BAB II

LANDASAN TEORI

II.1 Teori Sistem Informasi

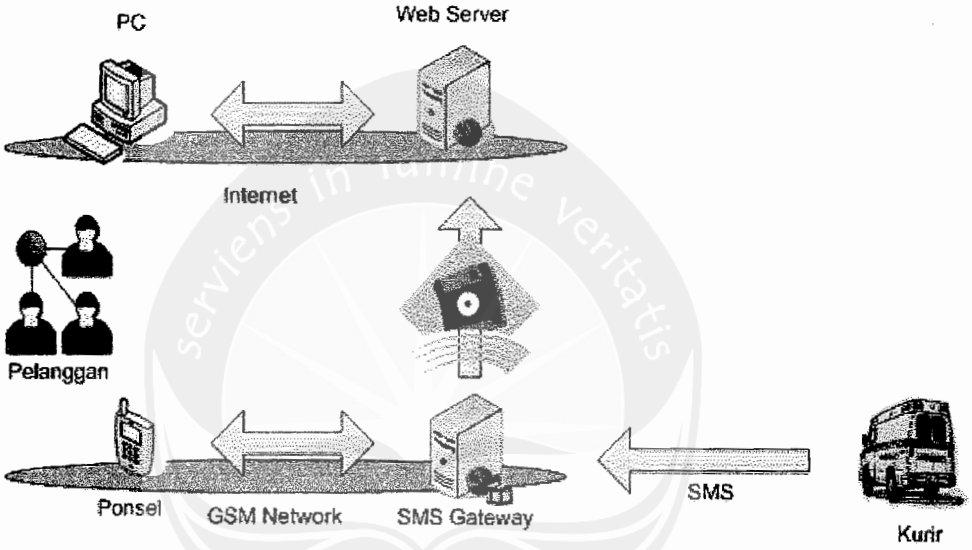
Pengertian sistem informasi tidak bisa dilepaskan dari pengertian **sistem** dan **informasi.** Definisi dari **Sistem** adalah beberapa set komponen yang bekerja bersama-sama untuk menampilkan suatu tugas. Sedangkan definisi dari **Informasi** adalah data yang diambil kembali, diolah, atau sebaliknya digunakan sebagai dasar untuk peramalan atau pengambilan keputusan. Sumber dari informasi adalah data. Data adalah fakta dan angka yang tidak sedang digunakan pada proses keputusan, dan biasanya berbentuk catatan historis yang dicatatkan dan diarsipkan tanpa maksud untuk segera diambil kembali untuk pengambilan keputusan. Secara lugas sistem informasi didefinisikan sebagai kumpulan orang, prosedur, *hardware*, *software* yang saling berinteraksi untuk memberikan suatu pelayanan informasi bagi user.

Sistem informasi memiliki tiga fungsi dasar :

1. Menerima data (*input*)
2. Merubah data menjadi informasi (proses)
3. Untuk memproduksi dan mengkomunikasikan informasi ke dalam *timely fashion* bagi *user* untuk membuat keputusan (*output*). Sebagai contoh, banyak bank dan institusi keuangan yang menggunakan sistem informasi untuk membantu menentukan apakah nasabah disetujui untuk melakukan pinjaman.

II.2 Sistem Informasi Pelacakan Kiriman Barang

Sistem informasi yang dibuat untuk perusahaan jasa pengiriman barang dalam melacak keberadaan barang kiriman *client* menggunakan fasilitas *delivery tracking* melalui web dan SMS (*Short Message Services*) dengan teknologi GSM (*Global Mobile Communication*). Dengan adanya fasilitas ini membuat *client* mudah melacak barang kirimannya dan memberi kepuasan tersendiri bagi *client*.



Gambar 2.1 Delivery Tracking Services

Kinerja dari sistem informasi ini adalah sebagai berikut :

1. *Client* datang ke jasa pengiriman barang dengan membawa barang kirimannya.
2. Karyawan mencatat data pengiriman barang melalui web. Data pengiriman barang ini disimpan langsung ke dua database yaitu database pusat dan database di kantor cabang sendiri.

3. Setiap barang yang transit di kantor cabang tertentu baik barang yang telah sampai di kota tujuan atau yang sekedar transit saja harus dicatat oleh karyawan ke dalam database pusat.
4. Jika *client* meminta informasi tentang keberadaan barang kirimannya maka *client* mengirima SMS dengan format yang telah ditentukan. Alternatif lain untuk *client* mengetahui informasi tentang keberadaan barang kirimannya adalah melalui web.
5. Setelah sistem mendapat SMS dari *client* maka sistem akan mengecek format SMS *client* terlebih dahulu. Jika data yang diminta *client* (no_nota dan no_aktivasi) ada di dalam database pusat maka sistem akan menginformasikan keberadaan barang kiriman *client*.

Untuk tingkat keamanan sistem informasi pelacakan kiriman barang ini dapat dilakukan dengan pembagian tingkatan hak dan wewenang pengguna dalam mengakses sumber daya dalam beberapa tingkat disesuaikan dengan kebutuhan pengguna yang bersangkutan. Hal ini dapat dilakukan pada akses komputer dan akses ke aplikasi SimPelKiBarSMS. Pada akses komputer, administrator dipegang oleh satu atau beberapa orang yang berwenang penuh terhadap komputer tersebut. Pengguna dapat mengakses komputer menggunakan username dan password yang terpisah dengan pembatasan hak dan wewenang tergantung kebutuhan.

Kelebihan dari sistem ini adalah *client* dapat melacak barang kirimannya dengan 2 alternatif yaitu melalui web dan SMS, interface yang *user-friendly*. Sedangkan untuk kelemahan dari sistem ini terletak pada

jaringan komputernya karena sistem ini dibangun dengan menggunakan alpikasi web maka rawan terhadap akses illegal, serangan virus, hacker, dan beberapa potensi masalah lainnya.

II.3 *Web Based Information System*

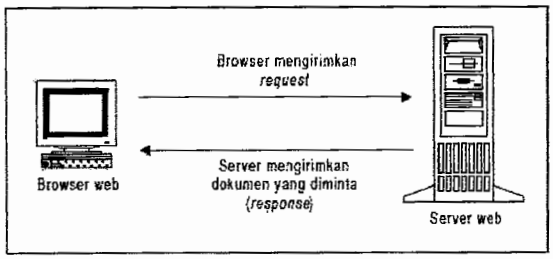
Saat ini komputer dan piranti pendukungnya telah masuk dalam setiap aspek kehidupan dan pekerjaan. Komputer yang ada sekarang memiliki kemampuan yang lebih dari sekedar perhitungan matematika biasa.

Interconnected Network yang biasanya sering disebut dengan *Internet* adalah sebuah sistem komunikasi global yang menghubungkan komputer-komputer dan jaringan-jaringan komputer di seluruh dunia. Komputer dan jaringan dengan berbagai *platform* yang mempunyai perbedaan dan ciri khas masing-masing bertukar informasi dengan sebuah protokol *standard* yang dikenal dengan nama *TCP/IP*.

Web adalah fasilitas *hypertext* untuk menampilkan data berupa teks, gambar, bunyi, animasi, dan data multimedia lainnya, yang mana data tersebut saling berhubungan satu sama lainnya.

II.3.1 *Web Server*

Web server adalah suatu perangkat lunak yang mengatur halaman *web* dan membuat halaman-halaman *web* tersebut dapat diakses di klien, yaitu melalui jaringan lokal atau melalui jaringan *Internet*. Ada banyak *web server* yang tersedia diantaranya *Apache*, *IIS* (*Internet Information Service*), dan *IPlanet's Enterprise server*.



Gambar 2.2 Konsep dasar browser dan server web

II.3.2 *Web Browser*

Web browser digunakan untuk menjelajah situs *web* lewat layanan *HTTP*. Untuk mengakses layanan *WWW* (*World Wide Web*) dari sebuah komputer digunakan program *web client* yang disebut *web browser* atau *browser* saja. Jenis-jenis *browser* yang biasa digunakan adalah *Internet Explorer*, *Netscape*, *NCSA Mosaic*, *Arena*, dan masih banyak lainnya.

II.3.3 *Web* Statis

Web statis merupakan suatu halaman yang berisi skrip HTML editor dan disimpan sebagai *file .htm* atau *.HTML*. Disebut statis karena halaman tersebut dari waktu ke waktu isinya tidak berubah. Karena halaman *web* statis ini tidak memerlukan pemrosesan di *server*, pembuatannya dapat dilakukan menggunakan editor *HTML* dan hasilnya dapat dilihat pada *web browser*.

II.3.4 *Web* Dinamis

Pembuatan halaman *web* dinamis dapat dilakukan dengan dua cara yaitu secara *client side* atau secara *server side*. Penggunaan *client side* dan *server side* tidak saling bertentangan melainkan saling melengkapi. Seorang *web developer* harus dapat menentukan bagian

mana yang diletakkan secara *client side* dan mana yang diletakkan secara *server side*.

II.3.4.1 *Web* Dinamis *Client Side*

Di dalam model *client side* ini, terdapat modul-modul atau *plug-in* yang ditambahkan ke suatu *browser* untuk menciptakan halaman *web* dinamis. Pada umumnya skrip *HTML* dikirim ke *browser* bersama dengan *file* yang berisi sekumpulan instruksi, dan *file* tersebut mengacu pada isi halaman *HTML* tersebut. Akan tetapi, juga merupakan hal yang umum jika sekumpulan instruksi tersebut berada menjadi satu dengan *file HTML*. Kemudian, *browser* menggunakan sekumpulan instruksi tersebut untuk menghasilkan skrip *HTML* ketika terdapat *request* pada halaman tersebut. Singkatnya, halaman tersebut dihasilkan secara dinamis pada saat terdapat *request*.

Kelemahan pada *client side* :

1. Butuh waktu untuk men-*download file* instruksi terutama jika *file* instruksi tersebut berada secara terpisah dan jumlahnya lebih dari satu.
2. Tidak semua *browser* dapat menterjemahkan dan setiap *browser* mempunyai cara berbeda untuk menterjemahkannya.
3. Merupakan suatu masalah untuk membuat skrip bersifat *client side* yang membutuhkan sumber daya dari *server* misalnya *database* di *server*.
4. Setiap skrip *client side* dapat dilihat oleh semua orang, hal ini bisa jadi tidak diinginkan oleh pembuatnya.

Walaupun model *client side* memiliki beberapa kelemahan seperti yang telah disebutkan di atas, akan tetapi model *client side* tetap dibutuhkan oleh seorang *web developer*. *Client side* sangat berguna untuk mengadakan interaksi dengan *user* dalam frekuensi yang cukup tinggi dan data yang diperlukan relatif sedikit dan telah tersedia sebelumnya.

II.3.4.2 *Web* Dinamis *Server Side*

Pada model *server side*, skrip *HTML* yang di dalamnya juga terdapat sekumpulan instruksi, dikirimkan ke *web server*. Seperti halnya *client side*, sekumpulan instruksi tersebut digunakan untuk menghasilkan skrip *HTML* ketika ada *request* terhadap halaman tersebut. Perbedaannya adalah kumpulan instruksi pada *client side* diproses di *client* atau di komputer *user* yang me-*request*-nya, sedangkan pada model *server side* kumpulan instruksi tersebut diproses di komputer *server*.

II.4 Sistem Informasi Berbasis *Mobile*

Dalam beberapa tahun terakhir, banyak perhatian ditujukan pada pengembangan *mobile computing* di dalam industri TI. Berbagai perangkat *mobile* yang berhasil dikembangkan mencakup *handphone*, PDAs (*Personal Digital Assistants*), *communicators*, dan komputer *portable*. Mobilitas telah menjadi sebuah paradigma komputasional yang baru, dimana proses-proses yang terjadi berpindah bersama dengan *user* atau pengguna.

Tentunya ada beberapa penjelasan mengapa mobilitas dan *mobile computing* berkembang dengan pesat. Beberapa alasan dapat ditemukan dari lingkungan atau masyarakat, organisasi, dan *tools*. Corak dasar masyarakat modern

adalah mobilitas dimana jika dibandingkan 50 tahun yang lalu kebanyakan orang jarang meninggalkan rumahnya. Dalam konteks organisasi, alasan utama untuk melakukan mobilitas adalah karena kebanyakan pekerjaan dalam perusahaan modern beralih ke arah pekerjaan yang dikerjakan secara bersama atau tim. Pada bentuk organisasi sebelumnya umumnya tidak mempromosikan bentuk pekerjaan secara bersama-sama atau tim. Adapun penjelasan lainnya adalah adopsi *mobile phones*, dimana didesain untuk memungkinkan orang-orang dapat berhubungan satu dengan lainnya seperti tanpa ada jarak yang memisahkan. Hal inilah yang kemudian menjadi cara baru yang dapat diterapkan dalam berbagai organisasi. Dalam perkembangannya, sebuah sistem informasi jarak jauh yang memberikan hak akses khusus anggota mulai mencoba menggunakan media perangkat *mobile* sebagai sarana untuk menghubungkan antar pengguna dalam sebuah sistem. Saat ini banyak diantara kita bergantung pada sistem dan aplikasi yang menggunakan media perangkat *mobile* yang harus berjalan dengan baik dan terpercaya, misalnya saja sistem *mobile banking*.

Sistem informasi berbasis *Mobile* adalah suatu sistem informasi yang menggunakan media perangkat *mobile* sebagai perantara. Sistem informasi itu sendiri memiliki pengertian suatu *set* komponen yang bekerja bersama-sama untuk mengatur penambahan, penyimpanan, manipulasi dan distribusi dalam in*form*asi atau sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan atau untuk mengendalikan organisasi. Jadi pada sistem informasi berbasis *mobile* ini, berbagai proses yang terdapat pada

sistem informasi dapat dilakukan dan diakses melalui berbagai perangkat *mobile* yang telah kompatibel.

II.5 Teknologi SMS (*Short Messaging Service*)

SMS (*Short Messaging Service*) merupakan teknologi yang memungkinkan untuk mengirimkan dan menerima pesan dalam bentuk *alphanumeric* ke dan dari *handphone*, *Personal Digitals Assistants* (PDAs) dan PC (*Personal Computers*).

Isu SMS pertama kali muncul di belahan Eropa pada sekitar tahun 1991 bersama sebuah teknologi komunikasi *wireless* yang saat ini cukup banyak penggunaanya, yaitu *Global System for Mobile Communication* (GSM). Dipercaya bahwa pesan singkat pertama yang dikirimkan menggunakan SMS dilakukan pada bulan Desember 1992, dikirimkan dari sebuah *Personal Computer* (PC) ke telepon *mobile* (bergerak) dalam jaringan GSM milik perusahaan *Vodafone* Inggris. Perkembangannya kemudian merambah ke benua Amerika, dipelopori oleh beberapa operator komunikasi bergerak berbasis digital seperti *BellSouth Mobility*, *PrimeCo*, *Nextel*, dan beberapa operator lain. Teknologi digital yang digunakan bervariasi dari yang berbasis GSM, *Time Division Multiple Access* (TDMA), hingga *Code Division Multiple Access* (CDMA).

Untuk dapat menggunakan *Short Message Service*, pengguna perlu melakukan pendaftaran atau langganan ke operator jaringan selular yang menyediakan layanan SMS dan memerlukan perangkat komunikasi *mobile* yang mendukung SMS.

Sebuah *short message* tunggal dapat mencapai panjang 160 karakter dari suatu teks ketika menggunakan alpabet Latin dan mencapi 70 karakter ketika menggunakan alpabet

non-Latin seperti *Arabic* dan *Chinese*. Panjang 160 karakter tersebut dapat meliputi kata atau angka atau kombinasi *aphanumeric*. Selain itu *short message* juga ada yang tidak berbasis teks (misalnya, dalam format *binary*). Jenis *short message* ini digunakan untuk *ringtone* dan layanan logo.

Dalam sistem SMS, mekanisme utama yang dilakukan adalah melakukan pengiriman *short message* dari satu terminal pelanggan ke terminal yang lain. Hal ini dapat dilakukan berkat adanya sebuah entitas dalam sistem SMS yang bernama *Short Message Service Centre* (SMSC), disebut juga *Message Centre* (MC). SMSC merupakan sebuah perangkat yang melakukan tugas *store and forward traffic short message*. Di dalamnya termasuk penentuan atau pencarian rute tujuan akhir dari *short message*.

Layanan SMS merupakan sebuah layanan yang bersifat *non-real time* dimana sebuah pesan singkat dapat di-*submit* ke suatu tujuan, tidak peduli apakah tujuan tersebut aktif atau tidak. Bila dideteksi bahwa tujuan tidak aktif, maka sistem akan menunda pengiriman ke tujuan hingga tujuan aktif kembali. Pada dasarnya sistem SMS akan menjamin *delivery* dari suatu *short message* hingga sampai tujuan. Kegagalan pengiriman yang bersifat sementara seperti tujuan tidak aktif akan selalu teridentifikasi sehingga pengiriman ulang *short message* akan selalu dilakukan kecuali bila diberlakukan aturan bahwa *short message* yang telah melampui batas waktu tertentu harus dihapus dan dinyatakan gagal terkirim.

Karakteristik utama SMS adalah SMS merupakan sebuah sistem pengiriman data dalam paket yang bersifat *out-of-band* dengan *bandwidth* kecil. Dengan karakteristik ini,

pengiriman *short message* dapat dilakukan dengan efisiensi yang tinggi. Pada awalnya SMS diciptakan untuk menggantikan layanan *paging* dengan menyediakan layanan serupa yang bersifat *two-way-messaging* ditambah dengan *notification service* untuk *voicemail*. Pada perkembangan selanjutya, muncul jenis-jenis layanan lain seperti *email*, *fax*, *interactive banking*, *information service*, *Location Based Services* (LBS) dan integrasi dengan aplikasi berbasis internet. Integrasi dengan aplikasi berbasis internet kemudian mendorong timbulnya layanan seperti *web-based messaging*, *gaming*, dan *chatting*.

II.6 *Mobile* FBUS 1.5

FBUS merupakan suatu metode komunikasi diantara produk *handphone* Nokia dan *port* serial pada PC yang digunakan untuk melakukan transmisi data, layanan, dan penyesuaian.

Mobile FBUS 1.5 adalah produk dari *Software Cave* dan merupakan *ActiveX Control* yang bersifat *freeware* dimana dapat digunakan untuk membuat perangkat lunak atau *software* untuk, *handphone* seperti mengirim SMS, memanipulasi logo operator, mengakses *phonebook*, dan lain-lain. *Mobile* FBUS merupakan *tool* yang ideal untuk menghubungkan perangkat komunikasi seperti *handphone* dengan aplikasi *Visual Basic*. *Mobile* FBUS dapat bekerja pada produk *handphone* GSM Nokia, dimana dalam hal ini *Software Cave* telah melakukan uji coba pada produk *handphone* Nokia seri 51xx/61xx dan 3210. Berikut ini adalah beberapa acuan dalam pemrograman *Mobile* FBUS 1.5 yaitu :

1. *Properties*

Kebanyakan *properties* dari *ActiveX Control Mobile* FBUS hanya dapat digunakan ketika kontrol tersebut terhubung dengan *handphone*.

a. ACDCPower property

Boolean Read only

Properti ini mengembalikan nilai *True*/Benar ketika *handphone* dihubungkan dengan sumber tenaga dari luar, misalnya baterai pembangkit.

b. *BatteryLogo property*

Single Read only

Properti ini mengembalikan tingkat/level baterai dari *handphone* yang terhubung. Unit yang digunakan tergantung pada model *handphone*.

c. *CellId property*

String-Read only

Properti ini mengembalikan *Cell Id* dari node jaringan pada handphone yang sedang terhubung. *Cell Id* bersifat unik untuk setiap transmitter atau pemancar jaringan GSM, walaupun *Location Area Code* (LAC) berada di antara sejumlah pemancar pada area yang sama.

d. *Connected property*

Boolean-Read only

Properti ini akan mengembalikan nilai jika kendali terhubung ke *handphone*.

e. *DateTime property*

Date-Read/Write

Properti ini mengembalikan atau men-*set* tanggal dan waktu pada jam yang ada dalam *handphone* yang terhubung.

Catatan :

1. Anda tidak dapat men-*set* detik karena detik yang menjadi bagian dari format VB *Date* akan diabaikan ketika Anda men-*set* waktu alarm.
2. Pada *handphone* Nokia seri 51xx Anda tidak dapat men-*set* tanggal melalui antarmuka yang ada pada *handphone*, sehingga kemungkinan dapat berisi nilai *random* ketika Anda melakukan *query*. Meskipun demikian, Anda dapat men-*set* tanggal melalui *Mobile* FBUS dan setelah itu *handphone* Anda akan tetap ter-*set* pada tanggal yang benar.

f. *KeyboardLocked property*

Boolean-Read only

Properti ini mengembalikan nilai *True* jika papan tombol pada *handphone* yang terhubung berada dalam posisi tidak dapat digunakan.

g. *ProviderCode property*

String-Read only

Properti ini mengembalikan kode provider untuk provider jaringan yang ada dalam format "ccc nn", dimana ccc merupakan kode negara sebanyak 3 digit dan nn adalah kode jaringan. Sebagai contoh, provider Belanda Ben memiliki kode provider "204 16".

h. *ProviderCountry property*

String-Read only

Properti ini mengembalikan nama negara provider selular dari handphone yang terhubung.

i. *ProviderName property*

String-Read only

Properti ini mengembalikan nama provider selular dari *handphone* yang terhubung.

j. *ProviderSMSC property*

String-Read only

Properti ini mengembalikan nomor *handphone* dari *Short Message Service Centre* (SMSC) provider selular pada handphone yang terhubung.

k. *RfLevel property*

Single-Read only

Properti ini mengembalikan kekuatan sinyal GSM yang terdeteksi oleh *handphone*. Unit yang digunakan tergantung pada model *handphone*.

l. *SMSClass property*

Integer-Read/Write

Properti ini mengembalikan atau men-set kelas pesan SMS yang akan digunakan dalam metode *SendSMSMessage*. Kelas pesan mendefinisikan bagaimana pesan ditampilkan oleh *handphone* penerima. Beberapa kelas yang biasa digunakan antara lain :

fbSmsClassFlash	pesan tertampil sebagai pesan sekejap (terlihat dengan cepat pada layar *handphone* tanpa konfirmasi pengguna).
fbSmsClassNormal	pesan diterima sebagai pesan SMS biasa.
fbSmsClass2	pengaruhnya sama dengan *fbSmsClass* Normal pada *handphone* Nokia.
fbSmsClass3	pengaruhnya sama dengan *fbSmsClassNormal* pada *handphone* Nokia.

2. *Method*

a. *Connect method*

Sub Connect [sComport As String="COM1"]

Metode ini digunakan untuk menghubungkan kendali pada *handphone*. Hal ini merupakan hal yang mendasar yang harus dilakukan, seolah-olah tidak ada metode lain yang akan bekerja apabila metode ini tidak dipanggil.

Parameter :

sComport-Optional-Port yang digunakan untuk menghubungkan. Jika diabaikan, maka yang digunakan adalah COM1. Tuliskan nama *comport* sebagai berikut "COM<nomor>" (misalnya : COM1, COM2, COM3 dan lain-lain).

Nilai :

Tidak ada

b. *Disconnect method*

Sub Disconnect

Metode ini akan mengakhiri hubungan dengan *handphone* dan dengan metode ini akan menonaktifkan seluruh metode yang ada dalam kontrol.

Parameter :

Tidak ada

Nilai :

Tidak ada

3. *Collections*

a. *Phonebook*

Phonebook(iLocation As Integer,[iMemoryType As fbMemory]) As FBPhonebookEntry

Collection ini dapat digunakan untuk mengatur *phonebook* yang tersimpan dalam SIM atau *memory handphone*.

Parameter :

iLocation - Mandatory - Lokasi *memory* untuk dibaca atau ditulisi.

iMemoryType - Optional - Jenis *memory* yang akan diakses. Jika parameter ini diabaikan, maka akan digunakan *memory* SIM.

Properties :

Group - Read/Write Integer - Grup ID untuk *entri phonebook*. 0 jika *entri* bukan bagian dari grup.

Last Error - Read only Integer - Kesalahan terakhir dari penggunaan *phonebook*. 0 jika penggunaan sukses.

Name - *Read/Write String* - Nama untuk *entri phonebook*.

Number - Read/Write String - Nomor telepon untuk *entri phonebook*.

Method :

WriteData-Gunakan perintah ini untuk menuliskan Nama,Nomor dan Grup ID per satu kali penggunaan. Perintah ini lebih cepat daripada memasukkan tiga *properties* tadi satu per satu.

4. SMS Objek

Obyek ini dapat digunakan untuk membaca dan menulis pesan SMS. *SMS Object* memiliki beberapa *properties*, metode dan sub-objek sebagai berikut.

Properti	Deskripsi
Last Error	Menghasilkan kode kesalahan terakhir
TotalMessage	Menghasilkan jumlah total SMS yang disimpan
UnreadMessage	Menghasilkan jumlah pesan yang belum dibaca
Method	**Deskripsi**
Refresh	Mengambil pesan yang tersimpan dalam handphone
SendMessage	Mengirim pesanSMS
Object	**Deskripsi**
DeliveryNotification	*Message box* berisi notifikasi pengiriman
Inbox	*Message box* berisi SMS yang diterima
Outbox	*Message box* beriasi SMS yang terkirim
FBSmsMessage	Sub-*Object* yang mengacu pada satu pesan di dalam kotak objek

4.1 Properties

a. SMS.LastError property

Integer-Read only

Properti ini akan mengembalikan *error* internal terakhir yang terjadi saat memanggil properti atau metode SMS. *Last Error* akan bernilai 0 ketika operasi telah terakhir selesai.

b. SMS.TotalMessages property

Integer-Read only

Properti ini akan mengembalikan total SMS yang ada di dalam SIM. Properti ini dapat berjalan meskipun SMS.*Refresh* tidak dijalankan.

c. SMS.UnreadMessages property

Integer-Read only

Properti ini akan mengembalikan jumlah SMS yang tidak terbaca dalam *memory*.

4.2 *Methods*

a. SMS.*Refresh method*

Sub Refresh()

Metode ini akan me-*refresh* daftar pesan yang tersimpan dalam SMS.*Inbox*, SMS.*OutBox* dan SMS.*Delivery-Notification*.

Parameter :

Tidak ada

Nilai :

Tidak ada

b. SMS.*SendMessage method*

Function SendMessage(sDest As String, sMessage As String,[iSmsClass As fbSmsClass=fbSmsClassNormal]) As Boolean

Metode ini akan mengirimkan SMS ke *sDest*, dengan isi *sMessage*. Parameter *iSMSClass* dapat digunakan untuk mengontrol tipe SMS yang dikirimkan.Tipenya adalah :

1. *fbSmsClassFlash* : pesan tertampil sebagai pesan sekejap (terlihat dengan cepat pada layar *handphone* tanpa konfirmasi pengguna).
2. *fbSmsClassNormal* : pesan diterima sebagai pesan SMS biasa.
3. *fbSmsClass2* : pengaruhnya sama dengan *fbSmsClass* Normal pada *handphone* Nokia.
4. *fbSmsClass3* : pengaruhnya sama dengan *fbSmsClass Normal* pada *handphone* Nokia.

Nilai :

Bernilai *True* jika fungsi sukses.

4.3 Sub-Objek

a. SMS.DeliveryNotifications/inbox/Outbox objects

Objek ini merupakan enkapsulasi dari 3 kotak SMS utama, dengan masing-masing kotak memiliki properti *Count*, yang merupakan jumlah pesan di dalam *inbox*. Pesan-pesan memiliki indeks di antara 1 dan *Count*.

b. FBSmsMessage object

Setiap kotak berisi sebuah *Collection* dari objek pesan yang bertipe *FBSmsMessage*, setiap pesan memiliki *properties* dan *method* seperti berikut ini.

Properti	**Deskripsi**
DateTime	Waktu dari pesan (*read-only Date*)
Destination	Tujuan Pesan (*read-only String*)
LastError	*Error* yang terakhir terjadi, berinilai 0 jika sukses (*read-only Integer*)
Sender	Pengirim Pesan (*read-only String*)
SentRead	Akan bernilai *True* jika pesan telah terkirim atau terbaca (*read-only Boolean*)
Text	Isi dari pesan (*read-only String*)
Method	**Deskripsi**
Delete	Menghapus pesan dari SIM atau memori telepon