# APLIKASI SEARCHING SMS PADA DESKTOP DENGAN MENGGUNAKAN ALGORITMA PENCARIAN STRING (ALGORITMA KNUTH-MORRIS-PRATT)

**TUGAS AKHIR**

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




Oleh :

Nama : Bayu Sulistyanto Untoro
No. Mahasiswa : 03. 523. 202

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

## LEMBAR PENGESAHAN PEMBIMBING

APLIKASI SEARCHING SMS PADA DESKTOP
DENGAN MENGGUNAKAN ALGORITMA PENCARIAN STRING
(ALGORITMA KNUTH-MORRIS-PRATT)

TUGAS AKHIR


Oleh :

Nama : Bayu Sulistyanto Untoro

No. Mahasiswa : 03. 523. 202


Yogyakarta, 10 Agustus 2007

Pembimbing,

Taufiq Hidayat, ST., MCS

## LEMBAR PENGESAHAN PENGUJI

APLIKASI SEARCHING SMS PADA DESKTOP
DENGAN MENGGUNAKAN ALGORITMA PENCARIAN STRING
(ALGORITMA KNUTH-MORRIS-PRATT)

TUGAS AKHIR

Oleh :

Nama : Bayu Sulistyanto Untoro

No. Mahasiswa : 03. 523. 202

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 20 Agustus 2007

Tim Penguji.

Taufiq Hidayat, ST., MCS
Penguji I

Hendrik, ST
Penguji II

Nur Wijayaning Rahayu, S.Kom
Penguji III

Mengetahui,
Ketua Program Studi Teknik Informatika
Universitas Islam Indonesia

Yudi Prayudi, S.Si., M.Kom

## LEMBAR PERNYATAAN KEASLIAN
## HASIL TUGAS AKHIR

Saya yang bertandatangan di bawah ini,

Nama : Bayu Sulistyanto Untoro
No. Mahasiswa : 03. 523. 202
Jurusan : Teknik Informatika

Menyatakan bahwa seluruh komponen dan isi dalam Laporan Tugas Akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan hasil karya saya sendiri, maka saya siap menanggung resiko dan konsekuensi apapun.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 10 Agustus 2007



Bayu Sulistyanto Untoro

## HALAMAN PERSEMBAHAN

Karya tulis ini dipersembahkan

**Special For. . .**

**Ayah** dan **Ibu** tercinta, sebagai bentuk pengabdianku kepadamu
serta terima kasih atas segala kasih sayang, bimbingan,
dukungan dan doa yang diberikan sampai saat ini dan selamanya . . .
Sebagai wujud baktiku, cintaku dan kasih sayangku selama ini . . .

**And for. . .**

**Mariesa Rahmawati, SH.**, terima kasih atas segala
perhatian, cinta, kasih sayang dan dukungan.

## HALAMAN MOTTO

مَا وَدَّعَكَ رَبُّكَ وَمَا قَلَىٰ ﴿٣﴾

Tuhanmu sama sekali tak akan meninggalkanmu

dan tak akan membencimu

**Q. S. Ad Dhuhaa : 3**

إِنَّ مَعَ ٱلْعُسْرِ يُسْرًا ﴿٦﴾ فَإِذَا فَرَغْتَ فَٱنصَبْ ﴿٧﴾

وَإِلَىٰ رَبِّكَ فَٱرْغَب ﴿٨﴾

Sungguh, bersama kesukaran itu pasti ada kemudahan.

Oleh karena itu, jika kamu telah selesai dari suatu tugas,

kerjakan tugas lain dengan sungguh-sungguh.

Dan hanya kepada Tuhanmulah hendaknya

kamu memohon dan mengharap

**Q. S. Asy Syarh : 6-8**

# KATA PENGANTAR

بسم الله الرحمن الرحيم

*Assalamu'alaikum wr.wb.*

Syukur Alhamdulillah penulis panjatkan kehadirat Allah SWT, yang telah memberikan rahmat, hidayah, dan karunia-Nya, sehingga laporan Tugas Akhir dapat penulis selesaikan. Tak lupa shalawat serta salam kami haturkan kepada junjungan kita Nabi Muhammad S.A.W, yang telah memberi uswatun khasanah bagi umat manusia.

Tugas Akhir ini dibuat sebagai salah satu syarat yang harus dipenuhi untuk memperoleh gelar sarjana di jurusan Teknik Informatika Universitas Islam Indonesia.

Tugas Akhir yang dilaksanakan penulis adalah membuat aplikasi searching SMS pada desktop dengan menggunakan algoritma pencarian string (algoritma Knuth-Morris-Pratt) dan bahasa pemrograman Java.

Dalam kesempatan ini penulis mengucapkan terima kasih yang sebesar–besarnnya kepada :

1. Ayah dan ibu, atas segala kasih sayang, doa selama ini dan selamanya yang tak pernah terlupakan selama-lamanya.
2. Bapak Fathul Wahid, ST., MSc, selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia dan dosen pembimbing akademik.
3. Bapak Yudi Prayudi, S.Si., M.Kom, selaku ketua Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

4. Bapak Taufiq Hidayat, ST.,MCS selaku dosen pembimbing Tugas Akhir, yang telah memberikan pengarahan dan bimbingan selama pelaksanaan Tugas Akhir dan penulisan laporan.
5. Seluruh staf pengajar FTI UII. khususnya dosen-dosen jurusan Teknik Informatika yang telah memberikan bekal ilmu.
6. Kekasihku Mariesa Rahmawati SH, yang telah memberi dukungan baik moral maupun spiritual.
7. Kakak dan Adikku, Taufik Andika K, Fauzi Robbit B
8. Teman-teman yang baik hati, Jaya, Hendrikus, Abe, Syahban, Egik, Rofi, Dwi, Ichal, Siswantara
9. Seluruh teman-teman angkatan 2003 Teknik Informatika yang telah memberikan keceriaan dan semangat tanpa henti kepada penulis.
10. Serta semua pihak yang tidak dapat penulis sebutkan satu per satu yang telah membantu dari awal hingga akhir.

Penulis menyadari bahwa laporan ini masih belum sempurna, karena keterbatasan kemampuan dan pengalaman. Oleh karena itu penulis mengharapkan kritik dan saran membangun untuk membantu penulis di masa yang akan datang.

Akhir kata penulis berharap agar laporan ini dapat bermanfaat bagi semua pihak.

*Wassalamu'alaikum Wr Wb.*

Yogyakarta, 10 Agustus 2007

Penulis

## DAFTAR ISI


HALAMAN JUDUL ..... i
LEMBAR PENGESAHAN PEMBIMBING ..... ii
LEMBAR PENGESAHAN PENGUJI ..... iii
LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR ..... iv
HALAMAN PERSEMBAHAN ..... v
HALAMAN MOTTO ..... vi
KATA PENGANTAR ..... vii
SARI ..... ix
TAKARIR ..... x
DAFTAR ISI ..... xi
DAFTAR GAMBAR ..... xiv
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 Metode Pengumpulan Data ..... 4
1.6.2 Metode Pengembangan Sistem ..... 4
1.7 Sistematika Penulisan ..... 5

**BAB II**

**LANDASAN TEORI** ..... 7

2.1 Algoritma Pencarian String ..... 7
2.2 Algoritma Knuth-Morris-Pratt ..... 8

2.2.1 Pengertian Algoritma Knuth-Morris-Pratt ..... 8
2.2.2 Fungsi Pinggiran (Border Function) ..... 9
2.2.3 Kompleksitas Waktu Algoritma KMP ..... 11
2.3 Pemrograman Berorientasi Objek ..... 12
2.4 Java ..... 13
2.5 Java 2 ..... 14
2.6 Keunggulan Java ..... 16
2.7 SMS (Short Message Services) ..... 17
2.8 Format pengiriman SMS ..... 18
2.9 Access and Terminals Command (AT Command) ..... 20
2.10 UML (Unified Modeling Language) ..... 21

**BAB III**

**METODOLOGI** ..... 24
3.1 Analisis Kebutuhan Perangkat Lunak ..... 24
3.1.1 Metode Analisis ..... 24
3.1.2 Hasil Analisis ..... 25
3.1.2.1 Masukan Sistem (*Input*) ..... 25
3.1.2.2 Keluaran Sistem (*Output*) ..... 25
3.1.2.3 Kebutuhan Perangkat Lunak ..... 26
3.1.2.4 Kebutuhan Perangkat Keras ..... 28
3.1.2.5 Antarmuka (*Interface*) ..... 28
3.1.2.6 Fungsi yang Diharapkan ..... 29
3.2 Perancangan Perangkat Lunak ..... 29
3.2.1 Metode Perancangan ..... 29
3.2.2 Hasil Perancangan ..... 29
3.2.2.1 Use Case Diagram ..... 30
3.2.2.2 Class Diagram ..... 31
3.2.2.3 Sequence Diagram ..... 32
3.2.2.4 Activity Diagram ..... 38
3.2.2.5 Perancangan Basis Data ..... 39

3.2.2.5.1 Struktur tabel..... 39
3.2.2.5.2 Relasi Tabel..... 40
3.2.2.6 Rancangan Antarmuka..... 41
3.2.2.6.1 Rancangan antarmuka utama..... 41
3.2.2.6.2 Rancangan antarmuka phonebook..... 43
3.2.2.6.3 Rancangan antarmuka form phonebook..... 44
3.2.2.6.4 Rancangan antarmuka help..... 45

**BAB IV**
**HASIL DAN PEMBAHASAN**..... 46
4.1 Implementasi Perangkat Lunak..... 46
4.1.1 Batasan Implementasi..... 46
4.1.2 Implementasi..... 47
4.1.2.1 Mekanisme penggunaan aplikasi..... 47
4.1.2.2 Implementasi Antarmuka..... 48
4.1.2.2.1 Antarmuka Utama..... 48
4.1.2.2.2 Antarmuka Phonebook..... 50
4.1.2.2.3 Antarmuka Form Phonebook..... 51
4.1.2.2.4 Antarmuka About..... 52
4.1.2.3 Implementasi Prosedural..... 52
4.2 Analisis Kerja Perangkat Lunak..... 56
4.2.1 Pengujian Perangkat Lunak..... 56
4.2.1 Pengujian waktu pencarian..... 63
4.2.3 Analisis hasil pengujian..... 64

**BAB V**
**KESIMPULAN DAN SARAN**..... 66
5.1 Kesimpulan..... 66
5.2 Saran..... 67

**DAFTAR PUSTAKA**..... 68

# DAFTAR GAMBAR


Gambar 2.1 Lingkungan kerja teknologi Java..... 15

Gambar 3.1 Use case diagram..... 31

Gambar 3.2 Class Diagram..... 32

Gambar 3.3 Sequence diagram pilih koneksi..... 33

Gambar 3.4 Sequence diagram searching SMS..... 33

Gambar 3.5 Sequence diagram tampil inbox ..... 34

Gambar 3.6 Sequence diagram tampil outbox ..... 34

Gambar 3.7 Sequence diagram delete SMS inbox ..... 35

Gambar 3.8 Sequence diagram delete SMS Outbox ..... 35

Gambar 3.9 Sequence diagram insert phonebook..... 36

Gambar 3.10 Sequence diagram edit phonebook..... 36

Gambar 3.11 Sequence diagram delete phonebook..... 37

Gambar 3.12 Sequence diagram tampil phonebook..... 37

Gambar 3.13 Activity Diagram ..... 38

Gambar 3.14 Rancangan antarmuka halaman utama..... 41

Gambar 3.15 Rancangan antarmuka halaman phonebook..... 43

Gambar 3.15 Rancangan antarmuka form phonebook..... 44

Gambar 3.16 Rancangan antarmuka halaman help..... 45

Gambar 4.1 Antarmuka halaman utama..... 50

Gambar 4.2 Antarmuka phonebook..... 51

Gambar 4.3 Antarmuka form phonebook..... 51

Gambar 4.4 Antarmuka about..... 52
Gambar 4.5 Pesan koneksi port berhasil..... 58
Gambar 4.6 Pesan koneksi port gagal..... 58
Gambar 4.7 Hasil pengujian pencarian SMS..... 59
Gambar 4.8 Pesan input belum diisi..... 60
Gambar 4.9 Pesan hasil pencarian tidak ditemukan..... 60
Gambar 4.10 Hasil pengujian tampil inbox..... 60
Gambar 4.11 Hasil pengujian tampil outbox..... 61
Gambar 4.12 Konfirmasi tambah phonebook ..... 62
Gambar 4.13 Pesan data belum diisi..... 62
Gambar 4.14 Konfirmasi tutup aplikasi..... 63
Gambar 4.14 Gambar pengujian waktu pencarian SMS..... 64

## DAFTAR TABEL

Tabel 3.1 Tabel Inbox..... 39

Tabel 3.2 Tabel Outbox..... 40

Tabel 3.3 Tabel Phonebook..... 40

Tabel 3.4 Tabel relasi..... 40

# BAB I

# PENDAHULUAN

## 1.1 Latar belakang

Teknologi komputer dan telepon seluler mengalami perkembangan yang sangat cepat. Hal ini ditandai dengan semakin banyaknya aplikasi yang ada saat ini, baik aplikasi berbasis desktop maupun aplikasi berbasis telepon seluler. Perkembangan antara aplikasi desktop dan aplikasi telepon seluler mempunyai hubungan yang erat. Aplikasi desktop dapat melengkapi kekurangan dari aplikasi telepon seluler yang ada saat ini. Hal ini dikarenakan tidak semua aplikasi telepon seluler mampu melakukan operasi seperti aplikasi berbasis desktop. Penyebab lainnya yaitu perangkat telepon seluler hanya memiliki *device* yang terbatas seperti memori yang kecil, sehingga tidak mampu melakukan operasi seperti perangkat komputer. Sebagai contoh aplikasi pemutar video pada ponsel hanya dapat memutar video berformat .avi dan tidak dapat mengkonversi sebuah video dari berbagai format ke dalam format `.avi`. Agar dapat mengkonversi ke format `.avi` dibutuhkan aplikasi desktop untuk melakukannya.

SMS (*Short Message Service*) mempunyai peranan besar dalam perkembangan telepon seluler. SMS merupakan layanan yang sampai sekarang sering digunakan oleh pengguna telepon seluler. Dibandingkan dengan komunikasi suara (*voice*), layanan SMS mempunyai kelebihan tersendiri yaitu hemat beaya dan dapat menyimpan informasi yang dikirim maupun diterima pengguna dalam memori telepon seluler itu sendiri atau memori kartu, sehingga

pengguna dapat melihat kembali SMS yang telah diterimanya apabila suatu saat dibutuhkan.

Pengguna telepon seluler banyak yang mengeluhkan ketika ingin mencari SMS yang telah tersimpan dalam telepon selulernya. Pengguna telepon seluler harus membuka satu persatu SMS yang tersimpan di telepon seluler, ketika akan mencari SMS yang diinginkan. Apabila SMS yang tersimpan dalam telepon seluler hanya sedikit, tentunya pencarian yang dilakukan tidak membutuhkan waktu yang lama. Berbeda halnya, apabila SMS yang tersimpan dalam telepon seluler sangatlah banyak. Hal ini tentunya akan membutuhkan waktu yang lebih lama apabila pengguna telepon seluler melakukan pencarian terhadap SMS yang akan dicarinya.

## 1.2 Rumusan masalah

Berdasarkan latar belakang di atas, maka dapat diambil rumusan masalah bagaimana membangun sebuah aplikasi searching SMS pada desktop yang memiliki kemampuan dalam pencarian sms.

## 1.3 Batasan masalah

Dalam membangun aplikasi tersebut terdapat beberapa batasan masalah,yaitu:

1. Input yang diberikan adalah kata yang akan dicari.
2. Output yang dihasilkan adalah daftar SMS yang di dalamnya terdapat kata yang dicari.

3. Pencarian dilakukan pada daftar SMS *inbox* dan *outbox*, berdasarkan isi SMS.
4. Program aplikasi yang digunakan untuk membangun aplikasi searching SMS menggunakan NetBeans dengan bahasa pemrograman Java.

## 1.4 Tujuan penelitian

Tujuan yang ingin dicapai dalam tugas akhir ini adalah :

1. Membangun aplikasi *searching* SMS pada desktop yang memiliki kemampuan dalam pencarian dan tidak membutuhkan banyak memori dalam pengoperasiannya.
2. Aplikasi yang dibangun dapat menambah perbendaharaan aplikasi berbasis desktop yang telah ada saat ini.
3. Mengimplementasikan teknologi J2SE, untuk membangun sebuah aplikasi.
4. Aplikasi yang dibangun dapat digunakan oleh pengguna telepon seluler dalam melakukan pencarian tarhadap daftar SMS.

## 1.5 Manfaat penelitian

Manfaat utama dari pembuatan aplikasi searching SMS pada desktop adalah membantu pengguna telepon seluler dalam melakukan pencarian SMS yang tersimpan dalam memori telepon selulernya dengan bantuan komputer.

Selain manfaat tersebut, pengguna juga dapat melihat sms yang tersimpan dalam *inbox* dan *outbox*.

## 1.6 Metodologi penelitian

Metodologi yang digunakan dalam penelitian ini meliputi metode pengumpulan data dan pengembangan sistem.

### 1.6.1 Metode pengumpulan data

Pengumpulan data dibedakan menjadi dua yaitu :

a. Data primer

1. Metode Observasi

Yaitu metode yang dilakukan dengan cara pengamatan secara langsung pada obyek yang diteliti dan segala sesuatu yang akan diteliti.

2. Metode Kepustakaan

Metode ini digunakan untuk mendapatkan informasi tambahan berupa referensi buku-buku kepustakaan yang digunakan sebagai acuan dalam pembangunan aplikasi antara lain buku, literatur, makalah dan hal-hal yang berkaitan dengan topik yang sedang penulis teliti

b. Data sekunder

Data sekunder diperoleh dengan cara mengumpulkan data dengan metode telaah dokumen yaitu mempelajari dokumen artikel dan catatan lain yang berkaitan dengan bidang permasalahan yang dihadapi.

### 1.6.2 Metode pengembangan sistem

Metode pengembangan sistem yang digunakan meliputi :

1. Analisis kebutuhan perangkat lunak

Analisa kebutuhan perangkat lunak merupakan tahap awal dari pengembangan perangkat lunak. Analisis kebutuhan perangkat lunak dilakukan untuk mendapatkan informasi tentang perangkat lunak yang akan dibangun.

2. Perancangan perangkat lunak

   Yaitu proses merancang perangkat lunak yang akan dibangun setelah melakukan analisis kebutuhan perangkat lunak.

3. Implementasi perangkat lunak

   Implementasi perangkat lunak adalah proses pembuatan aplikasi sesuai dengan rencana. Setelah pembuatan perancangan maka dapat dipresentasikan hasil dari perancangan yang telah dibuat.

4. Analisa kerja perangkat lunak

   Yaitu proses pengujian terhadap aplikasi yang dibangun apakah sesuai dengan yang diharapkan.

## 1.7 Sistematika penulisan

Sistematika penulisan dalam laporan tugas akhir ini dibagi menjadi lima bab. Hal ini dilakukan untuk mempermudah dalam pembahasan. Adapun sistematika penulisan laporan sebagai berikut :

**BAB I PENDAHULUAN**

Bab ini berisi pembahasan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan laporan.

**BAB II LANDASAN TEORI**

Bab ini menjelaskan tentang teori-teori yang berhubungan dengan penelitian antara lain teori-teori tentang algoritma pencarian kata, algoritma Knuth Morris Pratt, pemrograman berorientasi objek, Java, Java2, keunggulan Java, SMS, UML dan teori pendukung lainnya.

**BAB III METODOLOGI**

Bab ini berisi penjelasan tentang analisis kebutuhan perangkat lunak yang digunakan, hasil analisis kebutuhan perangkat lunak, perancangan antarmuka, metode dan hasil perancangan.

**BAB IV HASIL DAN PEMBAHASAN**

Bab ini membahas tentang analisis kinerja dari perangkat lunak. Bagian hasil memuat tentang hasil dicapai. Sedangkan bagian pembahasan berisi pembahasan hasil yang dicapai. Pembahasan dapat berupa uraian tentang mengapa hasil diperoleh, kelebihan dan kekurangan aplikasi yang dibangun.

**BAB V SIMPULAN DAN SARAN**

Dalam bab ini berisi kesimpulan yang merupakan rangkuman dari hasil analisis kinerja pada bagian sebelumnya. Selain itu dalam bab ini juga memuat saran yang perlu diperhatikan berdasarkan keterbatasan yang ditemukan dan asumsi-asumsi yang dibuat selama pembuatan aplikasi.

# BAB II

# LANDASAN TEORI

## 2.1 Algoritma Pencarian String

Pencarian *string* di dalam teks disebut juga pencocokan *string* (*string matching* atau *pattern matching*). *String-matching* fokus pada pencarian satu atau semua kata (lebih umum disebut *pattern*) dalam sebuah teks.

Metode untuk melakukan pencocokan ada empat macam yaitu :

1. Pencocokan dari kiri ke kanan (merupakan cara standar).
2. Pencocokan dari kanan ke kiri.
3. Pencocokan spesifik.
4. Pencocokan acak atau tidak relevan.

Persoalan pencarian string dirumuskan sebagai berikut:

- Teks (*text*), yaitu (*long*) *string* yang panjangnya *n* karakter.
- *Pattern*, yaitu *string* dengan panjang *m* karakter ($m < n$) yang akan dicari di dalam teks.

Penyelesaian pencarian string mempunyai bermacam-macam cara yang mempunyai kelebihan dan kekurangan masing-masing.

Contoh pencarian *string*:

Mencari kata pulang pada teks kapan ayah pulang dari Klaten?

Pattern: pulang

Teks: kapan ayah **pulang** dari Klaten?

↑
target

Contoh pencarian *string*:

Pattern: mana

Teks: bagai**mana** susunan **mana**jemen perusahaan kita?

## 2.2 Algoritma Knuth-Morris-Pratt

### 2.2.1 Pengertian Algoritma Knuth-Morris-Pratt

Algoritma Knuth Morris Pratt dikembangkan oleh D. E. Knuth, bersama-sama dengan J. H. Morris dan V. R. Pratt. Algoritma Knuth Morris Pratt merupakan algoritma yang digunakan untuk melakukan proses pencocokan string. Algoritma ini merupakan jenis *Exact String Matching Algorithm* yang merupakan pencocokan string secara tepat dengan susunan karakter dalam string yang dicocokkan memiliki jumlah maupun urutan karakter dalam string yang sama [MUN06].

Algoritma Knuth Morris Pratt mencari sebuah kata dalam teks dengan melakukan observasi awal (*preprocessing*) yaitu ketika muncul ketidaksamaan kata (*string*), maka pencarian mempunyai informasi mengenai kapan kesamaan selanjutnya bermula, dengan cara mengecek ulang kata sebelumnya. Algoritma Knuth Morris Pratt dapat memelihara informasi yang digunakan untuk melakukan jumlah pergeseran ke kanan dalam melakukan pencarian. Algoritma Knuth Morris Pratt melakukan pencocokan dari kiri ke kanan sehingga dengan menggunakan algoritma Knuth Morris Pratt waktu pencarian dapat dikurangi secara signifikan.

### 2.2.2 Fungsi Pinggiran (Border Function)

Misalkan $A$ adalah alphabet, dan $x = x_1x_2\ldots x_k$ , $k \in \mathbf{N}$, adalah *string* yang panjangnya m yang dibentuk dari karakter-karakter di dalam alfabet $A$.

- Awalan (*prefix*) dari $x$ adalah upa-string (*substring*) $u$ dengan

  $u = x_1x_2\ldots x_{k-1}$ , $k \in \{1, 2, \ldots, k-1\}$

  dengan kata lain, string $x$ diawali dengan string $u$.

- Akhiran (*suffix*) dari x adalah upa-string (substring) u dengan

  $u = x_{k-b}x_{k-b+1}\ldots x_k$ , $k \in \{1, 2, \ldots, k-1\}$

  dengan kata lain, string $x$ diawali dengan string $v$.

Pinggiran (*border*) dari $x$ adalah upa-string $r$ sehingga

$r = x_1x_2\ldots x_{k-1}$ dan $u = x_{k-b}x_{k-b+1}\ldots x_k$ , $k \in \{1, 2, \ldots, k-1\}$

dengan kata lain, pinggiran dari $x$ adalah upa-string yang keduanya awalan dan juga akhiran sebenarnya dari $x$ [MUN06b].

Misalkan $x$ = `komputer`

Awalan sebenarnya dari $x$ adalah

□, `k`, `ko`, `kom`, `komp`, `kompu`, `komput`, `kompute`

Akhiran sebenarnya dari $x$ adalah

□, `r`, `er`, `ter`, `uter`, `puter`, `mputer`, `omputer`

Pinggiran dari $x$ adalah

□, `k`, `r`

Pinggiran □ mempunyai panjang 0, pinggiran `k` dan pinggiran `r` masing-masing mempunyai panjang 1.

(ket: □ = *string* kosong)

Fungsi pinggiran *b(j)* didefinisikan sebagai ukuran awalan terpanjang dari *P* yang merupakan akhiran dari *P*[1..*j*].

Algoritma menghitung fungsi pinggiran adalah sebagai berikut:

```
procedure HitungPinggiran(input m : integer, P :
array[1..m] of char,
                          output b : array[1..m] of
integer)
{ Menghitung nilai b[1..m] untuk pattern P[1..m] }

Deklarasi
  k,q : integer

Algoritma:
  b[1]←0
  q←2
  k←0
  for q←2 to m do
    while ((k > 0) and (P[q] ≠ P[k+1])) do
      k←b[k]
    endwhile
    if P[q]=P[k+1] then
      k←k+1
    endif
    b[q]=k
  endfor
```

Algoritma KMP selengkapnya:

```
procedure PencarianKMP (input m, n : integer,
                        input P : array[1..m] of char,
                        input T : array[1..n] of char,
                        output idx : integer)

{ Mencari kecocokan pattern P di dalam teks T dengan
algoritma Knuth-Morris-Pratt. Jika ditemukan P di dalam T,
lokasi awal kecocokan disimpan di dalam peubah idx.
  Input: pattern P yang panjangnya m dan teks T yang
panjangnya n.
       Teks T direpresentasikan sebagai string (array of
character)
  Output:posisi awal kecocokan (idx). Jika P tidak
ditemukan, idx = -1}
```

```
Deklarasi
  i, j : integer
  ketemu : boolean
  b : array[1..m] of integer

procedure HitungPinggiran(input m : integer, P : array[1..m]
of char, output b : array[1..m] of integer)
  { Menghitung nilai b[1..m] untuk pattern P[1..m] }

Algoritma:
 HitungPinggiran(m, P, b)
  j←0
  i←1
  ketemu←false
  while (i ≤ n and not ketemu) do

    while((j > 0) and (P[j+1]≠T[i])) do
      j←b[j]
    endwhile

    if P[j+1]=T[i] then
      j←j+1
    endif
    if j = m then
      ketemu←true
    else
      i←i+1
    endif
 endwhile
 if ketemu then
   idx←i-m+1  { catatan: jika indeks array dimulai dari 0,
maka idx←i-m }
 else
   idx←-1
 endif
```

### 2.2.3 Kompleksitas Waktu Algoritma KMP

Untuk menghitung fungsi pinggiran dibutuhkan waktu $O(m)$, sedangkan pencarian *string* membutuhkan waktu $O(n)$, sehingga kompleksitas waktu algoritma KMP adalah $O(m+n)$ dimana m adalah panjang kata (*string*) yang dicari dan n adalah panjang teks (*text*) tempat pencarian. Algoritma KMP Memiliki kompleksitas waktu yang linier baik kasus terbaik maupun kasus terburuk.

```
Deklarasi
  i, j : integer
  ketemu : boolean
  b : array[1..m] of integer

procedure HitungPinggiran(input m : integer, P : array[1..m]
of char, output b : array[1..m] of integer)
  { Menghitung nilai b[1..m] untuk pattern P[1..m] }

Algoritma:
 HitungPinggiran(m, P, b)
  j←0
  i←1
  ketemu←false
  while (i ≤ n and not ketemu) do

    while((j > 0) and (P[j+1]≠T[i])) do
      j←b[j]
    endwhile

    if P[j+1]=T[i] then
      j←j+1
    endif
    if j = m then
      ketemu←true
    else
      i←i+1
    endif
 endwhile
 if ketemu then
   idx←i-m+1  { catatan: jika indeks array dimulai dari 0,
maka idx←i-m }
 else
   idx←-1
 endif
```

### 2.2.3 Kompleksitas Waktu Algoritma KMP

Untuk menghitung fungsi pinggiran dibutuhkan waktu $O(m)$, sedangkan pencarian *string* membutuhkan waktu $O(n)$, sehingga kompleksitas waktu algoritma KMP adalah $O(m+n)$ dimana m adalah panjang kata (*string*) yang dicari dan n adalah panjang teks (*text*) tempat pencarian. Algoritma KMP Memiliki kompleksitas waktu yang linier baik kasus terbaik maupun kasus terburuk.

3. Polymorphism

   Polymorphism berarti mempunyai banyak bentuk. Dalam pemrograman berorientasi objek karakteristik ini berlaku untuk tingkah laku ataupun method.

Keunggulan pemrograman berorientasi objek antara lain kemudahan untuk mengelola kompleksitas aplikasi yang dibuat, dapat menghasilkan aplikasi dengan struktur yang kokoh, pemeliharaan lebih mudah, dapat dilakukan penggunaan ulang (*reuseability*), efisiensi waktu, dapat diperluas (*Extendable*), mudah dalam pemeliharaannya (*maintainable*).

## 2.4 Java

Java merupakan bahasa pemrograman yang dikembangkan oleh Sun Microsystem pada tahun 1991. Pada awal mulanya bahasa pemrograman ini dinamakan OAK. Java berdiri di atas sebuah mesin interpreter yang diberi nama Java Virtual Machine (JVM). JVM inilah yang digunakan untuk membaca *bytecode* dalam file .class dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Java *bytecodes* merupakan instruksi mesin yang tidak spesifik terhadap prosesor komputer. Oleh karena itu bahasa Java disebut sebagai bahasa pemrograman yang *portable* karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM.

Platform Java terdiri dari kumpulan library, JVM, kelas-kelas *loader* yang dipaket dalam sebuah lingkungan rutin Java dan sebuah *compiler*, *debugger* dan kakas lain yang dipaket dalam Java Development Kit (JDK).

## 2.5 Java 2

Java versi 1.2 atau lebih dikenal dengan nama Java 2 merupakan hasil pengembangan dari Java versi sebelumnya yaitu JDK 1.1 (Java Development Kit versi 1.1). Java 2 terdiri dari JDK dan JRE versi 1.2. Aplikasi-aplikasi Java yang *compatible* dengan Java 2 ini dikenal dengan Java 2 Compliant [ADY02a]. Java 2 SDK adalah satu set perangkat lunak yang digunakan untuk manajemen dan membangun berbagai macam aplikasi Java. Java 2 telah berkembang sedemikian pesat, bukan hanya fitur-fitur standar melainkan telah merambah ke pelayanan enterprise dengan J2EE serta merambah ke dunia mobile dengan J2ME.

Java 2 dibagi menjadi 3 kategori yaitu :

1. J2SE (Java 2 Standart Edition)

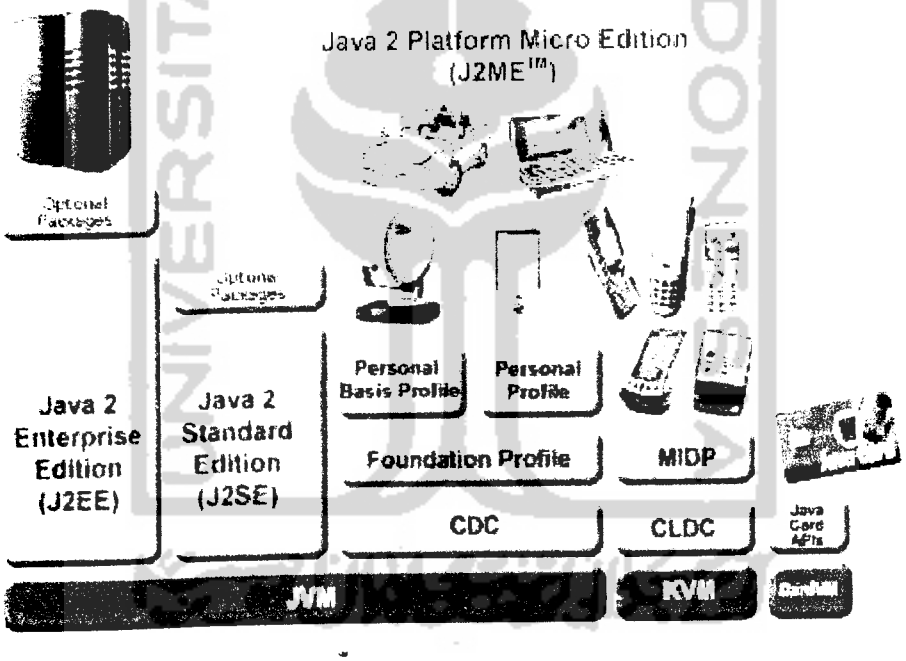
   Kategori ini digunakan untuk menjalankan dan mengembangkan aplikasi-aplikasi Java pada level *Personal Computer*. Java Development Kit (JDK) adalah salah satu tool dari J2SE untuk mengkompilasi dan menjalankan program Java dan JRE.

2. J2ME (Java 2 Micro Edition)

   J2ME merupakan bagian dari J2SE, oleh karena itu tidak semua library yang terdapat pada J2SE dapat digunakan pada J2ME. J2ME memiliki beberapa library khusus yang tidak dimiliki J2SE. Kategori ini digunakan untuk menjalankan dan mengembangkan aplikasi-aplikasi Java pada perangkat-perangkat berkapasitas memori terbatas.

3. J2EE (Java 2 Enterprise Edition)

Kategori ini digunakan untuk menjalankan dan mengembangkan aplikasi-aplikasi Java yang komplek pada lingkungan enterprise dengan menambahkan fungsionalitas-fungsionalitas Java semacam EJB (enterprise Java Bean), Java CORBA, Servlet dan JSP, serta Java XML (Extensible Markup Language). J2EE mencakup teknologi yang dapat digunakan di semua lapisan dari sebuah system informasi.




Gambar 2.1 Lingkungan kerja teknologi Java

## 2.6 Keunggulan Java

Java mempunyai beberapa keunggulan dibandingkan dengan bahasa pemrograman lainnya. Keunggulan bahasa pemrograman Java antara lain:

1. Berorientasi objek

   Java adalah bahasa pemrograman yang berorientasi pada objek. Java membagi program menjadi objek-objek serta memodelkan sifat dan tingkah laku masing-masing dalam menyelesaikan suatu masalah.

2. Java bersifat multiplatform

   Java dirancang untuk mendukung aplikasi yang dapat beroperasi di lingkungan jaringan berbeda. Untuk mengakomodasi hal tersebut, Java compiler membangkitkan *bytecodes* (sebuah format yang tidak tergantung pada arsitektur tertentu yang didesain untuk mengirimkan kode ke banyak platform perangkat keras dan perangkat lunak secara efisien). Java dapat dijalankan oleh banyak platform seperti Linux, Unix, Windows, Solari, maupun Mac.

3. Java bersifat multithread

   Multithreading adalah kemampuan suatu program komputer untuk mengerjakan beberapa proses dalam suatu waktu. Thread dalam Java memiliki kemampuan untuk memanfaatkan kelebihan multi prosessor apabila sistem operasi yang digunakan mendukung multi prosessor.

4. Dapat didistribusi dengan mudah

   Java memiliki library rutin yang lengkap untuk dirangkai pada protocol TCP/IP (seperti HTTP dan FTP) dengan mudah. Kemampuan

networking Java lebih kuat dan lebih mudah digunakan. Java memudahkan tugas pemrograman jaringan yang sulit seperti membuka dan mengakses sebuah soket koneksi. Java juga mamudahkan pembuatan CGI (*Common Gateway Interface*).

5. Bersifat dinamis

   Java dirancang untuk beradaptasi dengan lingkungan yang sedang berkembang. Java bersifat dinamis dalam tahap *linking*. Class yang ada dapat di *link* sebatas yang diperlukan, apabila diperlukan modul kode yang baru dapat di *link* dari beberapa sumber, bahkan dari sumber dalam jaringan Internet.

## 2.7 SMS (Short Message Services)

SMS adalah suatu layanan GSM (*Global System for Mobile Communications*) maupun CDMA (*Code Division Multiple Access*) yang mempunyai kemampuan untuk mengirim dan menerima pesan tertulis (teks) dari dan ke perangkat bergerak (*mobile device*) yang dikembangkan dan distandarisasikan oleh ETSI. Saat melakukan pengiriman SMS dari telepon selular, pesan SMS tersebut tidak langsung dikirim ke telepon seluler tujuan, akan tetapi terlebih dulu dikirim ke SMS Center (SMSC) dengan prinsip Store and Forward, setelah itu baru dikirim ke telepon seluler yang dituju. Pesan teks dikemas dalam satu paket atau *frame* yang berkapasitas maksimal 160 *byte* yang dapat direpresentasikan berupa 160 karakter huruf latin atau 70 karakter alfabet non-latin seperti alfabet Arab atau Cina [HEN05].

## 2.8 Format pengiriman SMS

Ada dua mode dalam pengiriman dan penerimaan pesan SMS yaitu mode teks dan mode PDU. Mode teks adalah format pesan dalam bentuk teks asli yang dituliskan pada saat akan mengirim pesan. Sedangkan mode PDU adalah format pesan dalam bentuk oktet heksadesimal dan oktet semi desimal dengan panjang mencapai 160 (7 bit). Adapun skema dari format PDU telah diatur dan ditetepkan oleh ETSI sebagai berikut :

| SCA | PDU Type | MR | DA | PID | DCS | VP | UDL | UD |
|---|---|---|---|---|---|---|---|---|

Keterangan :

**SCA (*Service Center Adrress*)**

*Service Center Adrress* adalah informasi dari alamat (nomor) SMSC.

**PDU Type**

Nilai *default* dari PDU Type untuk SMS adalah 11 hexa.

**MR (*Message Reference*)**

*Message Reference* adalah acuan dari pengaturan pesan SMS.

**DA (*Destination Addres*)**

*Destination Addres* adalah alamat (nomor) tujuan, terdiri atas panjangnya nomor.

**PID (*Protocol Identifier*)**

*Protocol Identifier* adalah format dari cara pengiriman pesan, yang biasanya diatur oleh telepon seluler. Misalnya format standar *text, fax, e-mail, telex.*

**DCS (*Data Coding Scheme*)**

*Data Coding Scheme* adalah rencana dari pengkodean data untuk menentukan kelas dari pesan tersebut apakah berupa SMS teks standar atau *blinking* SMS.

**VP (*Validity Period*)**

*Validity Period* adalah lama waktu pesan SMS disimpan di SMSC apabila pesan tersebut gagal.

**UDL (*User Data Length*)**

*User Data Length* adalah panjang pesan SMS dalam bentuk standar.

**UD (*User Data*)**

*User Data* adalah isi pesan dalam format heksadesimal.

Misalnya :

Sebuah pesan SMS dikirim ke nomor 628122898840 dengan isi pesan "Pesan pendek". Maka format PDU adalah

00 11 00 0C91261822988804 00 00 AB 0C D0F23CEC06C1CB6E72790D

Dengan keterangan sebagai berikut:

| Oktet / Digit Hexa | Keterangan |
|---|---|
| 00 | Informasi dari alamat (nomor) SMSC. |
| 11 | Nilai default dari PDU Type adalah 11 hexa, yang berarti bahwa 11 hexa = 00000100. |
| 00 | Pengaturan SMS dilakukan sendiri oleh telepon seluler. |
| 0C91261822988804 | Destination address. |
| 00 | Nilai default dari PID adalah 00 = Standar teks. |
| 00 | Pesan yang dikirim berupa SMS teks standar, jadi DCS hasilnya 00 |
| AB | Pada contoh di atas waktu VP-nya 5 hari, maka nilai VP adalah AB. |
| 0C | Pesan yang dikirim adalah "Pesan pendek" yang memiliki 12 karakter (0C h). |
| D0F23CEC06C1CB6E72790D | Hasil dari pengkodean isi SMS "Pesan pendek" ke dalam PDU. |

## 2.9 Access and Terminals Command (AT Command)

AT *Command* adalah perintah-perintah yang dapat digunakan sebagai sarana untuk membangun komunikasi antara komputer dengan peralatan *wireless* melalui *port* yang telah tersedia. AT *Command* juga dapat digunakan untuk mengetahui *vendor* dari telepon seluler yang digunakan, kekuatan sinyal, membaca pesan yang ada pada SIM *Card*, mengirim pesan, mendeteksi pesan SMS baru yang masuk secara otomatis, menghapus pesan pada SIM *card* dan masih banyak lagi fungsi lainnya [WAH05].

Sebelum menampilkan daftar SMS yang tersimpan dalam memori telepon seluler maupun memori kartu, maka harus mengetahui format SMS terlebih dahulu. Setiap telepon seluler mendukung format PDU, namun ada juga telepon seluler yang telah mendukung format teks. Untuk mengetahui apakah telepon seluler yang digunakan mendukung format PDU maupun format teks, *AT Command* yang digunakan yaitu :

AT+CMGF? Respon (0,1)

Apabila respon dari AT Command tersebut bernilai 0 maka telepon seluler mendukung format PDU saja, Apabila respon dari *AT Command* tersebut bernilai 1 maka telepon seluler mendukung format teks saja, Apabila *respon* dari *AT Command* tersebut bernilai 0,1 maka telepon seluler mendukung format PDU dan format teks.

AT Command untuk menampilkan daftar sms yang ada di telepon seluler yang memiliki format pdu dan mendukung format teks antara lain :

| Format PDU | Format Teks | Keterangan |
|---|---|---|
| AT+CMGL=0 | AT+CMGL="REC UNREAD" | Menampilkan daftar SMS baru (yang belum terbaca) |
| AT+CMGL=1 | AT+CMGL="REC READ" | Menampilkan daftar SMS dalam Inbox (yang sudah terbaca). |
| AT+CMGL=2 | AT+CMGL=" STO UNSENT" | Menampilkan daftar SMS Draft (belum terkirim). |
| AT+CMGL=3 | AT+CMGL="STO SENT" | Menampilkan daftar SMS Outbox (terkirim). |
| AT+CMGL=4 | AT+CMGL="ALL" | Menampilkan seluruh SMS (SMS yang ada di Inbox, Outbox, draft). |

## 2.10 UML (*Unified Modeling Language*)

UML merupakan pengganti dari metode analisis beorientasi objek dan design berorientasi objek yang dibuat oleh Grady Booch, James Rumbaugh dan Ivar Jacobson.

UML adalah bahasa untuk menspesifikasikan, memvisualisasikan, membangun dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak). *Artifact* dapat berupa model, deskripsi atau perangkat lunak dari sistem perangkat lunak [HEN06].

Pada bahasa permodelan UML terdiri dari beberapa diagram, antara lain :

1. ***Use case diagram***

*Use case diagram* digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case diagram* terdiri dari

diagram untuk use *case* dan *actor*. *Use case* merepresentasikan operasi-operasi yang dilakukan oleh *actor*. *Actor* memrepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *use case* [FAR05].

**2. *Class diagram***

*Class diagram* menggambarkan *class* dan hubungan antar *class* di dalam sistem. *Class diagram* dibangun berdasarkan *use case diagram* yang telah dibuat sebelumnya.

Class *digambarkan* dengan sebuah kotak dibagi menjadi tiga bagian. Bagian yang paling atas diisikan nama *class*, bagian tengah diisikan variabel yang dimiliki *class*, dan bagian bawah diisikan method-method dari *class*. Nama *class* digunakan untuk mendeklarasikan *class*. Variabel digunakan untuk mendeskripsikan variabel dari *class*. Variabel memiliki tiga macam *acces level* yaitu: *public* (variabel dapat diakses secara langsung dari objek yang diturunkan), *private* (tidak dapat diakses secara langsung dari objek yang diturunkan, variabel *private* hanya dapat diakses di dalam class itu saja namun tidak dapat diakses oleh *subclass*), dan *protected* (seperti juga variabel *private*, hanya tersedia di dalam class di mana variabel itu dideklarasikan. Perbedaannya variabel protected dapat juga diakses di dalam subclass dari *class* di mana variabel *protected* ini dideklarasikan). Method menunjukkan operasi-operasi yang dapat dilakukan oleh *class*. Method terdiri dari dua bagian yaitu bagian deklarasi

method (mendefinisikan nama *method* dan attribut lainnya) dan bagian badan method (berisikan semua opersasi yang akan dilakukan oleh method). Method juga memiliki tiga tingkatan *acces* yaitu *public*, *private* dan *protected* [FAR05].

3. ***Sequence diagram***

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan sekitar sistem (termasuk pengguna, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri antara dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Tujuan penggunaan *sequence diagram* yaitu untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem.

4. ***Activity diagram***

Menggambarkan rangkaian aliran dari aktivitas *actor*, digunakan untuk mendeskrepsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case*. Pada umumnya *activity diagram* tidak menampilkan secara detail urutan proses, namun hanya memberikan gambaran global bagaimana urutan prosesnya. Sehingga sering kali diagram ini digunakan untuk memodelkan aktivitas bisnis dalam *level* konseptual [HEN06].

# BAB III

# METODOLOGI

## 3.1 Analisis Kebutuhan Perangkat Lunak

### 3.1.1 Metode Analisis

Metode analisis secara terminologi adalah suatu cara untuk mempelajari sesuatu dengan menguraikan bagian-bagiannya. Analisis kebutuhan sistem dilakukan untuk mengetahui, menentukan dan menerjemahkan permasalahan serta kebutuhan yang diperlukan dalam pengembangan sistem aplikasi, seperti masukan dan keluaran sistem, serta fungsi-fungsi lain yang dibutuhkan. Metode yang digunakan untuk menganalisis kebutuhan perangkat lunak yaitu dengan menggunakan metode analisis berorientasi objek yang dilakukan untuk mendapatkan pemahaman yang lebih baik mengenai kebutuhan dan persyaratan fungsional perangkat lunak yang akan dibangun. Metode ini digunakan karena memiliki beberapa kelebihan. Kelebihan metode analisis berorientasi objek antara lain :

1. Kemudahan untuk mengelola kompleksitas aplikasi yang dibuat.
2. Kemudahan untuk melakukan pengembangan aplikasi.
3. Dapat menghasilkan aplikasi dengan struktur yang kokoh.
4. Dapat dilakukan pemakaian ulang kode.
5. Pemeliharaannya lebih mudah.

### 3.1.2 Hasil Analisis

Berdasarkan hasil analisis kebutuhan, diperoleh gambaran tentang aplikasi yang akan dibangun. Secara garis besar gambaran aplikasi yang akan dibangun adalah aplikasi yang digunakan untuk mencari SMS pada daftar SMS *inbox* dan *outbox* berdasarkan kata yang dicari. Selain itu, aplikasi dapat digunakan untuk melihat daftar SMS yang tersimpan pada daftar *inbox* maupun daftar *outbox*.

#### 3.1.2.1 Masukan Sistem (*Input*)

*Input* yang diberikan pengguna ke sistem adalah

1. Kata yang akan dicari dalam daftar SMS.
2. *Port* untuk koneksi.
3. Data *phonebook* berupa no telepon dan nama.
4. Data SMS *inbox* dan *outbox* dari telepon seluler.

#### 3.1.2.2 Keluaran Sistem (*Output*)

*Output* yang dihasilkan oleh sistem adalah :

1. Daftar SMS hasil pencarian pada *inbox* maupun *outbox* yang didalamnya terdapat kata yang dicari, meliputi no telepon, isi pesan dan tanggal.
2. Daftar SMS yang terdapat dalam *inbox* telepon, meliputi nomor pengirim, isi pesan dan tanggal.
3. Daftar SMS yang terdapat dalam *outbox* telepon, meliputi nomor tujuan, isi pesan dan tanggal.
4. Daftar *phonebook*, meliputi nama dan no telepon.

### 3.1.2.3 Kebutuhan Perangkat Lunak

Perangkat lunak (*software*) yang digunakan dalam pembuatan aplikasi *searching* SMS adalah :

1. Microsoft Windows XP

   Microsoft Windows XP merupakan sistem operasi yang digunakan dalam pembuatan aplikasi *searching* SMS.

2. JDK-6-Windows-i586

   JDK-6-Windows-i586 merupakan Java SDK versi terbaru untuk sistem operasi Windows. Java SDK adalah satu set perangkat lunak yang digunakan untuk manajemen dan membangun berbagai macam aplikasi Java.

   Pembuatan aplikasi *searching* SMS menggunakan bahasa pemrograman Java dan JDK yang digunakan adalah JDK-6-windows-i586. Alasan penggunaan JDK-6-windows-i586 yaitu JDK ini merupakan JDK versi terbaru dari Sun Microsystem yang mana JDK ini lebih *powerfull* dibanding versi sebelumnya.

3. Javacomm20-win32.

   Javacomm merupakan paket kelas tambahan pada Java 2 Software Development Kit (Java 2 SDK) untuk hubungan dengan port komputer.

   Aplikasi *searching* SMS yang dibangun merupakan aplikasi pencarian yang mana sumber pencarian berasal dari telepon seluler (SMS di memori telepon seluler maupun memori kartu), untuk itu diperlukan

Javacomm20-win32 sebagai paket tambahan dalam Java yang digunakan untuk mendeteksi dan menghubungkan komputer dengan telepon seluler.

4. Mysql-connector-java-5.0.4-bin

   Mysql connector merupakan paket kelas tambahan pada Java yang digunakan untuk menghubungkan aplikasi Java yang dibangun dengan MySQL dimana basisdata aplikasi tersebut dibangun.

5. Netbeans IDE 5.5

   Netbeans 5.5 merupakan software *open sorce* yang didirikan oleh Sun Microsystems. Netbeans 5.5 merupakan versi terbaru dari Netbeans yang dapat digunakan untuk mengembangkan perangkat lunak dengan bahasa pemrograman Java. Selain itu Netbeans IDE ini dapat mendukung bahasa pemrograman lainnya.

   Pembuatan aplikasi *searching* SMS menggunakan Netbeans 5.5 sebagai IDEnya. Alasan penggunaan Netbeans 5.5 yaitu dapat memudahkan penulisan program, *compile* program dan menjalankan program Java.

6. MySQL 5

   Pembuatan aplikasi *searching* SMS menggunakan MySQL 5 sebagai basisdatanya. MySQL 5 dipilih sebagai basisdata dari aplikasi *searching* SMS karena keunggulannya dalam pembuatan dan manajemen basisdata.

### 3.1.2.4 Kebutuhan Perangkat Keras

Spesifikasi perangkat keras *(hardware)* yang digunakan dalam pembuatan aplikasi *searching* SMS adalah:

1. Komputer dengan processor Intel P4 1.8 Ghz.
2. RAM 512MB pc3200.
3. Hardisk 40 GB.
4. Monitor.
5. Mouse
6. Keyboard.
7. Telepon Seluler
8. Kabel data.

### 3.1.2.5 Antarmuka (*Interface*)

Antarmuka merupakan unsur penting dalam pembuatan suatu aplikasi. Antarmuka merupakan jembatan antara user dan sistem dalam melakukan interaksi. Kebutuhan antarmuka (*interface*) dalam pembuatan aplikasi *searching* SMS adalah aplikasi GUI (*Graphical User Interface*) yaitu aplikasi yang tampilannya berupa grafik. Selain itu antarmuka aplikasi *searching* SMS ini bersifat *user friendly*, dengan tujuan agar program yang telah dibangun dapat digunakan dengan mudah oleh pengguna. Artinya perangkat lunak yang dibangun mudah dimengerti dan tidak membingungkan pengguna. Hal ini dilakukan dengan pembuatan menu-menu yang mudah dimengerti oleh pengguna.

#### 3.1.2.6 Fungsi yang diharapkan

Fungsi dan kinerja yang diharapkan dalam sistem ini antara lain :

1. Perangkat lunak yang dibuat diharapkan mampu menampilkan data berupa daftar SMS hasil pencarian yang didapat dalam daftar SMS *inbox* maupun *outbox*.
2. Perangkat lunak yang dibuat dapat menerima data masukan yang dimasukkan oleh *user* secara berulang-ulang dengan nilai yang berbeda-beda, dalam hal ini adalah kata yang akan dicari.

## 3.2 Perancangan Perangkat Lunak

### 3.2.1 Metode perancangan

Setelah melakukan tahap analisis kebutuhan perangkat lunak, maka dilanjutkan dengan perancangan perangkat lunak.

Berdasarkan data yang diperoleh, maka perancangan aplikasi *searching* SMS menggunakan metode perancangan berorientasi objek.

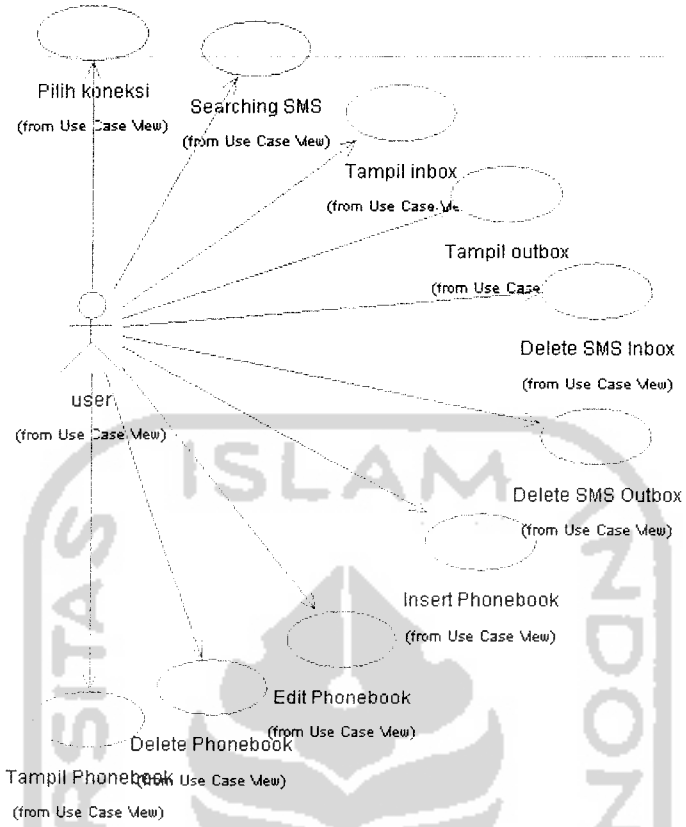
### 3.2.2 Hasil Perancangan

Hasil perancangan sistem ini dibedakan menjadi beberapa bagian sesuai dengan bagian-bagian yang digunakan pada metode perancangan berorientasi objek Perancangan aplikasi *searching* SMS menggunakan bahasa pemodelan UML (*Unified Modelling Language*). Hal ini dilakukan karena dengan menggunakan bahasa pemodelan UML dapat membantu dalam perancangan dan analisisa perangkat lunak yang akan dibangun.

Perancangan aplikasi *searching* SMS terdiri dari beberapa diagram antara lain *use case diagram, class diagram, sequence diagram, activity diagram.*

### 3.2.2.1 Use Case Diagram

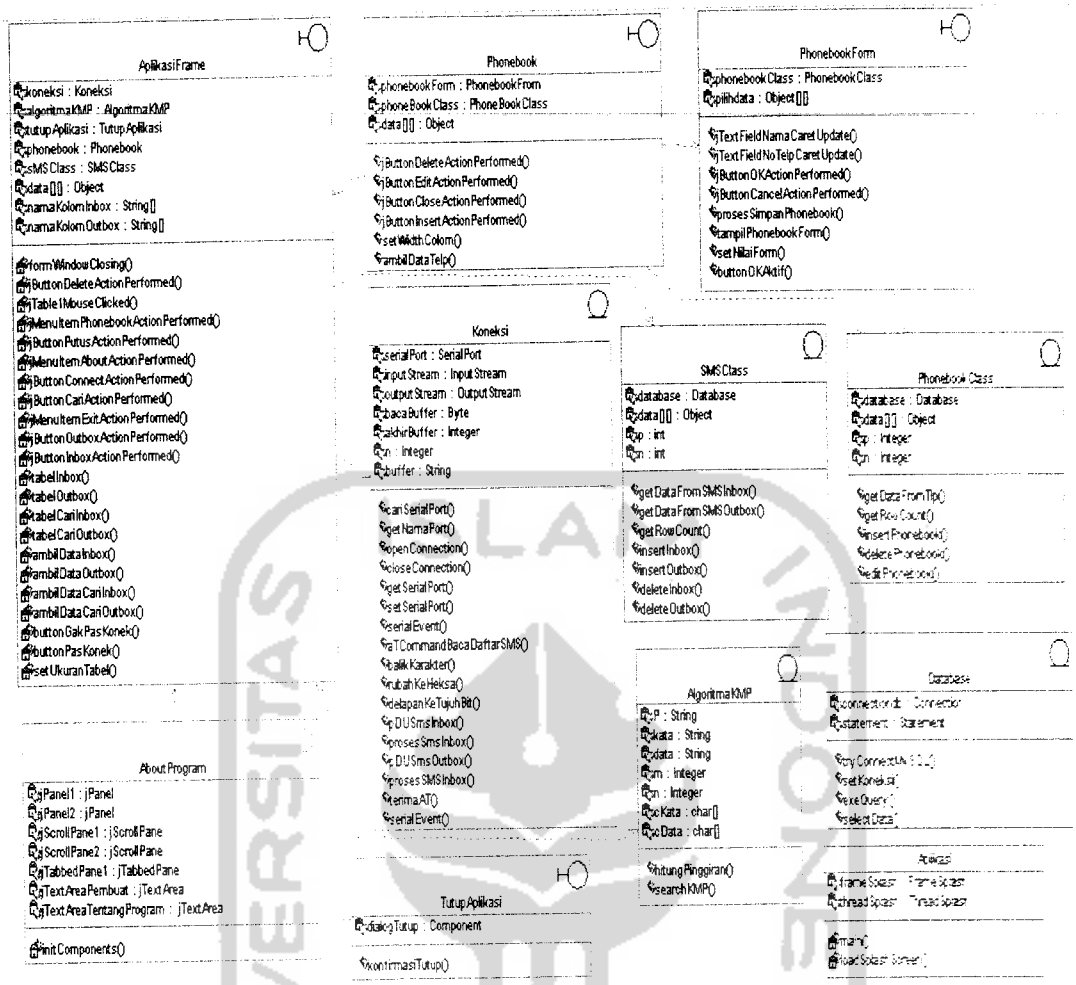
*Use case diagram* pada perancangan aplikasi *searching* SMS terdiri dari aktor user (pengguna aplikasi *searching* SMS) dan 10 *use case* yaitu *use case* pilih koneksi, *use case searching* SMS, *use case* tampil *inbox* dan *use case* tampil *outbox*, *use case delete* SMS *inbox*, *use case delete* SMS *outbox*, *use case insert phonebook*, *use case edit* phonebook, *use case delete phonebook*, *use case* tampil *phonebook*. *Use case diagram* pada perancangan aplikasi *searching* SMS ditunjukkan pada gambar 3.1




Gambar 3.1 Use case diagram

### 3.2.2.2 Class Diagram

Class pada aplikasi *searching* SMS terdiri dari beberapa *class* yang sesuai dengan fungsinya masing-masing. Adapun *class* yang terdapat pada aplikasi *searching* SMS antara lain *class* Aplikasi, *class* AplikasiFrame, *class* AboutProgram, *class* Koneksi, *class* SMSClass, *class* Phonebook, *class* PhonebookForm, *class* PhonebookClass, *class* AlgoritmaKMP, *class* Database, *class* TutupAplikasi. Pembuatan aplikasi *searching* SMS, *class diagram* ditunjukkan pada gambar 3.2.




Gambar 3.2 Class diagram

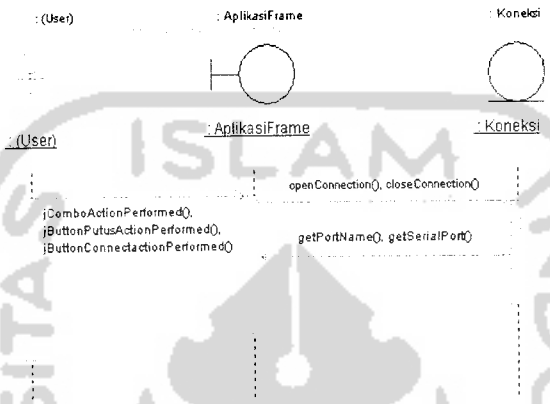
### 3.2.2.3 Sequence Diagram

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan sekitar sistem (termasuk pengguna, dan sebagainya) berupa *massage* yang digambarkan terhadap waktu. Tujuan penggunaan *sequence diagram* yaitu untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem.

Pada pembuatan aplikasi *searching* SMS, *sequence diagram* ditunjukkan sebagai berikut:

1. *Sequence diagram* pilih koneksi

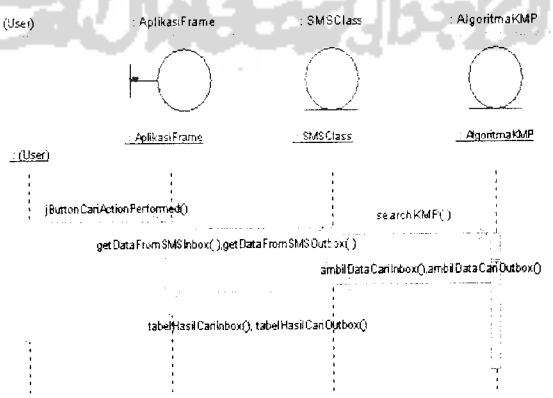
   *Sequence diagram* pilih koneksi digunakan untuk menggambarkan interaksi antar objek dalam melakukan koneksi dengan *port* komputer..




Gambar 3.3 Sequence diagram pilih koneksi

2. *Sequence diagram searching* SMS

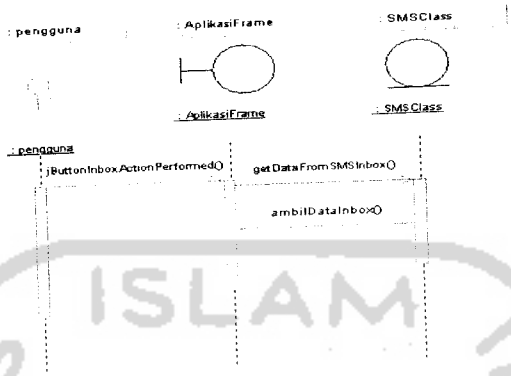
   *Sequence diagram searching* SMS digunakan untuk menggambarkan interaksi antar objek dalam melakukan pencarian SMS berdasarkan kata yang dimasukkan.




Gambar 3.4 Sequence diagram searching SMS

3. *Sequence diagram* tampil *inbox*

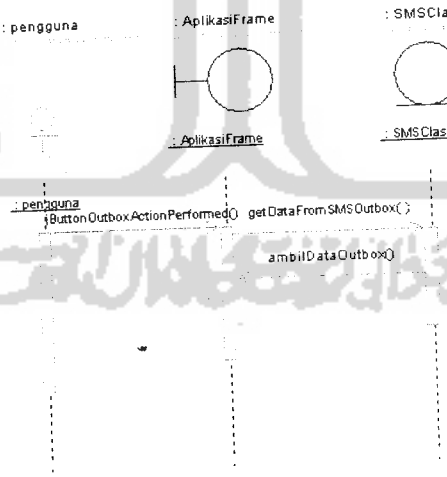
*Sequence diagram* tampil *inbox* digunakan untuk menggambarkan interaksi antar objek dalam menampilkan daftar SMS yang tersimpan dalam *inbox*.




Gambar 3.5 Sequence diagram tampil inbox

4. *Sequence diagram* tampil *outbox*

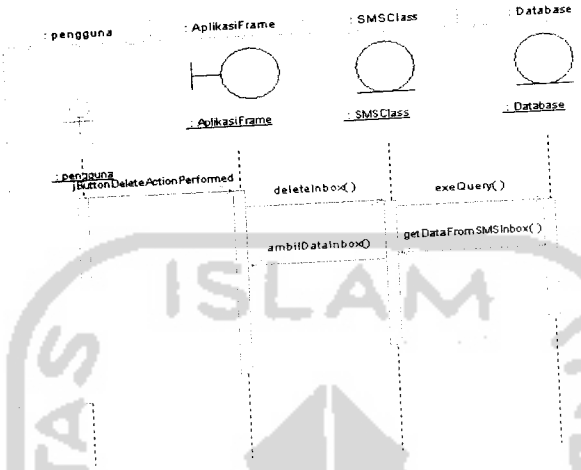
*Sequence diagram* tampil *outbox* digunakan untuk menggambarkan interaksi antar objek dalam menampilkan daftar SMS yang tersimpan dalam *outbox*.




Gambar 3.6 Sequence diagram tampil outbox

5. *Sequence diagram delete* SMS *inbox*

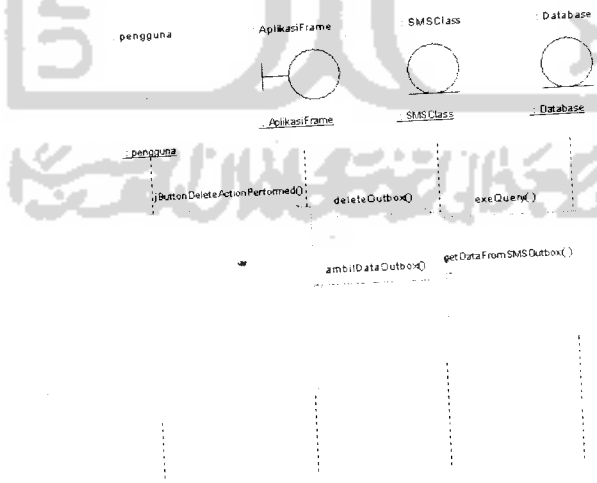
*Sequence diagram delete* SMS *inbox* digunakan untuk menggambarkan interaksi antar objek dalam menghapus SMS yang tersimpan di dalam *inbox*.




Gambar 3.7 Sequence diagram delete SMS inbox

6. *Sequence diagram delete* SMS *Outbox*

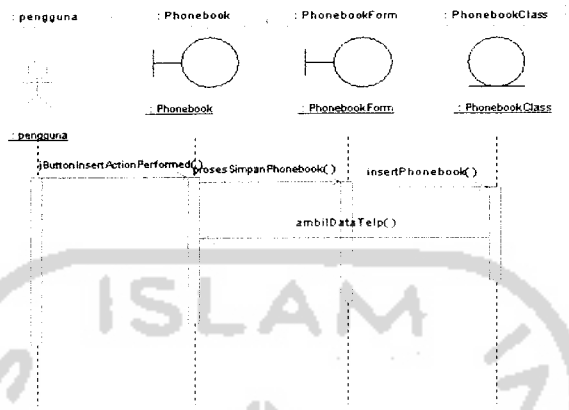
*Sequence diagram delete* SMS *Outbox* digunakan untuk menggambarkan interaksi antar objek dalam menghapus SMS yang tersimpan di dalam *outbox*.




Gambar 3.8 Sequence diagram delete SMS outbox

7. *Sequence diagram insert phonebook*

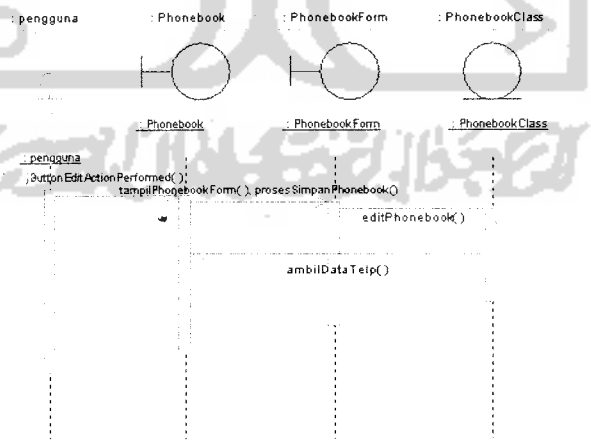
   *Sequence diagram insert phonebook* digunakan untuk menggambarkan interaksi antar objek dalam menambahkan data baru pada *phonebook.*




Gambar 3.9 Sequence diagram insert phonebook

8. *Sequence diagram edit phonebook*

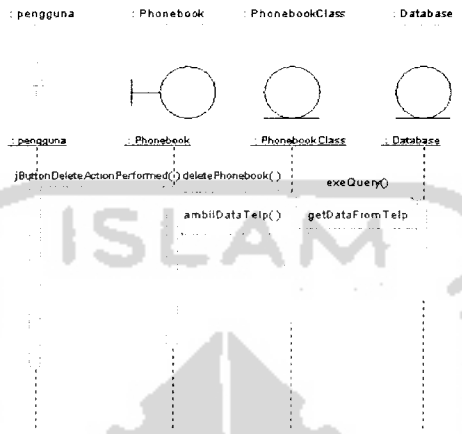
   *Sequence diagram searching* SMS digunakan untuk menggambarkan interaksi antar objek dalam mengubah daftar telepon yang telah tersimpan dalam *phonebook.*




Gambar 3.10 Sequence diagram edit phonebook

9. *Sequence diagram* delete *phonebook*

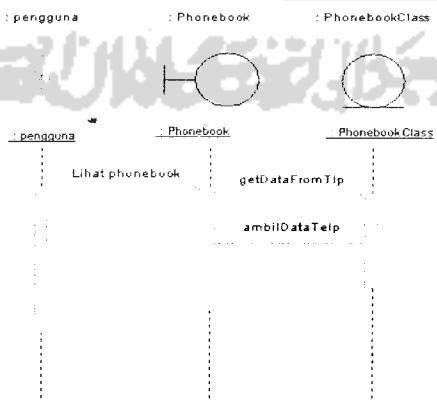
*Sequence diagram* delete *phonebook* digunakan untuk menggambarkan interaksi antar objek dalam menghapus data yang telah tersimpan pada *phonebook.*




Gambar 3.11 Sequence diagram delete phonebook

10. *Sequence diagram* tampil *phonebook*

*Sequence diagram* tampil *phonebook* digunakan untuk menggambarkan interaksi antar objek dalam menampilkan daftar *phonebook* yang tersimpan dalam basisdata.

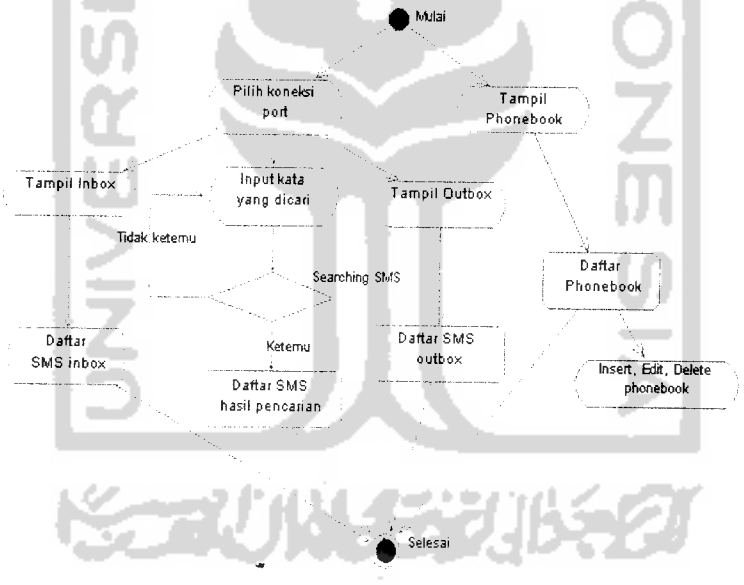



Gambar 3.12 Sequence diagram tampil phonebook

### 3.2.2.4 Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas *actor*, digunakan untuk mendeskrepsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case*. Pada umumnya *activity diagram* tidak menampilkan secara detail urutan proses, namun hanya memberikan gambaran global bagaimana urutan prosesnya. Sehingga sering kali diagram ini digunakan untuk memodelkan aktivitas bisnis dalam *level* konseptual [HEN06].

Pada pembuatan aplikasi *searching* SMS, *activity diagram* ditunjukkan sebagai berikut:




Gambar 3.13 Activity diagram

### 3.2.2.5 Perancangan Basisdata

Basisdata merupakan kumpulan dari tabel yang saling berhubungan yang satu dengan yang lainnya.. Dalam tabel memiliki kunci utama (*primary key*), kunci tersebut yang menghubungkan antar tabel. Tentunya tabel yang lain memiliki foreign key. Perancangan basisdata pada aplikasi *searching* SMS menggunakan sebuah basisdata bernama sms yang digunakan untuk menempatkan beberapa informasi yang mendukung aplikasi. Adapun basisdata sms terdiri dari tiga tabel antara lain tabel *inbox*, *outbox*, tabel *phonebook*.

#### 3.2.2.5.1 Struktur tabel

Tabel merupakan fasilitas yang tersedia dalam basisdata yang berfungsi untuk menyimpan data. Tabel terdiri dari baris dan kolom, dimana kolom merupakan *field* dan baris merupakan jumlah data yang terdapat dalam tabel atau disebut juga dengan *record*.

Tabel yang digunakan dalam perancangan basisdata pada aplikasi *searching* SMS sebagai berikut :

**1. Tabel inbox**

Tabel *inbox* digunakan untuk menyimpan daftar SMS *inbox*. Struktur tabel *inbox* dapat dilihat pada tabel 3.1 :

| No | Field | Tipe Data | Atribut | Keterangan |
|---|---|---|---|---|
| 1 | index_sms | INTEGER(8) | Primary | Index sms |
| 2 | no_telp | VARCHAR(16) | Foreign | No telepon |
| 3 | pesan | VARCHAR(160) | | Isi SMS |
| 4 | tanggal | TIMESTAMP | | Tanggal |

Tabel 3.1 Tabel Inbox

**2. Tabel Outbox**

Tabel *outbox* digunakan untuk menyimpan daftar SMS *outbox*. Struktur tabel *outbox* dapat dilihat pada tabel 3.2 :

| No | Field | Tipe Data | Atribut | Keterangan |
|---|---|---|---|---|
| 1 | index_sms | INTEGER(8) | Primary | Index sms |
| 2 | no_telp | VARCHAR(16) | Foreign | No telepon |
| 3 | pesan | VARCHAR(160) | | Isi SMS |
| 4 | tanggal | TIMESTAMP | | Tanggal |

Tabel 3.2 Tabel *outbox*

**3. Tabel Phonebook**

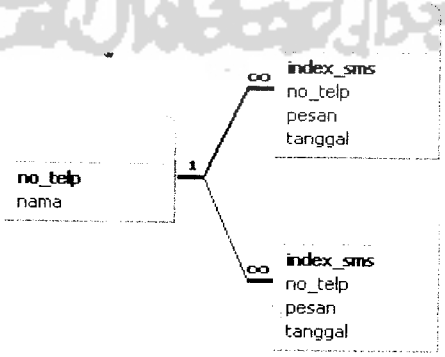
Tabel *phonebook* digunakan untuk menyimpan daftar *phonebook*. Struktur tabel *phonebook* dapat dilihat pada tabel 3.3 :

| No | Field | Tipe Data | Atribut | Keterangan |
|---|---|---|---|---|
| 1 | no_telp | INTEGER(8) | Primary | No telepon |
| 2 | nama | VARCHAR(16) | | Nama |

Tabel 3.3 Tabel *Phonebook*

**3.2.2.5.2 Relasi Tabel**

Relasi tabel pada perancangan basisdata dalam pembuatan aplikasi *searching* SMS ditunjukkan sebagai berikut :

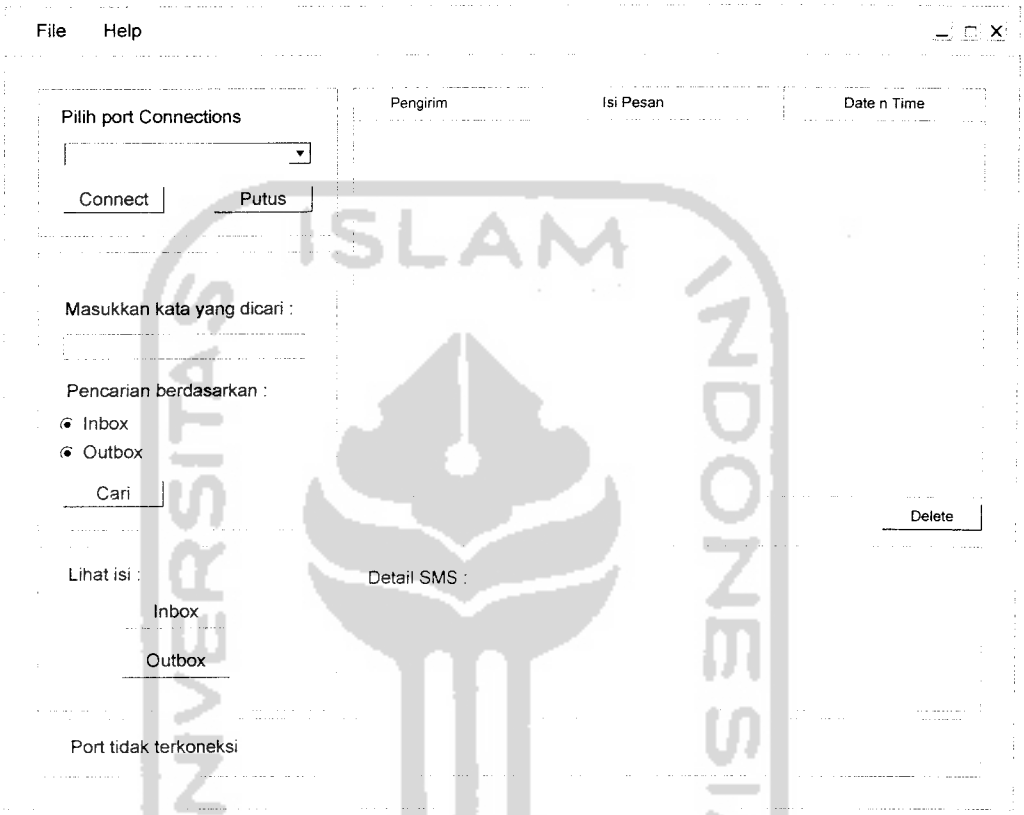



Tabel 3.4 Tabel relasi

### 3.2.2.6 Rancangan Antarmuka

#### 3.2.2.6.1 Rancangan antarmuka utama

Rancangan antarmuka halaman utama aplikasi *searching* SMS akan ditunjukkan pada gambar 3.14.




Gambar 3.14 Rancangan antarmuka halaman utama.

Rancangan antarmuka pada halaman utama terdiri dari:

1. Main menu

   Main menu dalam antarmuka utama aplikasi *searching* SMS terdiri dari :

   a. Menu file

   b. Menu *Phonebook*

   c. Menu help

2. Panel

Antarmuka aplikasi *searching* SMS terdiri dari enam panel utama yaitu :

a. Panel pilih koneksi

Panel pilih koneksi terdiri dari sebuah *combo box* yang berisi daftar port yang ada pada komputer, *button connect* yang berfungsi untuk mengoneksikan port berdasarkan pilihan user pada *combo box*, *button* putus yang berfungsi untuk memutus koneksi dengan serial port yang terkoneksi.

b. Panel pencarian

Panel ini terdiri dari sebuah *textfield* cari yang berfungsi untuk menerima *input* (berupa kata yang dicari) dari pengguna, sebuah *button* cari yang berfungsi untuk menjalankan proses pencarian, dua buah *radio button* yaitu *radio button inbox* dan *radio button outbox* yang berfungsi untuk memilih lokasi pencarian.

c. Panel lihat SMS

Panel lihat isi terdiri dari dua buah *button* yaitu *button inbox* dan *button outbox*. *Button inbox* digunakan untuk melakukan proses menampilkan daftar SMS di *inbox* dan *button outbox* digunakan untuk melakukan proses menampilkan daftar SMS di *outbox*.

d. Panel status koneksi.

Panel status koneksi terdiri dari sebuah label yang digunakan untuk menampilkan pesan berupa status koneksi dengan port. Apabila aplikasi terkoneksi dengan port maka label diset port terkoneksi, sebaliknya

apabila aplikasi tidak terkoneksi dengan port maka label diset port tidak terkoneksi.

e. Panel daftar SMS

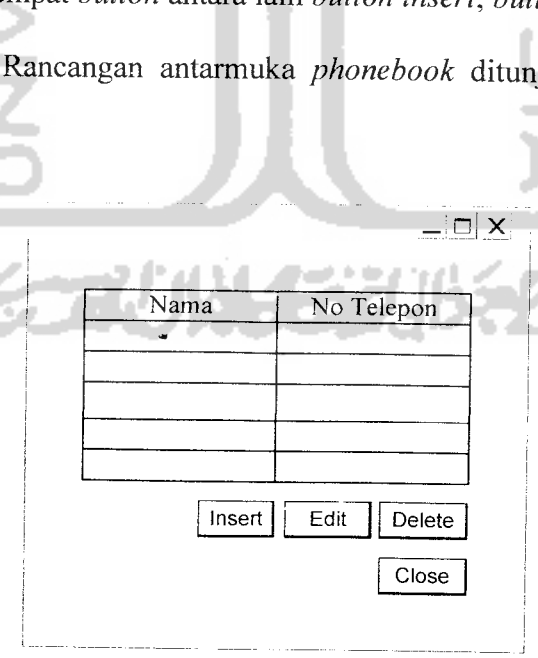
Panel ini terdiri dari sebuah tabel yang digunakan untuk menampilkan daftar SMS yang merupakan hasil proses pencarian sms, hasil lihat isi *inbox* dan hasil lihat isi *outbox*. Bentuk tabel daftar SMS meliputi pengirim, isi pesan dan tanggal.

f. Panel detail SMS

Panel ini digunakan untuk menampilkan detail isi SMS apabila SMS yang terdapat di tabel daftar SMS di pilih.

#### 3.2.2.6.2 Rancangan antarmuka phonebook

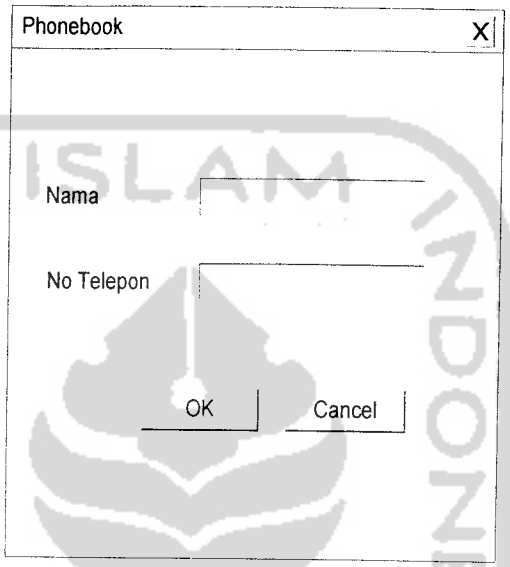
Rancangan antarmuka *phonebook* terdiri dari sebuah tabel yang berisi daftar *phonebook*, empat *button* antara lain *button insert*, *button edit*, *button delete* dan *button close*. Rancangan antarmuka *phonebook* ditunjukkan pada gambar 3.15.




Gambar 3.15 Rancangan antarmuka halaman *phonebook*

#### 3.2.2.6.3 Rancangan antarmuka form phonebook

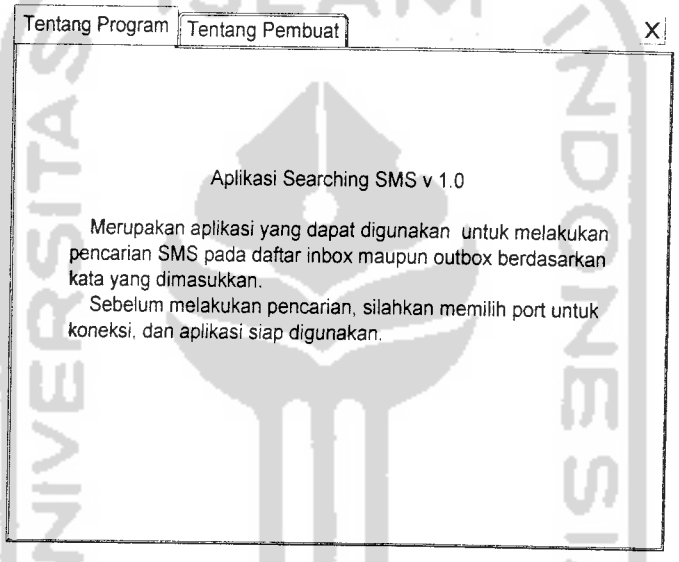
Rancangan antarmuka form *phonebook* berupa *dialog form*, dimana dalam antarmuka halaman ini terdapat dua buah editbox yang digunakan untuk menerima input dari user dan dua buah *button* konfirmasi. Rancangan antarmuka *help* ditunjukkan pada gambar 3.16.




Gambar 3.16 Rancangan antarmuka *form phonebook*

#### 3.2.2.6.4 Rancangan antarmuka help

Rancangan antarmuka *help* berupa *dialog form*, dimana dalam antarmuka halaman ini terdapat dua buah *tabbedpane* yaitu *tabbedpane* tentang program dan *tabbedpane* tentang pembuat. *tabbedpane* tentang program berisi keterangan tentang aplikasi *searching* SMS dan *tabbedpane* tentang pembuat berisi keterangan pembuat aplikasi. Pada halaman ini pengguna hanya dapat melihat saja. Antarmuka help ditunjukkan pada gambar 3.17.




Gambar 3.17 Rancangan antarmuka halaman *help*

# BAB IV

# HASIL DAN PEMBAHASAN

## 4.1 Implementasi Perangkat Lunak

Implementasi perangkat lunak merupakan tahapan yang bertujuan untuk memastikan apakah perangkat lunak yang dibangun dapat bekerja secara efektif dan efisien sesuai yang diharapkan.

Sebelum perangkat lunak digunakan dan diimplementasikan, maka perangkat lunak harus bebas dari kesalahan. Kesalahan yang mungkin terjadi antara lain kesalahan penulisan bahasa, kesalahan tampilan ataupun kesalahan proses. Setelah perangkat lunak bebas dari kesalahan, kemudian dapat dilakukan pengujian dengan menjalankan perangkat lunak.

### 4.1.1 Batasan Implementasi

Aplikasi yang dibuat adalah sebuah aplikasi desktop yang digunakan untuk melakukan pencarian SMS pada *inbox* maupun *outbox* berdasarkan masukan berupa kata yang akan dicari. Dalam proses pembuatan dan pengembangan aplikasi searching SMS pada desktop ini banyak ditemui asumsi baru yang lebih beragam dari batasan yang telah direncanakan, batasan-batasan itu antara lain:

1. Pencarian hanya pada daftar inbox atau outbox SMS.
2. Pengisian buku telepon diisi oleh pengguna.

3. Aplikasi digunakan pada ponsel GSM yang mendukung perintah AT Command.

### 4.1.2 Implementasi

Implementasi merupakan tahap di mana sistem siap dioperasikan pada keadaan yang sebenarnya, sehingga akan diketahui apakah sistem yang telah dibuat benar-benar sesuai dengan yang direncanakan. Pada implementasi perangkat lunak ini akan dijelaskan bagaimana program sistem ini bekerja, dengan memberikan tampilan *form-form* yang dibuat.

#### 4.1.2.1 Mekanisme penggunaan aplikasi

Mekanisme penggunaan aplikasi *searching* SMS dapat digambarkan sebagai berikut:

1. Pengguna melakukan koneksi dengan telepon seluler dengan cara memilih koneksi yang terdapat pada panel pilih koneksi.
2. Pengguna melakukan pencarian dengan memasukkan kata yang dicari dan memilih lokasi pencarian, dalam aplikasi ini lokasi pencarian yang dapat dipilih adalah lokasi pencarian berdasarkan *inbox* atau *outbox*.
3. Aplikasi mengolah *input* dari user yang berupa kata dengan melakukan pencarian pada lokasi yang dipilih oleh user.
4. Pengguna aplikasi ini dapat melihat daftar SMS yang terdapat dalam *inbox* maupun *outbox*.

#### 4.1.2.2 Implementasi Antarmuka

##### 4.1.2.2.1 Antarmuka Utama

Antarmuka halaman utama aplikasi searching SMS ditunjukkan pada gambar 4.1. Antarmuka halaman utama adalah tampilan pertama kali dari aplikasi *searching* SMS setelah pengguna menjalankan aplikasi. Antarmuka halaman utama ini bersifat *user frendly* dan interaktif. Hal ini dilakukan agar pengguna dapat mudah menggunakan aplikasi *searching* SMS.

Pada halaman utama aplikasi *searching* SMS terdiri dari beberapa komponen antara lain tiga menu dan enam panel, dimana masing-masing komponen memiliki fungsi yang berbeda-beda. Adapun fungsi dari masing-masing panel adalah sebagai berikut :

1. Panel pilih koneksi

   Panel pilih koneksi digunakan pengguna untuk memilih koneksi port yang akan menghubungkan komputer dengan telepon seluler. Pada panel pilih koneksi, port yang ditampilkan bersifat dinamis sesuai dengan jumlah serial port yang terdapat di komputer dan kabel data telepon seluler yang digunakan.

2. Panel pencarian

   Panel pencarian digunakan pengguna untuk melakukan pencarian SMS dengan memasukkan data berupa kata yang dicari. Pengguna dapat memilih lokasi pencarian SMS berdasarkan SMS yang tersimpan di *inbox* maupun berdasarkan SMS yang tersimpan di *outbox* dengan memilih salah satu dari *radio button* yang tersedia. Apabila pengguna ingin mencari SMS yang terdapat kata yang dicari, maka *user* menekan *button* cari.

Apabila pada lokasi yang dipilih (*inbox* atau *outbox*) terdapat SMS yang didalamnya terdapat kata yang dimasukkan, maka SMS yang terdapat kata yang dicari akan ditampilkan pada tabel hasil pencarian. Apabila dalam pencarian SMS tidak terdapat SMS yang didalamnya terdapat kata yang dicari, aplikasi akan memberi informasi kepada pengguna bahwa SMS yang dicari tidak ditemukan.

3. Panel lihat SMS

Panel lihat SMS digunakan untuk melihat SMS yang tersimpan dalam *inbox* telepon seluler maupun *outbox* telepon seluler. Pengguna dapat melihat SMS yang terdapat dalam *inbox* dengan menekan *button inbox* dan melihat SMS yang terdapat dalam *outbox* dengan menekan *button outbox*. Hasil dari lihat isi *inbox* maupun isi *outbox* ditampilkan pada tabel panel daftar SMS.

4. Panel status koneksi

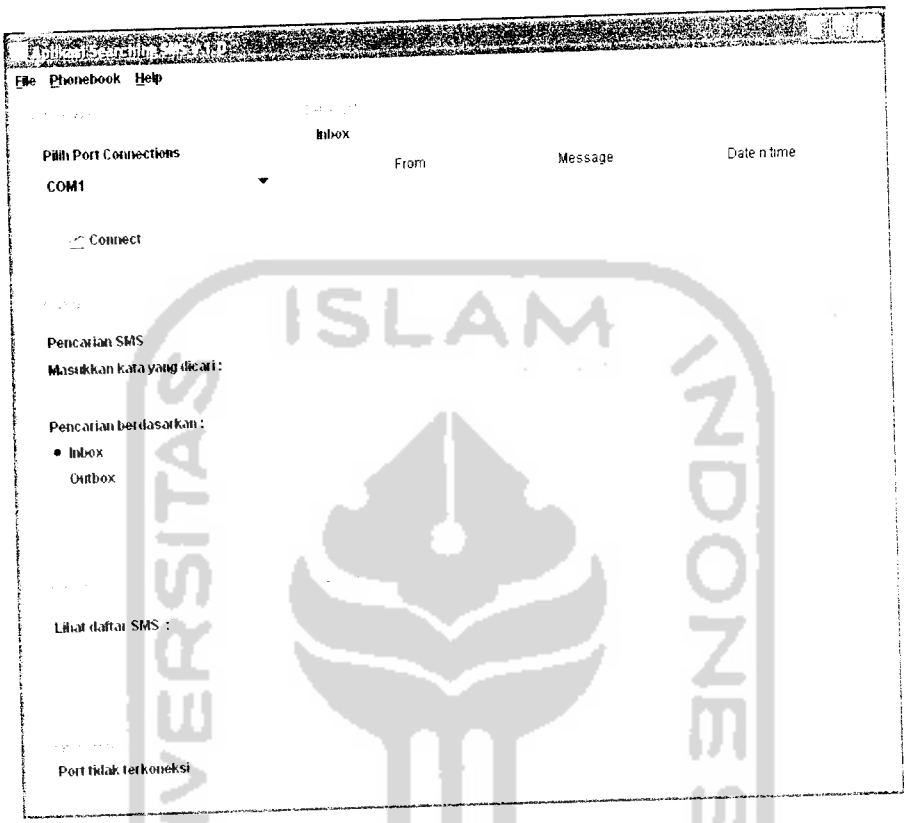
Panel ini digunakan untuk menampilkan pesan berupa status koneksi dengan port. Apabila aplikasi *searching* SMS terkoneksi dengan *serial port*, maka panel ini akan menampilkan pesan berupa status koneksi dengan port. Sebaliknya apabila aplikasi ini tidak terkoneksi dengan serial port maka panel ini akan memberitahukan bahwa aplikasi ini tidak terkoneksi dengan *serial port*.

5. Panel daftar SMS

Panel ini digunakan untuk menampilkan daftar SMS *inbox*, SMS *outbox* serta daftar sms hasil pencarian baik pencarian berdasarkan *inbox* maupun *outbox*. Pada panel ini, pengguna dapat menghapus SMS dengan menekan *button delete*.

6. Panel detail SMS

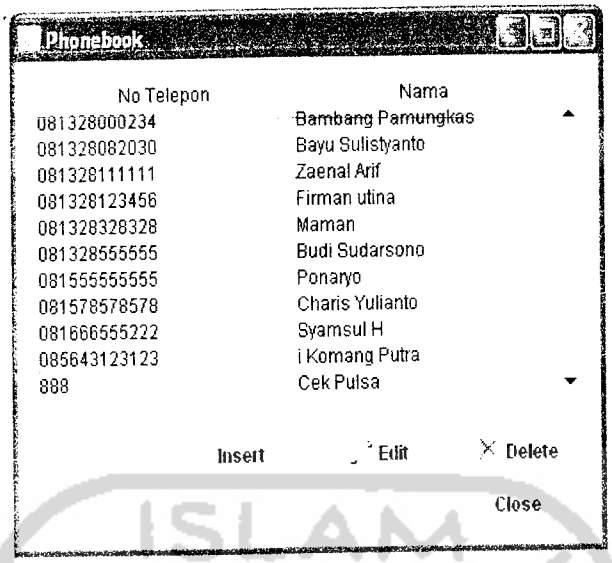
Panel ini digunakan untuk menampilkan detail isi pesan apabila sms yang terdapat dalam panel daftar sms ditekan.




Gambar 4.1 Antarmuka halaman utama

### 4.1.2.2.2 Antarmuka Phonebook

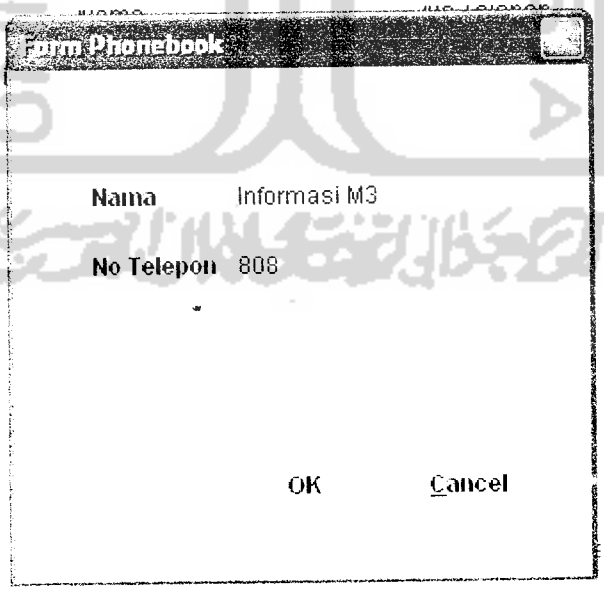
Antarmuka *phonebook* digunakan untuk manampilkan daftar telepon yang tersimpan. Pada antarmuka ini, pengguna dapat menambah, mengubah serta menghapus daftar telepon. Antarmuka *phonebook* ditunjukkan pada gambar 4.2.




Gambar 4.2 Antarmuka *phonebook*

#### 4.1.2.2.3 Antarmuka Form Phonebook

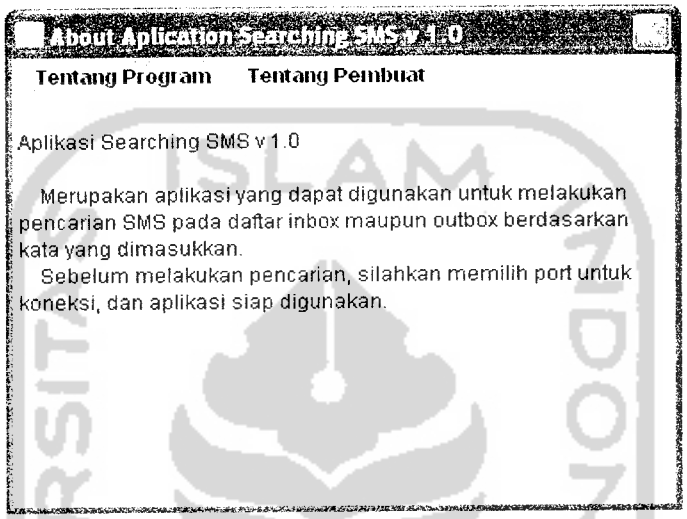
Pengguna dapat menambah daftar baru maupun mengubah daftar *phonebook* dengan mengisikan no telepon dan nama pada *form* yang disediakan. Antarmuka *form phonebook* ditunjukkan pada gambar 4.3.




Gambar 4.3 Antarmuka *form phonebook*

#### 4.1.2.2.4 Antarmuka About

Antarmuka *about* digunakan untuk menampilkan informasi tentang aplikasi *searching* SMS dan informasi tentang pembuat aplikasi. Antarmuka *about* ditunjukkan pada gambar 4.4.




Gambar 4.4 Antarmuka about

### 4.1.2.3 Implementasi prosedural

Aplikasi *searching* SMS ini dibangun dengan menggunakan beberapa prosedur yang bertujuan agar aplikasi dapat dijalankan dan bekerja sesuai tujuan. Adapun prosedur inti dalam pembuatan aplikasi searching SMS antara lain :

**1. Prosedur koneksi port**

Prosedur koneksi *port* digunakan untuk melakukan koneksi antara telepon seluler dengan komputer. Koneksi dilakukan dengan mengatur nilai parameter input yaitu nilai *serial port*, nilai *Boud Rate(bps),* nilai *databits*, nilai *stopbits*, nilai *parity* dan nilai *flow control*.

```
public boolean openConnection(String namaPortSerial){
    boolean status=false;
    Enumeration daftarPort=CommPortIdentifier.getPortIdentifiers();
    while(daftarPort.hasMoreElements()){
        CommPortIdentifier portId=(CommPortIdentifier)
daftarPort.nextElement();
        if(portId.getPortType()==CommPortIdentifier.PORT_SERIAL){
            if(portId.getName().equals(namaPortSerial)){
                try {
                    setSerialPort((SerialPort)
portId.open("ATCommand",5000));
                    JOptionPane.showMessageDialog(null,"Port
"+namaPortSerial+" berhasil terkoneksi");
                status=true;
                } catch (PortInUseException piue) {
                    JOptionPane.showMessageDialog(null,"       Koneksi
port gagal \nPort "+namaPortSerial+" sedang digunakan ");
                    status=false;
                }
            }
        }
    }
    try{
        inputStream=getSerialPort().getInputStream();
        outputStream=getSerialPort().getOutputStream();
    }catch (IOException e){
        JOptionPane.showMessageDialog(null," Error input output
stream");
    }
    try{

getSerialPort().setFlowControlMode(getSerialPort().FLOWCONTROL_NONE);

getSerialPort().setSerialPortParams(19200,getSerialPort().DATABITS_8,g
etSerialPort().STOPBITS_1,getSerialPort().PARITY_NONE);
        getSerialPort().notifyOnDataAvailable(true);
        //kebanyakan hp support boudrate 19200, databits8, stopbits 1,
parity none
    }catch (UnsupportedCommOperationException e){
        JOptionPane.showMessageDialog(null,"error serial port " +
e.getMessage());
    }
    try{
        getSerialPort().addEventListener(this);
    }catch(TooManyListenersException e){
        JOptionPane.showMessageDialog(null,"getserial port " +
e.getMessage());
    }
    // Mencoba mendapatkan InputStream dan OutputStream
    try {
        inputStream = serialPort.getInputStream();
        outputStream = serialPort.getOutputStream();
    }
    // Apabila nilai InputStream dan OutputStream gagal didapatkan
    catch (IOException ioe) {
        JOptionPane.showMessageDialog(null,"Ada kesalahan.\n" + ioe);
    }
    return status;
}
```

## 2. Prosedur AT Command baca daftar SMS di telepon seluler

Prosedur ini digunakan untuk melakukan pengiriman AT Command membaca daftar SMS yang tersimpan dalam telepon seluler.

```
public void aTCommandBacaDaftarSMS(){
        try{
            outputStream.write(("AT+CMGF=0" + "\15").getBytes());
            outputStream.flush();
            outputStream.write(("AT+CMGL=4" + "\15").getBytes());
            outputStream.flush();
        }catch(IOException ioe){
            JOptionPane.showMessageDialog(null,"send at command : " +
ioe.getMessage());
        }
    }
```

## 3. Prosedur menampilkan daftar SMS

Prosedur ini digunakan untuk menampilkan daftar SMS baik daftar SMS inbox, outbox, maupun hasil pencarian.

```
public void tabelInbox(){
        ambilDataInbox();
        jTable1.setModel(new
javax.swing.table.DefaultTableModel(data,namaKolomInbox) {
            Class[] types = new Class[]{

java.lang.Object.class,java.lang.Object.class,java.lang.Object.class
            };
            public Class getColumnClass(int rowIndex,int columnIndex)
{
                return types [columnIndex];
            }
            boolean[] canEdit = new boolean [] {
                false, false, false
            };
            public boolean isCellEditable(int rowIndex, int
columnIndex) {
                return canEdit [columnIndex];
            }
        });
        setUkuranTabel();
    }

// Mengambil data dari Inbox
    private void ambilDataInbox() {
        data = new Object[sMSClass.getRowCount()][4];
        data = (Object[][]) sMSClass.getDataFromSMSInbox();
    }
```

## 4. Prosedur pencarian kata

Prosedur pencarian kata merupakan prosedur yang dibuat dengan menggunakan algoritma Knuth Morris Pratt. Prosedur ini digunakan dalam melakukan pencarian kata.

```
public boolean searchKMP(String kata, String data) {
        boolean status = false;
        int n = data.length();
        int m = kata.length();
        katachar = kata.toCharArray();

        int[] geser = hitungPinggiran(kata);

        int i = 0; // i = geser skr
        int j = 0; // j = posisi kata yang dicocokan
        while (i + m <= n) {

            boolean cari = true;
            while (cari) {
                if (j == m) cari = false;
                else {
                 if (data.charAt(i+j) != kata.charAt(j)) {
                 cari = false;
                      } else j++;
                }
            }
            if (j == m) {
                return true;
            }
            if (j == 0)
                i = i + 1;
            else {
                // kata[0 .. j-1] ktm data[i .. i+j-1]
                i = i + geser[j-1]; // i dtambah
                j = Math.max(j - geser[j-1],0);
                // set j pada posisi pertama kata
            }
        }
        return false;
    }
```

**5. Prosedur koneksi basis data**

Prosedur ini digunakan untuk melakukan koneksi dengan basisdata aplikasi.

```
public Connection tryConnectMySQL(){
        try {
          Class.forName("org.gjt.mm.mysql.Driver");
        } catch(ClassNotFoundException cnfe) {
              System.err.println("Error Loading Driver: "+cnfe);
        }
         try{
          String urldb="jdbc:mysql://localhost:3306/sms";
          connectiondb = DriverManager.getConnection(urldb,"root","");
        } catch(Exception e){
          JOptionPane.showMessageDialog(null,"Error Database: Tidak
          dapat konek ke Basis data");
        }
        return connectiondb;
    }   //Akhir method tryConnectMySQL
```

**6. Prosedur hapus data**

Prosedur ini digunakan untuk menghapus data dari basis data aplikasi.

```
//apabila yang didelete merupakan tabel outbox

if(jLabelStatusDaftarSMS.getText().equals("Outbox")){
       String[] values=new String[jTable1.getSelectedRowCount()];
       for(int i=0;i<values.length;i++){

       values[i]=String.valueOf(data[jTable1.getSelectedRows()[i]][3])
       ;
       sMSClass.deleteOutbox(values);
}
tabelOutbox();
```

## 4.2 Analisis Kerja Perangkat Lunak

### 4.2.1 Pengujian Perangkat Lunak

Pengujian perangkat lunak diperlukan dalam membangun sebuah perangkat lunak. Hal ini dilakukan untuk mendeteksi kesalahan-kesalahan yang mungkin terdapat dalam perangkat lunak yang dibuat. Kesalahan yang ada pada perangkat lunak selanjutnya diperbaiki agar perangkat lunak yang dibuat dapat

bekerja secara maksimal. Pengujian dilakukan pada setiap fungsi yang ada pada aplikasi dengan memasukkan data-data yang tidak *valid* agar dapat diketahui bagaimana sistem dalam menangani data-data yang tidak *valid* atau kesalahan-kesalahan yang mungkin dilakukan oleh pengguna sistem. Materi pengujian meliputi komponen masukan, proses dan keluaran.

Pengujian yang dilakukan pada aplikasi *searching* SMS mencakup kemampuan sistem dalam merespon setiap tindakan pengguna, mencegah terjadinya kesalahan, serta menangani apabila terjadi kesalahan (*error handling*) yang disebabkan oleh pengguna sistem atau faktor manusia (*human factor*).

Pengujian pada sistem terbagi atas dua cara, yaitu dengan prosedur normal dan dengan prosedur tidak normal. Berikut contoh pengujian yang dilakukan terhadap beberapa proses yang ada pada sistem, antara lain :

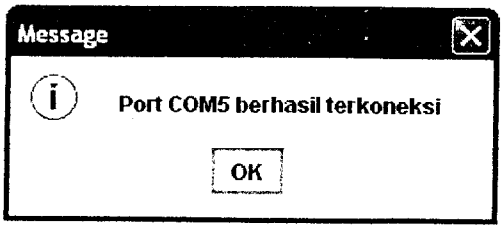
**1. Proses Koneksi Port**

Pengujian koneksi port dilakukan pada keadaan normal dan pada keadaan tidak normal. Pengujian dilakukan dengan memilih port COM5 dan menekan *button connect* untuk melakukan koneksi dengan telepon seluler.

a. Pengujian Normal

Pengujian normal dilakukan pada kondisi tidak ada aplikasi lain yang melakukan koneksi dengan port COM5.

Pada pengujian normal, aplikasi searching SMS berhasil melakukan koneksi dengan port COM5. Pesan yang muncul ketika koneksi *port* berhasil, ditunjukkan pada gambar 4.5

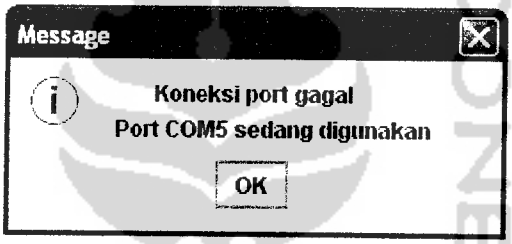



Gambar 4.5 Pesan koneksi *port* berhasil

b. Pengujian tidak normal

Pengujian tidak normal dilakukan pada kondisi dimana *port* COM5 sedang digunakan oleh aplikasi lain.

Pada pengujian tidak normal koneksi dengan port COM5 gagal, karena port sedang digunakan aplikasi lain. Pesan yang muncul ketika koneksi *port* tidak berhasil ditunjukkan pada gambar 4.6




Gambar 4.6 Pesan koneksi port gagal

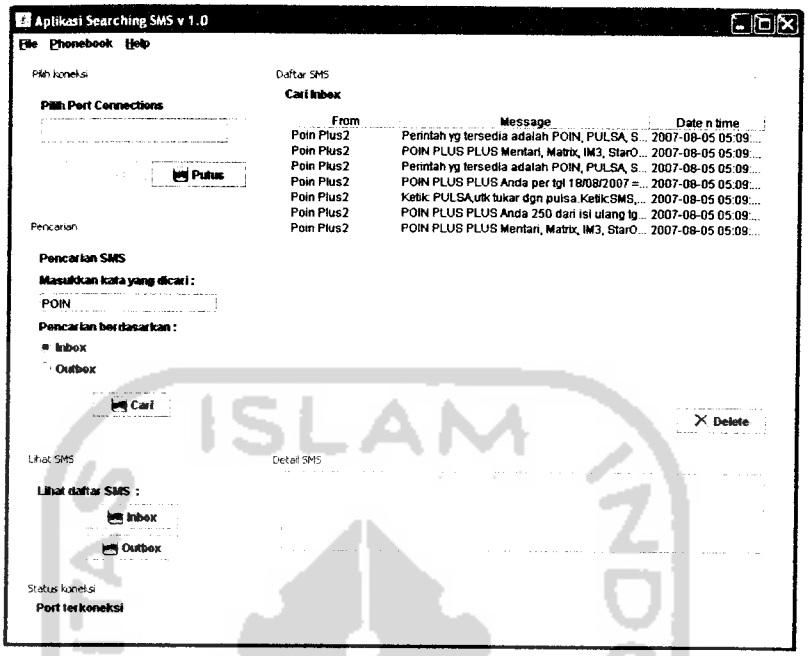
**2. Proses pencarian SMS**

Pengujian pencarian SMS dilakukan pada keadaan normal dan pada keadaan tidak normal.

a. Pengujian Normal

Pengujian pada keadaan normal dilakukan dengan memasukkan kata "POIN", memilih pencarian berdasarkan *inbox* dan menekan *button* cari yang terdapat pada panel pencarian. Pada pengujian kedua *output* yang dihasilkan

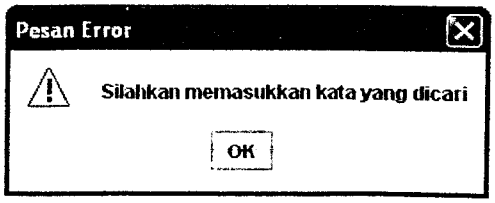
adalah daftar SMS yang terdapat kata POIN yang tersimpan dalam *inbox*. Hasil pengujian pencarian SMS ditunjukkan pada gambar 4.7.



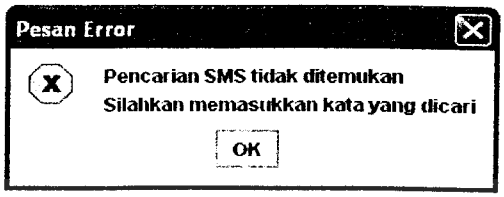

Gambar 4.7 Hasil pengujian pencarian SMS

b. Pengujian tidak normal

Pengujian dilakukan dengan memilih pencarian berdasarkan outbox dan menekan button cari tanpa memasukkan kata. Pada pengujian ini muncul pesan kesalahan seperti yang ditunjukkan pada gambar 4.8. Selanjutnya pengujian dilakukan dengan memasukkan kata JAPAN , memilih pencarian berdasarkan *outbox* dan menekan *button* cari yang terdapat pada panel pencarian. Pada pengujian pertama hasil pencarian tidak ditemukan dalam daftar SMS *outbox* dan muncul pesan pencarian tidak ditemuan. Pesan pencarian tidak ditemukan ditunjukkan pada gambar 4.9



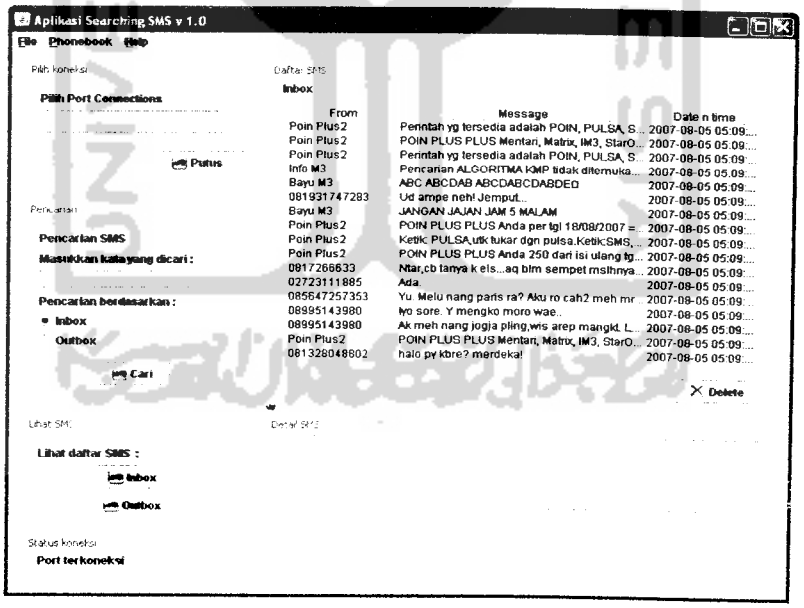

Gambar 4.8 Pesan *input* belum diisi




Gambar 4.9 Pesan hasil pencarian tidak ditemukan

### 3. Proses lihat SMS inbox

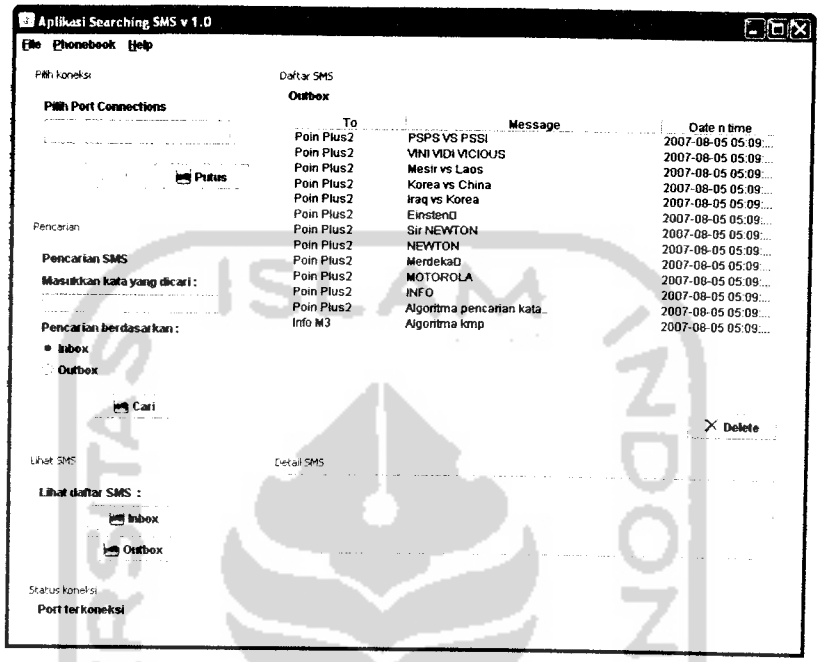
Pengujian dilakukan dengan menekan *button inbox* yang terdapat pada panel lihat SMS. *Output* yang dihasilkan adalah daftar SMS yang terdapat dalam *inbox*. Hasil pengujian tampil inbox ditunjukkan pada gambar 4.10.




Gambar 4.10 Hasil pengujian tampil *inbox*

**4. Proses lihat SMS outbox**

Pengujian dilakukan dengan menekan *button outbox* yang terdapat pada panel lihat SMS. Output yang dihasilkan adalah daftar SMS yang terdapat dalam *outbox*. Hasil pengujian tampil *outbox* ditunjukkan pada gambar 4.11.



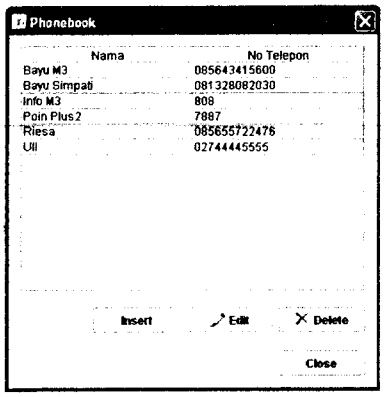

Gambar 4.11 Hasil pengujian tampil outbox

**5. Proses tambah phonebook**

Pengujian tambah *phonebook* dilakukan pada dua keadaan, yaitu pada keadaan normal dan pada keadaan tidak normal.

a. Pengujian Normal

Pengujian pada keadaaan normal dilakukan dengan memasukkan semua data yaitu no telepon dan nama. Apabila pengguna telah mengisikan data yang diminta maka akan muncul *phonebook* seperti yang ditunjukkan pada gambar 4.12




Gambar 4.12 Konfirmasi tambah *phonebook*

b. Pengujian tidak normal

Pengujian pada keadaaan tidak normal dilakukan dengan mengisikan data berupa huruf (bukan angka) ke *form* no telepon. Pada pengujian ini muncul pesan seperti yang ditunjukkan pada gambar 4.13

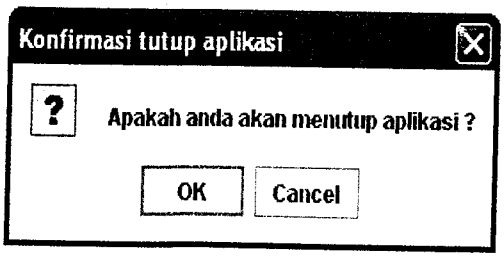



Gambar 4.13 Pesan data belum diisi

## 6. Proses Tutup Aplikasi

Apabila pengguna akan menutup aplikasi *searching* SMS maka akan muncul pesan konfirmasi berupa pilihan Ok dan *cancel*. Apabila pengguna

memilih *button* OK maka aplikasi akan ditutup, sebaliknya apabila pengguna memilih *button cancel* maka aplikasi searching SMS masih aktif. Konfirmasi tutup aplikasi ditunjukkan pada gambar 4.14.

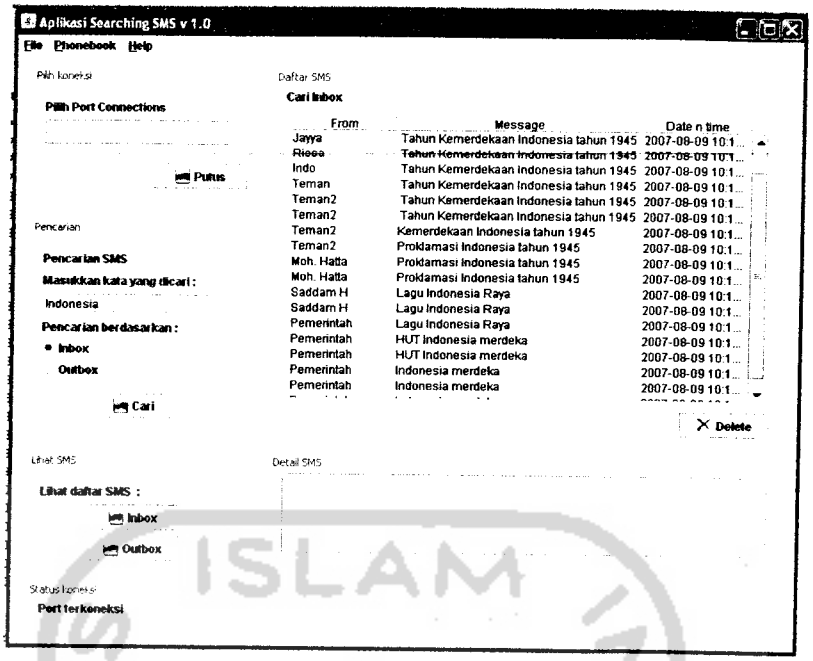



Gambar 4.14 Konfirmasi tutup aplikasi

### 4.2.2 Pengujian waktu pencarian

Pengujian dilakukan pada aplikasi yang telah dibangun dengan menghitung lama waktu pencarian SMS berdasarkan jumlah SMS yang tersimpan dalam daftar SMS.

Pada pengujian waktu pencarian ini, jumlah SMS yang tersimpan dalam daftar SMS *inbox* adalah 200 SMS dengan SMS yang terdapat kata Indonesia sebanyak 20 SMS. Pengujian dilakukan dengan memasukkan kata "Indonesia" pada *textfield* cari dan memilih lokasi pencarian *inbox*. Hasil dari pengujian yaitu pada daftar SMS menampilkan daftar SMS yang didalamnya ada kata "Indonesia" sebanyak 20 SMS dangan waktu yang dibutuhkan aplikasi dalam melakukan pencarian SMS yaitu 0.2 detik. Hasil pengujian waktu pencarian dapat dilihat pada gambar 4.15.




Gambar 4.15 Gambar pengujian waktu pencarian SMS

### 4.2.3 Analisis hasil pengujian

Berdasarkan hasil pengujian yang dilakukan diperoleh kesimpulan mengenai kinerja sistem Aplikasi *searching* SMS dengan menggunakan algoritma KMP sebagai berikut :

1. Sistem mampu melakukan pencarian SMS berdasarkan kata yang dimasukkan dengan menggunakan metode pencarian yang tersedia pada aplikasi, serta menampilkan daftar SMS hasil pencarian tersebut sesuai dengan istilah yang telah dipilih pengguna.
2. Sistem mampu membaca daftar SMS *inbox* maupun *outbox* dari telepon seluler dan menyimpannya dalam basis data aplikasi.

3. Sistem telah mampu menambah data *phonebook* yang diinputkan oleh pengguna, serta menyimpannya ke dalam basis data aplikasi.
4. Sistem telah mampu menyimpan hasil perubahan data *phonebook* ke dalam basis data aplikasi sesuai dengan data *phonebook* yang diubah oleh pengguna.
5. Sistem telah mampu menghapus data SMS maupun data *phonebook* dari basisdata aplikasi sesuai dengan data yang telah dipilih oleh pengguna.
6. Sistem telah mampu merespon tindakan yang telah dilakukan pengguna dengan menampilkan kotak pesan yang berisi informasi tentang berhasil atau tidaknya sistem memproses perintah yang telah diberikan pengguna.
7. Sistem telah mampu mencegah kesalahan oleh pengguna dengan menampilkan kotak pesan yang berisi permintaan konfirmasi kepada pengguna mengenai tindakan yang akan dilakukan pengguna selanjutnya.

# BAB V

# KESIMPULAN DAN SARAN

## 5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat diperoleh kesimpulan sebagai berikut:

1. Penggunaan algoritma Knuth Morris Pratt (KMP) pada aplikasi *searching* SMS dapat membantu pencarian isi SMS berdasarkan kata yang dimasukkan.
2. Aplikasi searching SMS dapat memudahkan pengguna telepon seluler dalam melakukan pencarian dalam daftar SMS berdasarkan kata yang dimasukkan ke sistem.
3. Perancangan sistem dengan menggunakan metode berorientasi objek dapat mempermudah dalam pengembangan sistem secara keseluruhan, sebagai contoh ketika ingin menambahkan suatu method, cukup dengan menambahkan method tersebut ke dalam *class*. Demikian pula ketika diimplementasikan dalam kode program, method tersebut cukup ditambahkan ke dalam *class* yang sesuai.

## 5.2 Saran

Mengingat berbagai keterbatasan yang dialami penulis terutama masalah pemikiran dan waktu, beberapa saran yang dapat diberikan untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Perlu pengembangan aplikasi searching SMS yang lebih luas pada perangkat lainnya, misalkan : pengembangan aplikasi pada PDA.
2. Menu dalam aplikasi searching SMS masih relatif sederhana, sehingga perlu adanya pengembangan menu yang lain agar lebih bermanfaat bagi pengguna, seperti untuk mendeteksi SIM Card, mendeteksi merk telepon seluler, dan sebagainya.

# DAFTAR PUSTAKA


[ADY02] Ady Wicaksono, *Pemrograman Aplikasi Wireless dengan Java*, Jakarta : Elex Media Computindo, 2002.

[ATM06] Alsasian Atmopawiro, *Pengkajian dan Analisis Tiga Algoritma Efisien Rabin-Karp, Knuth-Morris-Pratt, dan Boyer-Moore dalam Pencarian Pola dalam Suatu Teks*, Bandung : Laboratorium Ilmu dan Rekayasa Komputasi Departemen Teknik Informatika, ITB, 2006.

[EKA06] Gahayu Handari Ekaputri, Yulie Anneria Sinaga, *Aplikasi Algoritma Pencarian String Knuth-Morris-Pratt dalam Permainan Word Search*, Bandung : Laboratorium Ilmu dan Rekayasa Komputasi Departemen Teknik Informatika, ITB, 2006.

[HAR03] A Aditya Hartanto, *Java 2 Micro Edition Mobile Interface Device Programming*, Jakarta : Elex Media Computindo, 2003.

[HEN05] Arnoud Hendry, Vincent Jonack, *SMS and MMS Interworking in Mobile Networks*, TeAM YYePG, 2005.

[HEN06] Hendrik, *Modul Praktikum Rekayasa Perangkat Lunak*, Yogyakarta : Lab Sirkel, Teknik Informatika UII, 2006.

[LEC06] Lecroq, Thierry, Christian Charras, *Handbook of Exact String-Matching Algoritms, 2006.*

[MUN06] Rinaldi Munir, *Diktat Kuliah Strategi Algoritmik*, Bandung : Program Studi Teknik Informatika ITB, 2006.

[WAH05] Wahana Komputer, *Pengembangan Aplikasi Sistem Informasi Akademik Berbasis SMS dengan Java,* Jakarta : Salemba Infotek, 2005.