implementasi.

4.2 Hasil Perancangan

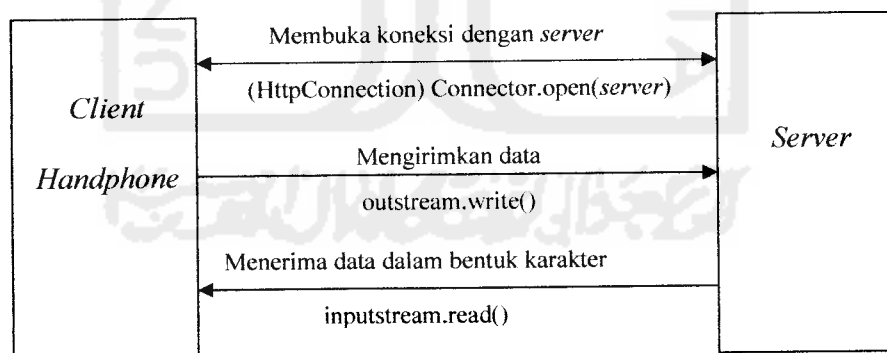
4.2.1 Prinsip Kerja Sistem



Gambar 4.1 Alur kerja program

Client merupakan *handphone* yang mendukung teknologi Java. Aplikasi pada *client* meliputi pemasukan data, penyimpanan data dan mengirimkan data ke *server*. Data akan dikirim ke *server* dimana pada *server* merupakan *server* POP3 jika melakukan proses *download email* dan *server* SMTP jika melakukan proses kirim *email*. Pada *server*, *script* yang ada akan melakukan pengecekan data yang masuk jika semua data sudah benar maka akan diteruskan, sebaliknya jika ada data yang tidak benar maka *server* akan memberitahukan kesalahan pada *client*.

4.2.2 Algoritma Sistem



Gambar 4.2 Koneksi antara *client* dan *server*

Ada 2 proses dalam system ini, yaitu : pengiriman *email* dan baca *email*.

Untuk baca *email*, proses yang terjadi pada system:

1. *Client* membuka koneksi dengan *server*.
2. *Client* mengirimkan perintah untuk mengunduh data *email*.
3. *Client* menerima data dalam bentuk karakter.

Untuk kirim email, proses yang terjadi pada system:

1. *Client* membuka koneksi dengan *server*.
2. *Client* mengirimkan perintah untuk mengirim *email*.
3. *Server* mengirimkan *email* ke *email server* tujuan.

4.2.2.1 Prinsip Kerja Aplikasi

1. Pada *Client*

a. Menampilkan menu pilihan, yaitu

- *Inbox, form* ini digunakan untuk melakukan proses *download*, membaca, menghapus dan, meneruskan *email* baik pada *server* maupun *client*.
- *Kirim Email, form* untuk mengisikan data yang diperlukan untuk keperluan pengiriman *email*. Data tersebut adalah alamat *email* tujuan, judul *email*, isi *email* dan pengirim serta dapat menyimpan data tersebut pada *record*.
- *Daftar Email, form* untuk mengisikan nama, alamat *email* dan menyimpannya pada *record*.
- *Konfigurasi, form* konfigurasi alamat *email* dan *server*.
- *About, form* informasi tentang pembuat program.

- Keluar, untuk keluar dari program.

2. Pada *Server*

Belum ada proses yang terjadi

4.2.2.2 Pengolahan data pada RMS

1. Pada *Client*

Untuk melakukan pengolahan data pada RMS diperlukan *recordstore*. Pada satu *recordstore* bisa terdapat beberapa *database*. Aplikasi ini memerlukan 2 *database* untuk menyimpan data. *Database* untuk menyimpan daftar *email* dan *database* untuk menyimpan data konfigurasi.

a. Penyimpanan Data

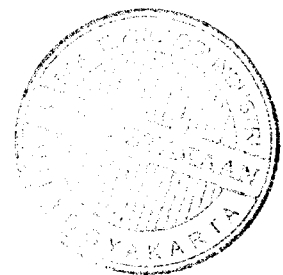
- Buka *recordstore*.
- Mengubah data dalam bentuk *bytearray*.
- Menyimpan data dengan *recordID* baru.
- Tutup *recordstore*.

b. Penghapusan Data

- Buka *recordstore*.
- Masukkan data yang akan dihapus.
- Pencarian data pada *recordstore*.
- Hapus data pada *recordstore*.
- Tutup *recordstore*

c. Pengeditan Data

- Buka *recordstore*.
- Masukkan data yang akan diubah.



- Pencarian data pada *recordstore*.
- Masukan data baru.
- Menyimpan data baru pada *recordID* data yang diubah.
- Tutup *recordstore*.

2. Pada *Server*

Belum ada proses yang terjadi.

4.2.2.3 Pengiriman *email*

1. Pada *Client*

- a. Membuka koneksi dengan *server*
- b. Mengirimkan data yang ada ke *server*
- c. Menerima balasan dari *server*

2. Pada *Server*

- a. Mengecek data yang diterima, yaitu alamat *email server*, *user name* dan *password*.
- b. Mengirimkan nama dan password ke *email server* untuk melakukan autentikasi. Jika sah proses dilanjutkan, jika tidak sah kirim pesan kesalahan ke *client*.
- c. Jika autentikasi sah, proses selanjutnya mengirimkan *email* ke alamat *email* tujuan.

4.2.2.4 Cek *email*

Pengecekan *email* digunakan untuk melihat *email* yang ada pada *inbox*.

1. Pada *Client*

- a. Membuka koneksi dengan *server*.

- b. Mengirimkan data yang ada ke *server*.
- c. Menerima balasan dari *server*.
- d. Memilih judul *email* yang akan dibaca.
- e. Mengirimkan ID *email* yang akan baca.
- f. Membaca *email*.

2. Pada *Server*

- a. Mengecek data yang diterima, yaitu alamat *email server*, *user name* dan *password*.
- b. Mengirimkan nama dan password ke *email server* untuk melakukan autentikasi. Jika sah proses dilanjutkan, jika tidak sah kirim pesan kesalahan ke *client*.
- c. Jika autentikasi sah, proses selanjutnya mengambil data *email* pada *email server*. Data yang di ambil berupa judul, pengirim, tanggal, ID *email*
- d. Menerima ID *email* dari *client*, kemudian diteruskan untuk mengambil data *email* pada *email server*.

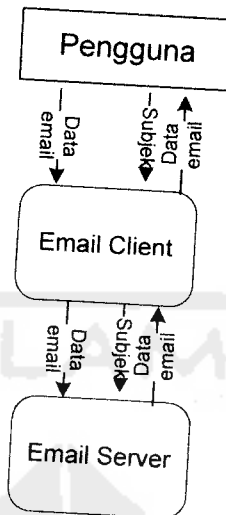
4.2.3 Diagram Arus Data

Tahapan dari pembuatan diagram arus data pada Sistem *Email Client* Menggunakan *Handphone*, adalah sebagai berikut:

1. Diagram Konteks

Diagram konteks digunakan untuk menggambarkan hubungan *input/output* antara sistem dengan dunia luarnya (kesatuan luar), suatu diagram konteks

sistem selalu mengandung satu proses saja yang mewakili proses dari seluruh sistem, seperti yang ditampilkan pada gambar 4.3.



Gambar 4.3 Diagram Konteks Pengaksesan *Email* Melalui *Handphone*

Pada gambar 4.3 ini merupakan gambaran dari seluruh sistem secara umum, dimana pengguna bisa berhubungan dengan sistem untuk memasukan data *email*.

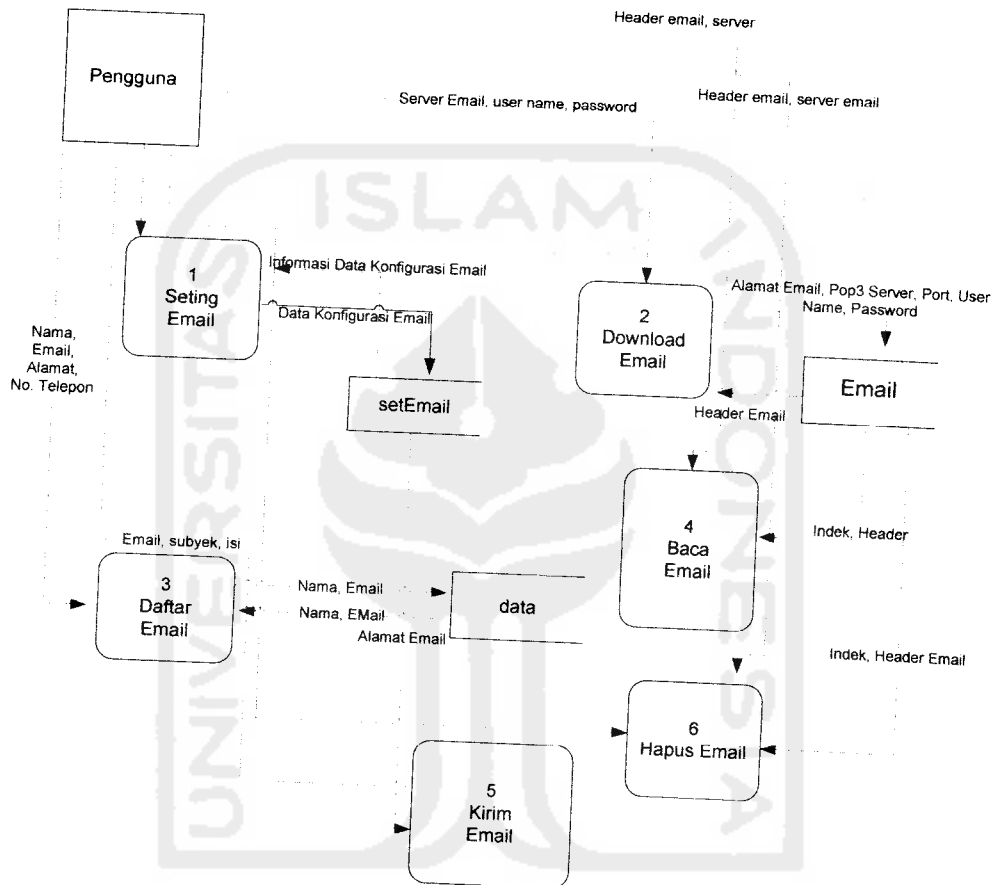
2. Diagram Arus Data (*Data Flow Diagram/DFD*) Level 1

Proses pada diagram arus data level ini dibagi menjadi 6 proses, yaitu operasi memasukan konfigurasi, *download email*, *baca email*, *hapus email*, menampilkan data *email* dan kirim *email* seperti pada gambar 4.4.

1. Konfigurasi digunakan untuk konfigurasi *email* pada *email client*
2. *Download Email* digunakan untuk men-*download email* dari *email server* ke *handphone*
3. *Baca Email* digunakan untuk membaca *email* yang telah di-*download*
4. *Hapus Email* digunakan untuk menghapus *email* yang ada pada mail *server*

5. Daftar *Email* digunakan untuk menambah, merubah maupun menghapus daftar *email* pada *handphone*.

6. Kirim *Email* digunakan untuk pengiriman *email*

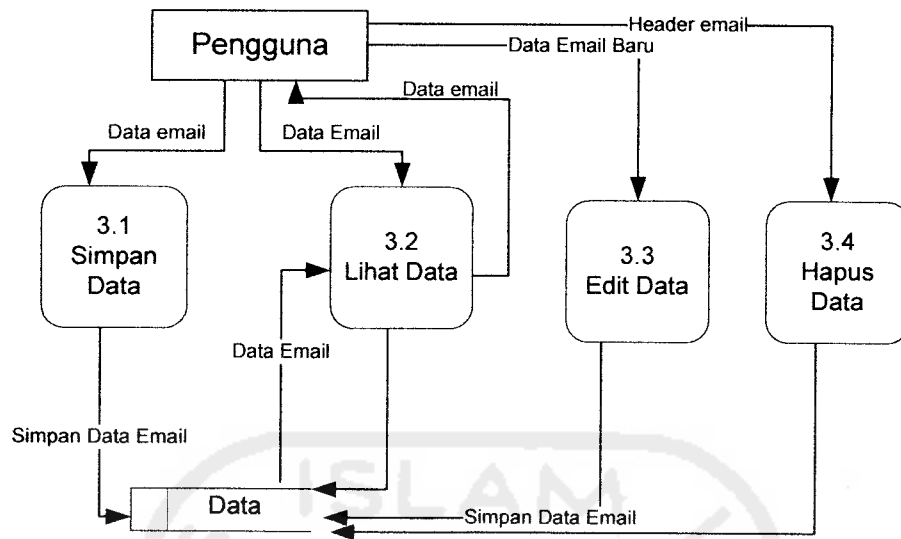


Gambar 4.4 DFD Level 1 Pengaksesan *Email* Melalui *Handphone*

3. Diagram Arus Data (*Data Flow Diagram/DFD*) Level 2.

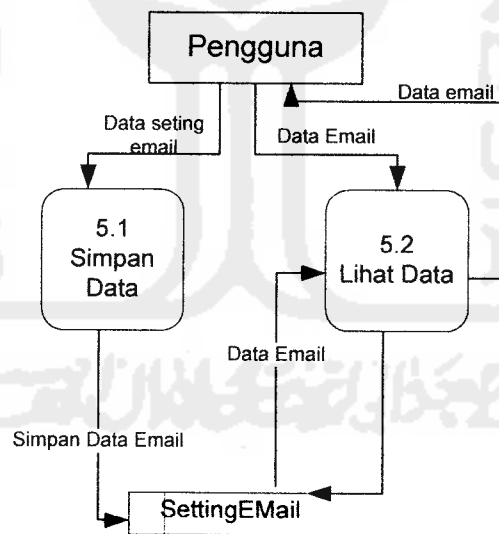
a. Proses Daftar *Email*

Di dalam Daftar *Email* terdiri dari operasi tambah, hapus, edit data *email* tabel data.

Gambar 4.5 DFD Level 2 Proses Daftar *Email*

b. Proses Konfigurasi Email

Di dalam konfigurasi terdiri dari operasi lihat, ubah dan simpan pada tabel *Setting Email*.



Gambar 4.6 DFD Level 2 Proses Konfigurasi Email

4.2.4 Perancangan Program.

Salah satu kriteria yang harus dimiliki oleh perangkat lunak untuk mendapatkan predikat *User friendly* adalah bahwa perangkat lunak itu mempunyai antarmuka yang bagus, mudah dioperasikan, dan pengguna selalu merasa senang untuk menggunakan perangkat lunak tersebut [SAN 97].

Program aplikasi ini terdiri dari dua bagian penting. Bagian pertama adalah bagian antarmuka yang berfungsi sebagai sarana dialog antara manusia dan komputer. Bagian kedua adalah bagian aplikasi yang merupakan bagian yang berfungsi untuk menghasilkan informasi berdasarkan olahan data yang dimasukkan pengguna melalui algoritma yang disyaratkan oleh aplikasi ini.

4.2.5 Perancangan Bagian Antarmuka

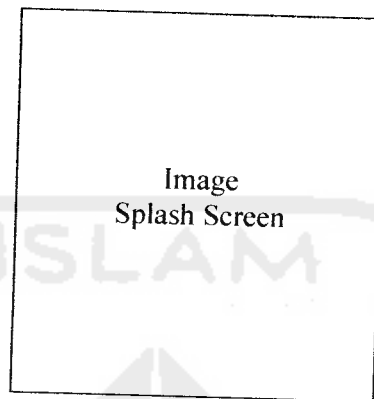
Program aplikasi pengaksesan *email* melalui *handphone* ini terdiri dari dua bagian penting. Bagian pertama adalah bagian antarmuka yang berfungsi sebagai sarana dialog antara manusia dan komputer. Bagian kedua adalah bagian aplikasi yang merupakan bagian yang berfungsi untuk menghasilkan informasi berdasarkan olahan data yang dimasukkan pengguna melalui algoritma yang disyaratkan oleh aplikasi ini.

4.2.5.1 Perancangan Tampilan Antarmuka

Rancangan antar muka dalam sistem pengaksesan *email* melalui *handphone* ini terdiri dari:

1. Rancangan Antarmuka Tampilan Awal

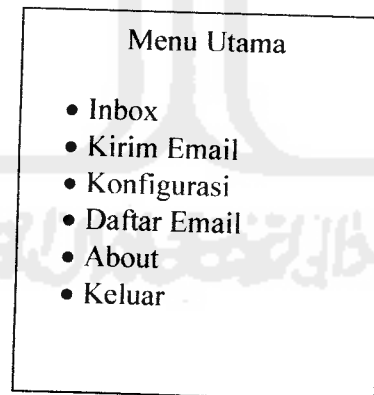
Tampilan awal sebagai pembuka, *splash screen* ini muncul selama program dijalankan.



Gambar 4.7 *Form* Tampilan awal

2. Rancangan Antarmuka Menu Utama

Pada menu utama tersedia pilihan untuk melakukan perintah selanjutnya.



Gambar 4.8 *Form* Menu Utama

3. Rancangan Antarmuka *Inbox*

Pada menu ini diberikan tiga pilihan untuk langkah selanjutnya, yaitu kirim *email* untuk mengirimkan *email*, ambil data untuk mengambil data yang sudah tersimpan dan cek *email* untuk mengecek inbox *email*.

```

graph TD
    subgraph Inbox
        subgraph MenuBox [Menu]
            M1[1. Download Email]
            M2[2. Baca Email]
            M3[3. Hapus Email]
            M4[4. Kembali ke menu]
        end
        B[Menu]
    end
  
```

Gambar 4.9 *Form Menu Inbox*

4. Rancangan Antarmuka *Baca Email*

Setelah memilih *download* pada *form* inbox, maka informasi yang ditampilkan adalah tanggal, judul, pengirim dan isi *email* serta informasi ada atau tidak ada file attachment pada *email* tersebut. Dalam *form* ini terdapat menu pilihan *Balas email*, *Forward* dan kembali ke menu.

```

graph TD
    subgraph BacaEmail [Baca Email]
        P[Pengirim dan Subyek :  
Dari :  
Subyek :  
Tanggal :]
        I[Isi Pesan :]
        subgraph MenuBox [Menu]
            M1[1. Kembali Ke Inbox]
            M2[2. Balas Email]
            M3[3. Forward]
            M4[4. Kembali ke menu]
        end
    end
    B[Menu]
  
```

Gambar 4.10 *Form Baca Email*

5. Rancangan Antarmuka Balas *Email*

Form balas *email* sama dengan *form* kirim *email*, tetapi pada *form* ini alamat *email* tujuan otomatis terisi dari alamat *email* yang sebelumnya dibaca. Didalam *form* ini terdapat menu pilihan; edit, kirim *email* dan kembali ke menu.

Kirim Email							
Dari :	<input type="text"/>						
Kepada :	<input type="text"/>						
CC :	<input type="text"/>						
Subyek :	<input type="text"/>						
Isi Pesan	<input type="text"/>						
<table border="1"> <thead> <tr> <th colspan="2">Menu</th> </tr> </thead> <tbody> <tr> <td>1. Kirim Email</td> <td></td> </tr> <tr> <td>2. Kembali Ke Menu</td> <td></td> </tr> </tbody> </table>		Menu		1. Kirim Email		2. Kembali Ke Menu	
Menu							
1. Kirim Email							
2. Kembali Ke Menu							
	Menu						

Gambar 4.11 *Form* Balas Email

6. Rancangan Antarmuka *Forward*

Form forward email sama dengan *form reply* dan *form* kirim *email*, tetapi pada *form* ini alamat *email* tujuan tidak terisi secara otomatis, melainkan isi dari *email* yang sebelumnya dibaca yang terisi secara otomatis. Didalam *form* ini terdapat menu pilihan edit, kirim *email* dan kembali ke menu.

Kirim Email							
Dari :	<input type="text"/>						
Kepada :	<input type="text"/>						
CC :	<input type="text"/>						
Subyek :	<input type="text"/>						
Isi Pesan	<input type="text"/>						
<table border="1"> <thead> <tr> <th colspan="2">Menu</th> </tr> </thead> <tbody> <tr> <td>1.Kirim Email</td> <td></td> </tr> <tr> <td>2.Kembali Ke Menu</td> <td></td> </tr> </tbody> </table>		Menu		1.Kirim Email		2.Kembali Ke Menu	
Menu							
1.Kirim Email							
2.Kembali Ke Menu							
Menu							

Gambar 4.12 *Form Forward*

7. Rancangan Antarmuka Kirim *Email*

Form untuk mengirimkan *email*, masukan data-data yang diperlukan. Tombol *options* untuk menampilkan pilihan, send *email* untuk mengirimkan data ke *server*, kembali ke menu sebelumnya.

Kirim Email							
Dari :	<input type="text"/>						
Kepada :	<input type="text"/>						
CC :	<input type="text"/>						
Subyek :	<input type="text"/>						
Isi Pesan	<input type="text"/>						
<table border="1"> <thead> <tr> <th colspan="2">Menu</th> </tr> </thead> <tbody> <tr> <td>1.Kirim Email</td> <td></td> </tr> <tr> <td>2.Kembali Ke Menu</td> <td></td> </tr> </tbody> </table>		Menu		1.Kirim Email		2.Kembali Ke Menu	
Menu							
1.Kirim Email							
2.Kembali Ke Menu							
Menu							

Gambar 4.13 *Form* untuk mengirimkan *email*

8. Rancangan Antarmuka Konfigurasi

Form konfigurasi digunakan untuk mengubah konfigurasi yang meliputi host alamat *server*, alamat *email server*, port , nama dan password pada *email server* yang dimilikinya. Simpan data *email* untuk menyimpan data-data yang dimasukkan dan kembali ke menu..

The image shows a web form titled "Konfigurasi Email". It contains the following elements:

- Konfigurasi Email** (Title)
- Alamat Email :** followed by a text input field.
- User Name :** followed by a text input field.
- Password:** followed by a text input field.
- POP3 Server :** followed by a text input field.
- SMTP Server** followed by a text input field.
- Security SMTP** with two radio button options:
 - Off
 - On
- SSL** with two radio button options:
 - Off
 - On
- A **Menu** button that opens a sub-menu with two options:
 - 1.Kirim Email
 - 2.Kembali Ke Menu
- A **Menu** button at the bottom right of the form.

Gambar 4.14 *Form* Konfigurasi

9. Rancangan Antarmuka Daftar *Email*

Pada *form* ini akan ditampilkan 2 pilihan. Lihat untuk menampilkan data yang ada pada *record* dan kembali ke menu.

Daftar Nama	
Kembali	Lihat

Gambar 4.15 *Form Daftar Email*

10. Rancangan Antarmuka Data *Email*

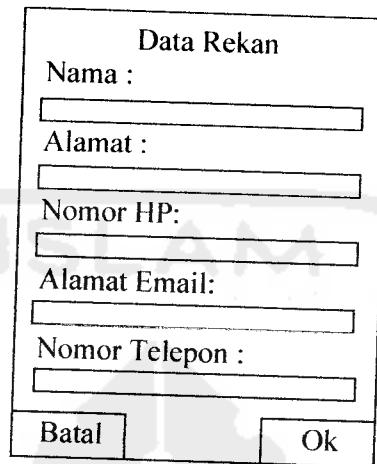
Pada *form* ini menampilkan no *record*, jumlah *record*, informasi tentang data yang tersimpan dalam *record*. Pada menu ini juga terdapat beberapa menu pilihan, yaitu ; kembali ke daftar nama, tambah data, hapus data dan kembali ke menu.

Data Email															
Record Id:															
Nama :															
Alamat Email:															
Alamat:															
Nomor Hp:															
Nomor Telepon:															
Jumlah Record:															
<table border="1"> <thead> <tr> <th colspan="2">Menu</th> </tr> </thead> <tbody> <tr> <td colspan="2">1. Kirim Email</td> </tr> <tr> <td colspan="2">2. Daftar Nama</td> </tr> <tr> <td colspan="2">3. Tambah Data</td> </tr> <tr> <td colspan="2">4. Edit Data</td> </tr> <tr> <td colspan="2">5. Hapus Data</td> </tr> <tr> <td colspan="2">6. Kembali Ke MEnu</td> </tr> </tbody> </table>		Menu		1. Kirim Email		2. Daftar Nama		3. Tambah Data		4. Edit Data		5. Hapus Data		6. Kembali Ke MEnu	
Menu															
1. Kirim Email															
2. Daftar Nama															
3. Tambah Data															
4. Edit Data															
5. Hapus Data															
6. Kembali Ke MEnu															
Menu															

Gambar 4.16 *Form Data Email*

11. Rancangan Antarmuka Tambah Data

Masukan nama dan alamat *email* pada kolom yang telah tersedia, tekan ok untuk menyimpan data pada *record* dan tekan batal untuk kembali ke *form* sebelumnya.

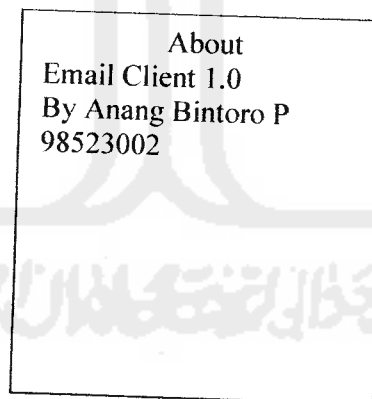


The image shows a dialog box titled "Data Rekan". It contains five text input fields, each with a label to its left: "Nama :", "Alamat :", "Nomor HP:", "Alamat Email:", and "Nomor Telepon :". At the bottom of the dialog, there are two buttons: "Batal" on the left and "Ok" on the right.

Gambar 4.17 *Form* Tambah Data

12. Rancangan Antarmuka *About*

Form about menampilkan informasi pembuat aplikasi



The image shows a dialog box titled "About". It contains the following text: "Email Client 1.0", "By Anang Bintoro P", and "98523002".

Gambar 4.18 *Form About*

BAB V

IMPLEMENTASI PERANGKAT LUNAK

Pada bab ini, program diimplementasikan untuk dapat dioperasikan pada kondisi yang sebenarnya, sehingga akan dipastikan apakah program ini dapat mencapai tujuan seperti yang diinginkan.

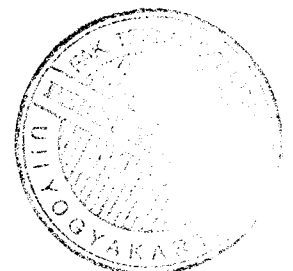
5.1 Batasan Implementasi

Perangkat lunak yang dibuat memiliki batasan-batasan dalam implementasi yang dapat mengarahkan bentuk program aplikasi dengan rancangan program. Pengembangan pembuatan perangkat lunak diasumsikan bahwa perangkat lunak yang dibuat disesuaikan dengan batasan-batasan seperti yang dijelaskan pada batasan masalah.

5.1.1 Perangkat Keras

Perangkat keras yang dipergunakan pada saat implementasi dan pembuatan program adalah:

- a. Intel Core Duo 2050 Mhz
- b. *Memory* RAM 512 MB
- c. *Hardisk* 80 GB
- d. DVD RW Combo
- e. Monitor *Ultra Wide Screen*
- f. Mouse
- g. Keyboard



h. *Handpone* Sony Ericsson W 800i

5.1.2 Perangkat Lunak

Pada implementasi ini perangkat lunak yang dipergunakan adalah

- Bahasa pemrograman Java 2 Micro Edition.
- Netbeans 5.5 dengan *Mobility Pack*.

5.2 Implementasi Perangkat Lunak

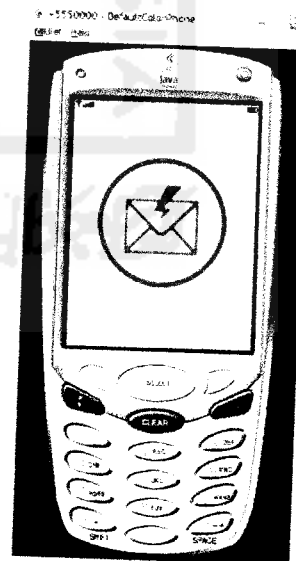
Pada bagian implemtasi perangkat lunak ini, memuat dokumentasi dan penjelasan tentang implementasi perangkat lunak yang meliputi proses-proses yang digunakan pada proses aplikasi pengaksesan *email* melalui *handphone*.

5.2.1 Tampilan Awal

Tampilan awal pada aplikasi pengaksesan *email* melalui *handphone* ini terdiri terdapat perintah *Launch* untuk menjalankan aplikasi *email client*. Apabila kita memilih untuk menjalankan aplikasi maka *splash screen* akan muncul.



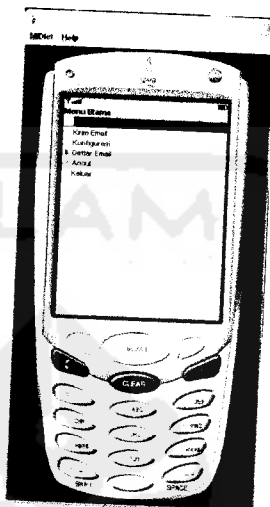
Gambar 5.1 Tampilan Awal



Gambar 5.2 Tampilan Splash Screen

5.2.2 Form Menu Utama

Form Menu Utama ini terdiri dari 5 menu pilihan, yaitu; Menu *Inbox* Kirim Email, Daftar Email, Konfigurasi, *About* dan Keluar. Form ini dapat dilihat pada gambar 5.3



Gambar 5.3 Form Menu Utama

Pada Menu Utama ini, prosedur pemilihan menu aplikasi yang digunakan adalah sebagai berikut:

```

if (displayable == frmMenu) {
    if (command == frmMenu.SELECT_COMMAND) {
        switch (get_frmMenu().getSelectedIndex()) {
            case 0:
                // Insert pre-action code here
                getDisplay().setCurrent(get_frmInbox());
                frmInbox.setTitle("POP3 Server : "+pop3Serv+",
                Port :"+pop3Port);
                // Insert post-action code here
                break;
            case 1:
                // Insert pre-action code here
                getDisplay().setCurrent(get_frmSend());
                // Insert post-action code here
                break;
            case 6:
                // Insert pre-action code here
                exitMIDlet();
                // Insert post-action code here
                break;
            case 2:
                // Insert pre-action code here
                getDisplay().setCurrent(get_frmSetting());
                //LoadKonfigurasi();
                loadSetting();
        }
    }
}

```

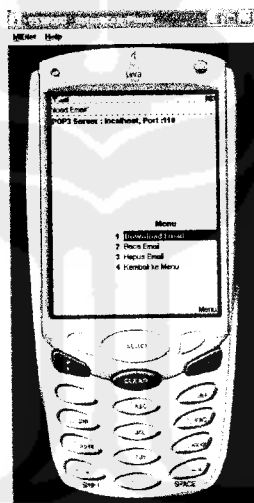
```

break;
case 5:
// Insert pre-action code here
getDisplay().setCurrent(get_frmAbout());
// Insert post-action code here
break;
case 3:
// Insert pre-action code here
getDisplay().setCurrent(get_frmData());
// Insert post-action code here
break;
case 4:
// Insert pre-action code here
getDisplay().setCurrent(get_frmHelp());
// Insert post-action code here
break;
}
}

```

5.2.3 Form Inbox

Form Inbox digunakan untuk menampilkan data pada *inbox email server*.



Gambar 5.4 *Form Menu Inbox*

Pada menu ini terdapat empat menu, *Download Email*, *Baca Email*, *Hapus Email* dan *Kembali ke Menu*.

1. *Download Email* digunakan untuk mengambil data *email* dari mail *server*.

2. Baca *Email* digunakan untuk membaca *email* yang telah di-
download.
3. Hapus *Email* digunakan untuk menghapus *email* yang ada pada
Email server.
4. Kembali ke Menu digunakan untuk kembali ke menu utama.

Prosedur untuk *download email*:

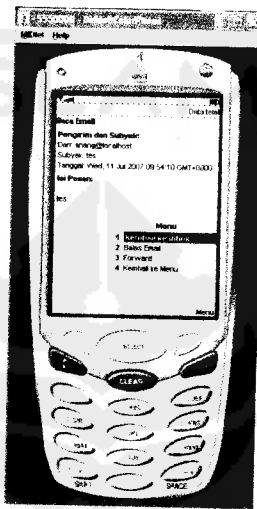
```
private void doInbox() {
    //1. Hapus list dan nomer urut email pada obyek Vector().
    get_frmInbox().deleteAll();
    noInbox.removeAllElements();
    offlineMsg.removeAllElements();
    sender.removeAllElements();
    subject.removeAllElements();
    //2. Ciptakan obyek Pop3Client.
    // Jika menggunakan IMAP, ubah constructor menjadi ImapClient()
    InboxClient inbox = new Pop3Client();
    try {
        //3. Koneksi ke Email Server
        inbox.open(pop3Serv, pop3Port, ssl, emailUser, emailPass);
        //4. Ambil Email
        for (int i = 0; i < inbox.getMessageCount(); i++) {
            String uid = inbox.getUniqueId(i);
            int size = inbox.getSize(i);
            Message msg = inbox.getMessage(i);
            get_frmInbox().append("Dari: " + msg.getHeaderValue("From", null) +
                "\r\nSubyek: " + msg.getHeaderValue("Subject", null) +
                "\r\nTanggal: " + msg.getHeaderValue("Date", null), null);
            //5. Buat nomer urut email pada obyek Vector()
            noInbox.addElement(new Integer(i));
            //6. Ambil alamat pengirim dan subyek.
            // Vector alamat pengirim dan subyek ini berguna untuk
            // membalas email.
            sender.addElement(newString(
                msg.getMachineAddress(msg.getHeaderValue("From", null)));
            subject.addElement(new
                String(msg.getHeaderValue("Subject", null)));
            //7. Simpan body email pada obyek Vector(). Proses body linanya
            //Catatan: Body dari suatu email harus di loop berdasarkan linanya
            String body = "";
            for (int j = 0; j < msg.getBodyLineCount(); j++) {
                if (body != null) {
                    body = body + "\r\n";
                }
                body = body + msg.getBodyLine(j);
            }
            offlineMsg.addElement(new String(body));
        }
    } catch (IOException e) {
        setMsg("IO Error! Pesan: " + e.getMessage(), AlertType.ERROR);
    } catch (MailException e) {
        setMsg("Mail Error! Pesan: " + e.getMessage(), AlertType.ERROR);
    } catch (Exception e) {
        setMsg("Error tak dikenal! Pesan: " + e.getMessage(), AlertType.ERROR);
    } finally {
        if (inbox.connected()) {
            try {
                inbox.close();
            }
        }
    }
}
```

```

    } catch (Exception e) {
        setMsg("Gagal menutup koneksi! Pesan: " +
            e.getMessage(), AlertType.ERROR);
    }
}
}
)

```

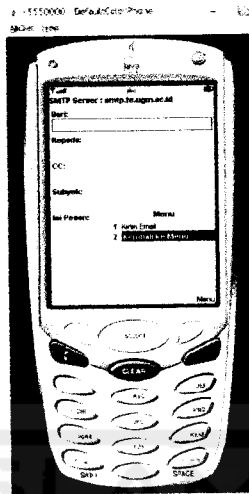
Setelah *download* dan membaca *email*, terdapat menu pilihan kembali ke *Inbox*, *Balas Email*, *Forward* dan kembali ke menu utama.



Gambar 5.5 *Form* Menu Baca *Email*

5.2.4 *Form* Menu Kirim *Email*

Form ini terdiri dari 2 menu pilihan, yaitu Kirim *Email* dan Kembali ke Menu, ini dapat dilihat pada gambar 5.6. Untuk mengirim *Email*, isi alamat *email* yang dituju, subjek *email* dan isi *email*. Setelah itu pilih Kirim *Email*, untuk membatalkan kiriman *email* tekan kembali ke menu utama.



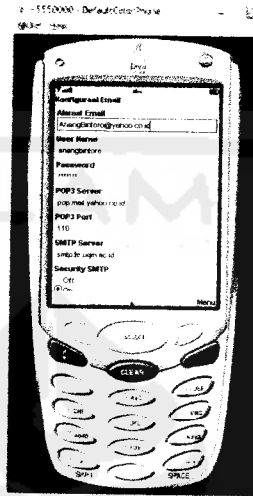
Gambar 5.6 Form Kirim Email

Prosedur yang digunakan dalam kirim *email* adalah:

```
private void doSend() {
    Message msg = new Message(alamatEmail, fldReceiver.getString(),
        fldSubject.getString());
    msg.setHeaderValue("CC", fldCC.getString());
    msg.addBodyLine(fldBody.getString());
    //1. Ciptakan obyek SMTP via koneksi socket
    SmtplibClient smtp = new SmtplibClient(localhost);
    try {
        //2. Buka koneksi SMTP. Anggap SMTP Server bisa menerima anonymous
        if (smtpPass)
            smtp.open(smtpHost, 25, ssl, emailUser, emailPass);
        else
            smtp.open(smtpHost);
        //3. Kirim email
        smtp.sendMessage(msg);
    } catch (IOException e) {
        setMsg("IO Error! Pesan: " + e.getMessage(), AlertType.ERROR);
    } catch (MailException e) {
        setMsg("Mail Error! Pesan: " + e.getMessage(), AlertType.ERROR);
    } finally {
        try {
            smtp.close();
        } catch (Exception e) {
            setMsg("Gagal menutup koneksi. Pesan: " +
                e.getMessage(), AlertType.ERROR);
        }
    }
}
}
```

5.2.5 *Form Konfigurasi*

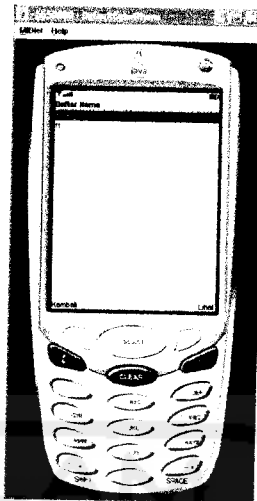
Form ini digunakan untuk mengubah konfigurasi yang meliputi host alamat *server*, alamat *email server*, *port*, nama dan *password* pada *email server* yang dimilikinya. *Form* ini dapat dilihat pada gambar 5.7.



Gambar 5.7 *Form Konfigurasi*

5.2.6 *Form Daftar Email*

Form daftar *email* menampilkan daftar nama yang terdapat pada basis data *email*. Dalam menu ini terdapat 2 pilihan menu; lihat untuk masuk ke menu data *email* dan kembali untuk kembali ke menu utama. *Form* ini dapat dilihat dalam gambar 5.8.

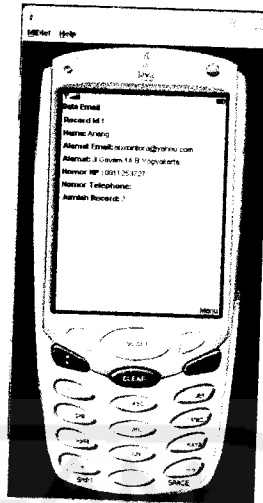


Gambar 5.8 *Form* Daftar Nama

5.2.7 *Form* Data *Email*

Form ini menampilkan informasi dari daftar nama jumlah data. *Form* ini dapat dilihat pada gambar 5.9. *Form* ini terdiri dari 6 menu :

1. Kirim *Email* digunakan untuk mengirim *email* kepada nama yang dipilih pada daftar.
2. Daftar *Email* digunakan untuk menampilkan nama-nama yang ada pada daftar *email*.
3. Tambah Data digunakan untuk menambah data *email* rekanan.
4. Edit Data digunakan untuk mengubah data.
5. Hapus Data digunakan untuk menghapus data.
6. Kembali ke Menu, digunakan untuk kembali ke menu *email*.



Gambar 5.9 *Form Data Email*

5.2.8 *Menu About*

Digunakan untuk menampilkan identitas pembuat perangkat lunak. *Form* ini dapat dilihat pada gambar 5.10.



Gambar 5.10 *Form About*

BAB VI

ANALISIS KINERJA PERANGKAT LUNAK

6.1 Pengujian Program

Dalam tahap ini akan dijelaskan tentang pengujian program aplikasi *email client* pada *handphone*. Dengan pengujian ini diharapkan tingkat kesalahan baik pengolahan data maupun dalam sistem itu sendiri menjadi kurang bahkan tidak ada.

Pengujian kinerja aplikasi ini bertujuan untuk mengetahui proses yang telah ada seperti pemasukan data, selain itu juga untuk mengetahui kesalahan-kesalahan yang ada sekaligus mengetahui upaya penanganan kesalahan tersebut. Penanganan kesalahan pada sistem aplikasi ini dilakukan dengan memberikan peringatan dalam bentuk pesan kesalahan yang berisikan informasi tentang keharusan untuk memberikan data tertentu dengan kata lain melakukan validasi terhadap masukan data yang dilakukan oleh pengguna sistem.

6.2 Pengujian dan Analisis

Pada tahap pengujian dan analisis membandingkan kebenaran dan kesesuaian dengan kebutuhan sistem aplikasi yang dibuat.

6.2.1 Pengujian Normal

Pengujian normal dilakukan dengan memberikan masukan data sesuai dengan yang dibutuhkan pada sistem aplikasi yang dibuat.

6.2.1.1 Pemasukan Data

a. *Form* Konfigurasi

Pada *form* konfigurasi pemasuka apabila data-data yang diinputkan sesuai dengan spesifikasi yang digunakan, berhasil seperti terlihat pada gambar 6.1.



Gambar 6.1 Pesan Data Berhasil disimpan

Apabila data yang diinputkan kurang lengkap maka akan tampil pesan koneksi Data belum lengkap, seperti terlihat pada gambar 6.2.



Gambar 6.2 Pesan Data Kurang Lengkap

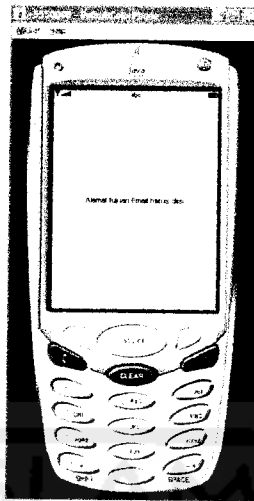
b. **Form untuk Kirim *Email* dan Forward *Email***

Pada *form* kirim *Email* data yang dibutuhkan adalah alamat *email* yang dituju, subyek *email* dan isi *email*. Sebagai contoh apabila data *email* yang diinputkan sesuai dengan spesifikasi, maka akan tampil pesan bahwa *Email* terkirim seperti terlihat pada gambar 6.3.



Gambar 6.3 Pesan *Email* Terkirim

Apabila data alamat *email* yang diinputkan tidak lengkap akan muncul pesan bahwa alamat *email* harus diisi seperti terlihat pada gambar 6.4.



Gambar 6.4 Pesan tujuan alamat *Email* harus diisi

Apabila subyek *email* tidak terisi maka akan tampil pesan bahwa *Email* harus lengkap terlihat pada gambar 6.5.



Gambar 6.5 Pesan Data *Email* harus lengkap

6.3 Kelebihan dan Kekurangan Aplikasi

Aplikasi *email client* berbasis J2ME dapat digunakan untuk mengirim dan menerima *email*. Pada kasus ini menggunakan koneksi GPRS/3G dengan memanfaatkan jaringan GSM (Telkomsel).

6.3.1 Kelebihan

Setelah dilakukan pengujian aplikasi ini memiliki kelebihan:

1. Dapat mengirimkan dan menerima *email*.
2. Dapat mengirimkan dan menerima *email* lebih panjang dari 160 karakter sehingga lebih panjang dari SMS.
3. Biaya operasional lebih murah, yaitu Rp. 12/*Kbytes* (160 karakter sms kurang lebih 200 bytes, 1 *Kbytes* = 1024 *bytes*) dibandingkan menggunakan sms dengan biaya Rp. 250/sms.

6.3.2 Kekurangan

Setelah dilakukan pengujian aplikasi ini memiliki kekurangan:

1. Tidak dapat mengirimkan dan menerima file *attachment*.
2. Hanya dapat menerima *email* pada *web mail* yang menyediakan POP3 *server*.
3. Web mail yang dapat digunakan tergantung pada operator . dalam kasus ini menggunakan GSM *provider* Telkomsel. *Email server* yang dapat digunakan www.yahoo.co.id.
4. Tidak dapat menyimpan data *email*.

BAB VII

PENUTUP

7.1 Kesimpulan

Kesimpulan yang dapat diambil dari penelitian yang berjudul “Bagaimana Membangun *Email Client* menggunakan Java 2 Micro Edition Yang Dapat Diakses Menggunakan *Handphone*” adalah:

1. *Email* yang digunakan harus mendukung POP 3 *server*.
2. *Email Client* hanya dapat menerima dan mengirim *email* berupa teks dan tidak dapat menerima *attachment* dikarenakan kapasitas memori *handphone* yang terbatas.
3. *Handphone* yang digunakan adalah *handphone* yang berbasis Java.
4. Penggunaan *server* POP 3 tergantung operator seluler masing-masing.
5. *Email* yang diterima disimpan dalam vektor bukan dalam *database*.

7.2 Saran

Saran yang diberikan berdasarkan hasil penelitian adalah sebagai berikut:

1. Perlu adanya peningkatan dan pengembangan lebih lanjut terhadap aplikasi untuk mendukung penyimpanan *email* dalam *handphone*.
2. Perlu adanya penambahan fitur yang dapat mengakomodasi *attachment* pada *email*.

DAFTAR PUSTAKA

- [FIE96] Fielding, R. et al. Multipurpose Internet *Mail* Extensions (MIME) Part One, November 1996.
- [FIE 99] Fielding, R. et al., *Hypertext Transfer Protocol - HTTP/1.1*, RFC 2616, June 1999
- [GIG01] Giggs, Roger. Antero Taivalasari, Mark VandernBriknk. *Progammng Wireless With Java 2 Platform Micro Edition*, Adyson Wesley 2001.
- [WIC02] Wicaksono, Adi. *Pemrogrman Aplikasi Wirelles dengan Java 2*, Jakarta: Elek Media Komputindo, 2002.
- [HAR03] Hartanto, Antonius Aditya. *Tip dan Trik Java 2 Micro Edition Mobile Inteface Device Programing*, Jakarta: 2003.
- [HAR06] Hartati, Sri, Suharto, G. Heri, B dan Wijono Susilo, *Pemrograman Swing Java dengan Netbeans 5*, Yogyakarta: Andi Offset. 2006.
- [MAR06] Mardiono, Tri *Membangun Solusi Mobile Business dengan Java*. Surabaya: Elex Media Komputindo, 2006.