BAB II

LANDASAN TEORI

2.1 Jaringan Telekomunikasi

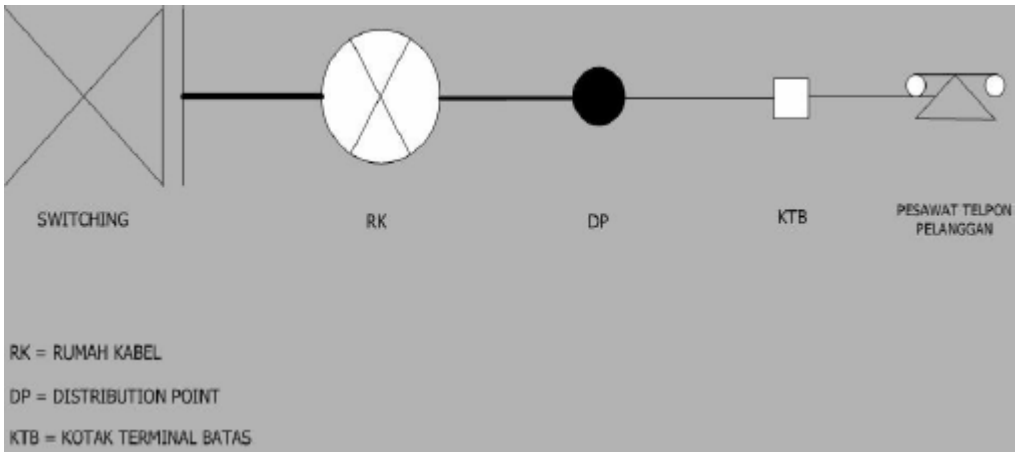
Jaringan telekomunikasi merupakan sarana bagi informasi atau komunikasi mengalir dari pengirim ke penerima. Jaringan ini terdiri dari dua bentuk utama yaitu jaringan fisik atau bisa disebut jaringan kabel dan jaringan radio atau jaringan nirkabel. Pada paparan berikut akan dijelaskan dua jenis jaringan yang dipergunakan oleh TELKOM saat ini yaitu jaringan kabel dan jaringan CDMA (*Code Division Multiple Access*).

2.1.1 Jaringan Kabel

Sebagian besar pelanggan tetap TELKOM adalah pelanggan berbasis kabel tembaga. Jauh sebelum dikenalnya teknologi GSM ataupun CDMA yang penggunaannya sekarang sedang meningkat pesat, kabel tembaga milik PT Telkom sudah masuk ke berbagai pelosok tanah air. Dari lokasi rumah pelanggan yang berada strategis di pusat kota apalagi di pinggir jalan besar, sampai ke berbagai pelosok kampung dan gang-gang sempit pemukiman padat. Jaringan kabel TELKOM merupakan jaringan kabel tertua yang telah melayani berbagai lapisan masyarakat di tanah air ini dalam jasa telekomunikasi.

Berbagai perangkat canggih silih berganti mengantikan teknologi yang usang, namun tetap kabel kabel tembaga belum tergantikan. Konfigurasi mulai

dari Perangkat *switching* sampai pesawat telpon pelanggan pun masih sama. Konfigurasinya dapat dilihat pada gambar 2.1 di bawah ini :



Gambar 2.1 Konfigurasi Jaringan Kabel

Penjelasan singkat tentang gambar 2.1 di atas adalah :

1) *Switching* atau biasa disebut juga dengan sentral telepon, adalah sekumpulan perangkat yang memiliki fungsi-fungsi dasar dalam proses penyambungan dan pemutusan panggilan atau komunikasi.

2) Rumah Kabel (RK), adalah terminasi kabel-kabel primer.

3) *Distribution Point (DP)* adalah terminasi dari kabel kabel sekunder.

4) Kotak Terminal Batas adalah terminasi dari kabel Drop Wire

Jenis – jenis gangguan yang umumnya terjadi pada jaringan kabel adalah sebagai berikut :

1) Mati total yaitu gangguan dimana pengguna jaringan tidak mendapatkan nada sambung.

2) Kemerosok atau suara yang tidak jernih.

3) Induksi yaitu merupakan gangguan yang terjadi karena adanya kebocoran suara dari saluran yang berbeda dari saluran pengguna.

4) Gangguan yang bersifat massal (GAMAS) disebabkan oleh kerusakan pada salah satu perangkat seperti terjadi kebakaran pada kabel sekunder dan sebagainya.

2.1.2 Jaringan CDMA

CDMA merupakan singkatan dari *Code Division Multiple Acces* yang merupakan teknologi jaringan telekomunikasi yang mempergunakan gelombang radio sebagai sarana pengiriman informasi. Teknologi ini mempergunakan metode *multiple access* yang membagi gelombang radio dengan *pseudo random code* yang menandai setiap pengguna.

Frekuensi gelombang radio yang dipergunakan oleh operator CDMA di Indonesia berkisar diantara 800 Mhz dan 1900 Mhz tergantung dari izin frekuensi yang dimiliki oleh operator tersebut. PT. TELKOM sebagai salah satu operator CDMA mempergunakan frekuensi 1900 Mhz di area pelayanan kota Bandung.

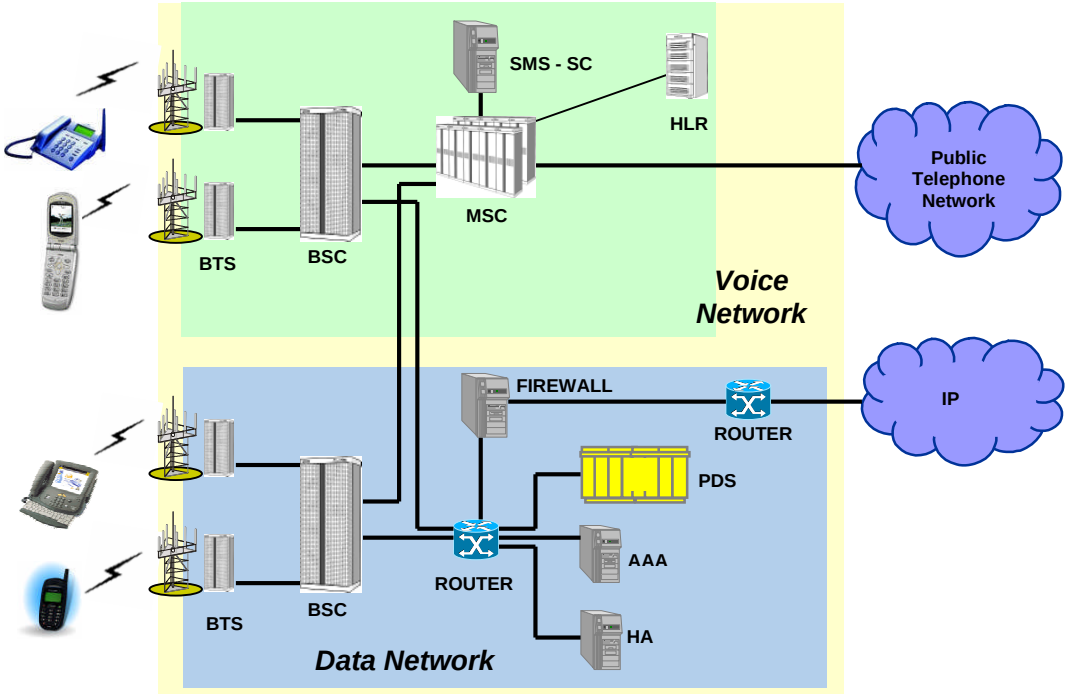
Konfigurasi dari jaringan CDMA terdiri dari beberapa tahap yaitu :

1) Network and Switching Subsystem (NSS) yang tidak berbeda jauh dari sentral telepon jaringan kabel sebagai pusat penyambungan dan pemutusan panggilan komunikasi. Ini merupakan jaringan pusat dari sistem komunikasi CDMA Subsistem ini meliputi MSC, HLR , SMS-C, ROUTER, dan FireWall.

2) Base Station Subsytem (BSS) yang bertanggung jawab dalam mengelola trafik dan pensinyalan antara sebuah pesawat handphone dan NSS. Subsistem ini meliputi BSC dan BTS.

3) Perangkat pesawat handphone sebagai alat komunikasi bagi pengguna CDMA Jenis pesawat yang tersedia adalah pesawat jenis telepon genggam (*handheld*), telepon rumah (*fix wireless*) dan modem data.

Gambar 2.2 mengambarkan konfigurasi dari jaringan telekomunikasi CDMA.



Gambar 2.2 Konfigurasi Jaringan CDMA

2.2 *Short Message Service* (SMS)

SMS adalah kependekan dari *Short Message Service* yang merupakan fitur dari telepon selular yang berbasis teknologi CDMA (*Code Division Multiple Access*). Sebuah fitur untuk mengirimkan pesan singkat berupa teks. Teknologi CDMA ini memiliki standar yang telah disepakati oleh *Standarization Organization* antara lain ETSI (*European Telecomunication Standards Institute*), IrDA (*Infrared Data Association*) serta ISO (*International Standards Organization*).

Fitur SMS saat ini cukup populer dan banyak digunakan oleh pemakai jasa telekomunikasi karena selain penggunaannya mudah dan biayanya murah, SMS juga bisa dimanfaatkan untuk sistem *Alert*, seperti aplikasi yang akan dikembangkan.

2.2.1 Format Teks

Pada mode pengiriman sms melalui modem CDMA , format PDU tidak dipergunakan. Mode pengiriman yang didukung oleh modem CDMA adalah Octet, IS91EP, ASCII, IA5, dan USC-2 Unicode. Dalam pengembangan aplikasi ini akan dipergunakan mode pengiriman dengan mode ASCII dengan panjang maksimum sebanyak 160 karakter untuk satu kali pengiriman pesan.

2.2.2 AT *Command*

AT *Command* digunakan untuk melakukan komunikasi dengan *modem* dan juga untuk mengontrol *modem*. Telepon seluler pada dasarnya adalah *modem*,

sehingga AT *Command* pun berlaku pada telepon seluler. Begitu juga dengan *modem CDMA.* AT Command untuk telepon seluler ditetapkan dengan adanya standar IS 707.3.

Perintah-perintah AT *Command* sebenarnya hampir sama dengan perintah > (*prompt*) pada DOS. Perintah-perintah yang dimasukkan ke *port* dimulai dengan kata AT yang merupakan kependekan dari kata *Attention,* lalu diikuti oleh karakter lainnya, yang memiliki fungsi-fungsi unik.

Berikut akan dituliskan beberapa perintah-perintah AT *Command* yang merupakan perintah yang diberikan kepada *Terminal Equipment*, dalam hal ini adalah komputer , dengan *CDMA Modem* yang berfungsi sebagai *Communication Terminal*.

1) +CMGR (*Read Message*)

Perintah ini untuk membaca pesan yang tersimpan. Pesan dibaca dari memori yang ditentukan oleh perintah +CPMS. Pesan yang belum terbaca akan memiliki status ”*REC UNREAD*”. Setelah perintah +CMGR status tersebut akan dirubah menjadi ”*REC READ*”. Perintah ini membaca sms yang tersimpan berdasarkan indeks dari sms tersebut.

Sintaksnya adalah AT+CMGR=<index>

Akan dibalas oleh modem dengan sintaks :

+CMGR:<stat>[,<oa>],<scts>,<lang>,<encode>,<priority>[,<cbn>],<length><CR><LF> <data>

Pengertian dari nilai parameter diatas adalah :

<stat> : status dari pesan didalam memori modem. Terdiri dari ”*UREAD*” yang berarti pesan belum terbaca , ”*READ*” berarti pesan sudah terbaca.

<oa> : nomor pengirim

<scts> : tanggal service center dengan format “yy/MM/dd,hh:mm:ss± zz” atau “Tahun/Bulan/Tanggal,Jam:Menit:Detik±Zona Waktu”

<lang> : bahasa

<encod> : mode enkoding yang dipergunakan

<priority> : prioritas pesan. ”0” untuk normal, ”1” untuk interaktif, ”2” untuk segera, ”3” untuk darurat.

<cbn> : nomor untuk panggilan kembali

<length> : panjang pesan

<CR> : *carriage return*, karakter enter dalam ASCII

<LF> : *line feeds* , karakter pindah baris baru dalam ASCII

<data> : teks pesan

Contoh :

AT+CMGR=1 <enter>

+CMGR: “REC UNREAD”, ”02270817470”, ”07/04/01,00:01:01+07”, 1,2,0,”02270817470”,9<CR><LF>

Test

OK

2) +CMGL (*List Message*)

Perintah ini untuk mengeluarkan daftar dari pesan yang tersimpan dalam memori. Pesan akan muncul dalam urutan berdasarkan indeks penyimpanannya.

Sintaksnya adalah : AT+CMGL=<stat>

Modem akan membalas dengan sintaks :

+CMGL<index>,<stat>,<da/oa>,<lang>,<encod>,<length><CR><LF><data>

Nilai parameter pada perintah ini sama seperti perintah sebelumnya.

Contoh : AT+CMGL=”UREAD”

+CMGL:1,”REC UNREAD”,”02270817470”,1,2,15<CR><LF>

Isi Pesan

OK

3) +CMGS

Perintah ini untuk mengirimkan pesan. Terdiri dari dua baris. Baris pertama merupakan perintah dan parameternya. Baris kedua adalah isi pesannya. Sebuah karakter akhir ditambahkan untuk memulai ataupun membatalkan proses pengiriman pesan oleh modem. Karakter tersebut dikenal sebagai karakter terminal <*term*>.

Sintaksnya adalah : AT+CMGS= <da> [,<length> [,<priority>] [,<privacy>] [,<reply>][,<cbn><cr><message text>[<term>]

Modem akan membalas dengan : +CMGS:<mr>

OK

Nilai parameter dalam perintah ini adalah :

<da> : nomor tujuan

<length> : panjang teks pesan

<piority> : prioritas pesan (sama seperti +CMGR)

<privacy> : nilai prifasi dari pesan. “0” berarti tidak dibatasi. “1” berarti dibatasi. “2” berarti terbatas untuk kalangan tertentu. “3” berarti rahasia.

<reply> : Pilihan status keberhasilan untuk pesan. “0” berarti tidak meminta status. ”1” untuk status kirim. ”2” untuk status sampai ke tujuan. ”3” untuk meminta baik status kirim maupun status sampai ke tujuan.

<cbn> : nomor yang bisa dipanggil kembali.

<cr> : *carriage return*, karakter enter dalam ASCII.

<message text> : isi pesan dalam mode teks.

<term> : *message terminator*, karakter pengakhir pesan. Diabaikan jika nilai <length> disi. Terdiri dari <ctrl-Z> untuk mengirim pesan dan <esc> untuk membatalkan proses pengiriman.

Contoh : AT+CMGS=”02270817470”<cr>

Ujicoba kirim pesan dengan modem CDMA<ctrl-Z>

+CMGS:<mr>

OK

4) +CMGD

Perintah ini dipergunakan untuk menghapus satu atau banyak pesan dari memori modem CDMA.

Sintaksnya : AT+CMGD=<index>[,<DelFlag>]

Nilai parameter yang dipergunakan adalah :

<index> : indeks dari pesan yang akan dihapus

<DelFlag> : tanda untuk mode penghapusan. “0” untuk hapus semua pesan pada lokasi tertentu. “1” untuk hapus semua pesan yang berstatus sudah dibaca. “2”

untuk semua pesan yang berstatus sudah dibaca dan dikirim. “3” untuk pesan dengan status terbaca, terkirim dan tidak terkirim. “4” untuk semua pesan.

Contoh : AT+CMGD=3

OK

2.3 *Visual Basic*

Visual Basic adalah salah suatu bahasa pemrograman komputer. *Visual Basic*, yang dikembangkan oleh *Microsoft* sejak tahun 1991, merupakan pengembangan dari pendahulunya yaitu bahasa pemrograman BASIC (*Beginner’s Allpurpose Symbolic Instruction Code*) yang dikembangkan pada era 1950-an. *Visual Basic* merupakan salah satu *Development Tool* yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi *Windows*. *Visual Basic* merupakan salah satu bahasa pemrograman komputer yang mendukung *object* (*Object Oriented Programming = OOP*).

2.3.1 IDE *Visual Basic*

IDE (*Intergrated Development Environment*) *Visual Basic* adalah Lingkungan Pengembangan Terpadu bagi programmer dalam mengembangkan aplikasinya. Dengan menggunakan IDE *programmer* dapat membuat *user interface*, melakukan *coding*, melakukan *testing* dan *debugging* serta mengkompilasi program menjadi *executable*.

IDE *Window* adalah jendela yang menunjukkan bagian-bagian yang dapat tampil pada IDE *Visual Basic*. Jendela-jendela ini dapat ditampilkan dengan menu *View*. Adapun jendela-jendela yang perlu diperhatikan adalah :

1) *Menu Bar*, digunakan untuk memilih tugas-tugas tertentu seperti menyimpan *project*, membuka *project* dan lain-lain.

2) *Main Toolbar*, digunakan untuk melakukan tugas-tugas tertentu dengan cepat.

3) Jendela *Project*, jendela ini berisi gambaran dari semua modul yang terdapat dalam aplikasi. *Icon Toggle Folder* digunakan untuk menampilkan modul modul dalam jendela tersebut secara grup atau mengurut berdasarkan nama. Ctrl+R digunakan untuk menampilkan jendela *project* ataupun menggunakan *icon project explorer*.

4) Jendela *Form Designer*, jendela ini merupakan tempat untuk merancang *user interface* dari aplikasi. Jadi jendela ini menyerupai kanvas bagi seorang pelukis.

5) Jendela *Toolbox*, jendela ini berisi komponen-komponen yang dapat digunakan untuk mengembangkan *user interface*. Secara garis besar fungsi dari *Toolbox control* yang digunakan adalah sebagai berikut :

(1) *Pointer*, bukan merupakan suatu kontrol. Gunakan icon ini ketika akan memilih kontrol yang sudah berada pada jendela *Form Designer*.

(2) *Label*, adalah kontrol yang digunakan untuk menampilkan teks yang tidak dapat diperbaiki oleh pemakai.

(3) *TextBox*, adalah kontrol yang mengandung *string* yang dapat diperbaiki oleh pemakai, dapat berupa satu baris tunggal atau banyak baris.

(4) *Frame*, adalah kontrol yang digunakan sebagai *container* bagi kontrol lainnya.

(5) *CommandButton*, merupakan kontrol yang hampir ditemukan pada setiap *form* dan digunakan untuk membangkitkan *event* proses tertentu ketika pemakai melakukan *click* padanya.

(6) *CheckBox*, digunakan untuk pilihan yang isinya bernilai *Yes/No*, *True/False*.

(7) *OptionButton*, sering digunakan lebih dari satu sebagai pilihan terhadap *option* yang hanya dapat dipilih satu.

(8) *ListBox*, mengandung jumlah item pilihan dan user dapat memilih lebih dari satu (bergantung pada properti *Multiselect*).

(9) *ComboBox*, merupakan kombinasi dari *TextBox* dan *ListBox* dimana pemasukan data dapat dilakukan dengan pengetikan maupun dengan pemilihan.

(10) *Shape* dan *Line*, digunakan untuk menampilkan bentuk seperti garis, persegi, bulatan dan oval.

(11) *Object Linked Embedded* (OLE), dapat digunakan sebagai tempat untuk menempelkan program eksternal seperti *Microsoft Excel* , *Word* dan lainnya, atau bisa juga berupa gambar.

(12) *MsComm*, dapat digunakan untuk mengontrol transmisi dan penerimaan data antara aplikasi dengan serial port.

6) Jendela *Code*, merupakan tempat untuk menulis *coding*. Jendela ini dapat ditampilkan dengan menggunakan kombinasi *Shift*-F7.

7) Jendela *Properties*, merupakan daftar properti-properti obyek yang sedang terpilih, contohnya untuk mengubah warna tulisan (*foreground*) dan warna latar belakang (*background*). Jendela properti ini dapat ditampilkan dengan menggunakan F4.

8) Jendela *Color Palette*, adalah fasilitas cepat untuk mengubah warna suatu obyek.

9) Jendela *Form Layout*, akan menunjukkan bagaimana form bersangkutan ditampilkan ketika *Runtime*.

2.3.2 *Object, Property, Method* dan *Event*

Dalam pemrograman berbasis objek seperti *Visual Basic*, terdapat istilah *Object, Property, Method* dan *Event*. Penjelasan dari istilah tersebut adalah sebagai berikut :

1) *Object*, adalah komponen di dalam sebuah program.

2) *Property*, adalah karakteristik yang dimiliki oleh sebuah *object*.

3) *Method*, adalah aksi yang dapat dilakukan oleh *object*.

4) *Event*, adalah kejadian yang dapat dialami oleh *object*.

Implementasinya dalam sebuah aplikasi misalnya sebuah komponen atau *object Form*, maka *Form* tersebut memiliki *Property, Method* dan *Event*. *Visual* Basic bersifat *even driven programming*, artinya kita dapat menyisipkan kode program pada *event* yang dimiliki suatu *object*.

2.3.3 Tipe Data

Data adalah informasi yang akan disimpan untuk diolah oleh komputer . *Visual Basic* mengenal beberapa tipe data, antara lain seperti disebutkan pada Tabel 2.1 berikut:

Tabel 2.1 Tipe Data Visual Basic

Tipe Data	**Keterangan**
Integer	Tipe data numerik berupa bilangan bulat , dengan kisaran -32768 s/d 32767
Byte	Tipe data berupa nilai bulat positif (tanpa pecahan) dengan kisaran 0 s/d 255
Decimal	Digunakan untuk menyimpan nilai desimal (pecahan) dengan ketepatan hingga 28 angka desimal
Boolean	Hanya memiliki dua buah nilai yaitu *TRUE* atau *FALSE*
String	Tipe data yang memiliki nilai alanumerik yang bisa berupa huruf, angka atau karakter khusus. Angka pada String tidak bisa melakukan operasi matematik
Single	Tipe data numerik dengan kisaran nilai –340282E+38 s/d 3402823E+38. Sering juga disebut *Single Precision*

Tipe Data	Keterangan
Date	Digunakan untuk menyimpan nilai tanggal dan jam dengan kisaran nilai 1 Januari 100 hingga 31 Desember 9999
Currency	Digunakan untuk menyimpan nilai uang dengan kisaran nilai 922337203685477,5808 s/d 922337203685477,5807
Long	Mirip dengan *Integer* dengan kisaran -2147483648 s/d 2147483647
Object	Tipe data yang menyimpan objek seperti *form*, kontrol dan sebagainya
Variant	Berisi segala macam tipe data, biasanya digunakan jika pemakai tidak tahu tipe data apa yang harus digunakan

2.3.4 Variabel

Data yang disimpan di dalam *Visual Basic* membutuhkan sebuah tempat. Tempat inilah yang disebut dengan variabel. Setiap variabel untuk menyimpan data dengan tipe data tertentu membutuhkan alokasi jumlah *memory (byte)* yang berbeda. Ukuran byte masing-masing tipe data seperti yang tertera pada table 2.1 diatas.

Visual Basic memungkinkan kita untuk membuat variabel tanpa deklarasi, walaupun kurang baik untuk program yang terstruktur. Program yang terstruktur

dapat menghindari kesalahan pengolahan yang diakibatkan oleh kesalahan dalam pengetikan nama variabel.

Deklarasi variabel pada *Visual Basic* dapat dilakukan dengan *keyword* pada tabel 2.2 berikut sebagai berikut :

Tabel 2.2 *Keyword* pada Visual Basic

Keyword	**Digunakan pada**
Public	Berlaku pada *level modul*
Private	Berlaku pada *level modul*
Dim	Berlaku pada *level modul* dan *level procedure*
Static	Berlaku pada *level procedure*

Variabel dibuat melalui penulisan deklarasi variabel di dalam kode program, dengan format :

Dim <nama_variabel> As <tipe_data>

Contoh : Dim seq_number As Integer

Sebuah variabel mempunyai ruang-lingkup (*scope*) dan waktu-hidup (*lifetime*). Variabel *global* adalah variabel yang dapat dikenali oleh seluruh bagian program. Nilai data yang tersimpan didalamnya akan hidup terus selama program berjalan. Variabel lokal adalah variabel yang hanya dikenali oleh satu bagian program saja. Nilai data yang tersimpan didalamnya hanya hidup selama bagian program tersebut dijalankan.

Variabel yang nilai datanya bersifat tetap dan tidak bisa diubah disebut konstanta. Penulisan deklarasi konstanta di dalam kode program adalah :

Const <nama_kontanta> As <type_data> = <nilai_data>

Contoh : Const phi As Integer = 3.14

2.3.5 *Operator*

Operator bisa diartikan sebagai simbol yang digunakan untuk melakukan suatu operasi terhadap nilai data. Simbol *operator* bisa berupa karakter ataupun kata khusus. *Visual Basic* menyediakan *operator* aritmatika, perbandingan dan logikal.

1) Operator Aritmetika

Operator aritmetika adalah operator yang digunakan untuk melakukan perhitungan matematis aritmetika, seperti penjumlahan, pengurangan dan sebagainya. Berikut daftar operator pada VB berdasar tingkat urutan pengerjaannya jika ditemui dalam suatu persamaan yang kompleks. Tabel 2.3 menggambarkan berbagai operator tersebut.

Tabel 2.3 Operator Aritmetika

OPERATOR	**ARTI**
^	Pangkat
*	Perkalian
/	Pembagian
+	Penjumlahan

-	Pengurangan
& (atau +)	Penggabungan *String*

2) Operator Perbandingan

Operator perbandingan adalah operator yang digunakan untuk membandingkan satu nilai dengan nilai yang lain untuk mendapatkan suatu nilai *TRUE* atau *FALSE*. Operator ini biasanya dipakai untuk mengambil suatu keputusan pada suatu kondisi. Tabel 2.4 berikut mencantumkan operator tersebut.

Tabel 2.4 Operator Perbandingan

OPERATOR	**ARTI**
=	Sama dengan
>	Lebih besar dari
<	Kurang dari
>=	Lebih besar atau sama dengan
<=	Kurang dari atau sama dengan
<>	Tidak sama dengan

3) Operator Logikal

Operator logikal sebenarnya hampir sama dengan operator perbandingan, yang fungsinya adalah membandingkan dua buah nilai untuk memberikan hasil *TRUE* atau *FALSE*. Perbedaannya adalah operator Logikal berfungsi

membandingkan perbandingan. Tabel 2.5 mencantumkan berbagai operator tersebut.

Tabel 2.5 Operator Logikal

OPERATOR	**ARTI**
AND	Akan menghasilkan nilai *TRUE* jika kedua nilai yang dibandingkan bernilai *TRUE*
OR	Akan menghasilkan nilai *TRUE* jika salah satu nilai yang dibandingkan bernilai *TRUE*
NOT	Akan menghasilkan nilai *TRUE* jika yang dibandingkan bernilai *FALSE*

2.3.6 Fungsi Bantu pada *Visual Basic*

Banyak fungsi Bantu yang dimiliki oleh *Visual Basic*, namun yang akan ditulis disini adalah beberapa fungsi yang dipakai pada pembuatan aplikasi ini.

1) CStr(ekspresi), untuk konversi suatu ekspresi ke *string*.

Contoh:

String = CStr(261811)

Maka variabel String akan berisi nilai “261811”

2) *Asc(string)*, untuk mengembalikan kode karakter dari huruf pertama di suatu string.

Contoh:

Nomer = Asc(“A’)

Akan mengembalikan 65

3) Chr(*charcode*), untuk mengembalikan karakter dari suatu kode karakter.

Contoh :

MyChar = Chr(65)

Akan mengembalikan A

4) Val(*string*), untuk mengembalikan angka yang terkandung dalam suatu string.

Contoh :

MyVal = Val(“2485”)

Maka variabel *MyVal* akan berisi nilai 2485

2.3.7 Struktur Kontrol

Struktur kontrol di dalam bahasa pemrograman adalah perintah dalam bentuk tertentu yang digunakan untuk mengontrol jalannya program. *Visual Basic* mengenal beberapa jenis struktur kontrol, yaitu :

1) IF <kondisi> THEN <kode program>

Bila <kondisi> bernilai True maka <kode program> akan dikerjakan.

2) IF <kondisi> THEN

<blok kode program 1>

ELSE

<blok kode program 2>

END IF

Bila <kondisi> bernilai True maka <blok kode program 1> akan dikerjakan, tetapi bila <kondisi> bernilai False maka <blok kode program 2> yang akan dikerjakan.

3) SELECT CASE <pilihan>

CASE <pilihan 1>

<blok kode program 1>

CASE <pilihan 2>

<blok kode program 2>

CASE <pilihan n>

<blok kode program n>

[CASE ELSE

<blok kode program x>]

END SELECT

Bila <pilihan> sesuai dengan <pilihan 1> maka <blok kode program 1> akan dikerjakan, dst. Tetapi bila bila <pilihan> tidak ada yang sesuai dengan <pilihan 1> s/d <pilihan n> maka <blok kode program x> yang akan dikerjakan.

4) FOR <pencacah> = <awal> TO <akhir> [STEP <langkah>]

<blok kode program>

NEXT <pencacah>

Bagian <blok kode program> akan diulang sampai tercapai nilai yang tertera pada <akhir>.

5) DO WHILE <kondisi>

<blok kode program>

LOOP

Bagian <blok kode program> akan diulang selama <kondisi> bernilai True. Pengulangan berhenti bila <kondisi> sudah bernilai False.

6) DO UNTIL <kondisi>

<blok kode program>

LOOP

Bagian <blok kode program> akan diulang sampai <kondisi> bernilai True. Pengulangan berhenti bila <kondisi> sudah bernilai True.

2.3.8 *Procedure*

Procedure adalah blok kode program yang berisi perintah-perintah untuk mengerjakan tugas tertentu. Bila di dalam kode program yang kita buat ada perintah-perintah untuk melakukan tugas yang sama di beberapa tempat, maka akan lebih baik perintah-perintah tersebut dibuat dalam sebuah *procedure*. Kemudian procedure itu bisa dipanggil bila diperlukan.

Jenis-jenis procedure pada Visual Basic adalah :

1) *Procedure Sub*, yaitu *procedure* yang tidak mengembalikan nilai setelah selesai dijalankan. Bentuk penulisannya :

[PUBLIC | PRIVATE] SUB <nama_sub> ([<argumen>])

…

<isi procedure>

…

END SUB

2) *Procedure Function*, yaitu *procedure* yang mengembalikan nilai setelah selesai dijalankan. Bentuk penulisannya :

[PUBLIC | PRIVATE] FUNCTION <nama_fungsi> ([<argumen>]) AS <tipe data>

…

<isi prosedure>

…

END FUNCTION

Untuk menggunakan sebuah prosedur, maka prosedur tersebut harus dipanggil. *Syntax* <nama_sub> atau <nama_fungsi> dibuat sebagai pengenal prosedur saat dipanggil.

Pernyataan pada saat akan memanggil suatu procedure sub adalah :

Call <nama_sub>([<argumen>])

Sedangkan *procedure function* bisa dipanggil langsung dengan menyisipkannya di dalam kode program yang memanggilnya.

2.4 *Structured Query Language* (SQL)

SQL digunakan sebagai sarana komunikasi dengan basis data relational yang memperbolehkan data untuk didefinisikan, dimodifikasi, dan dikontrol. Dengan menggunakan perintah SQL, dapat diambil data sesuai dengan kriteria yang telah ditentukan.

Dalam bahasa SQL pada umumnya informasi tersimpan dalam tabel-tabel yang secara logik merupakan struktur dua dimensi terdiri dari baris (*row* atau

record) dan kolom (*column* atau *field*). Sedangkan dalam sebuah *database* dapat terdiri dari beberapa tabel.

Bahasa SQL merupakan bahasa yang berisi pernyataan-pernyataan sederhana yang mudah dipahami. Beberapa pernyataan dalam SQL yang sering digunakan yaitu : *create, insert, delete,* dan *update.*

2.4.1 *Create*

Adalah perintah yang dapat digunakan untuk membuat database baru dan digunakan juga untuk membuat tabel-tabel pada database. Syntax penulisan untuk membuat database baru secara umum adalah :

CREATE DATABASE <nama_ database>;

Sedangkan syntax untuk membuat tabel adalah :

CREATE TABLE nama_tabel

(struktur tabel)

Contoh :

CREATE TABLE petugas_tlp

(id_ptgs int(6) not null primary key,

nama_ptgs char(30) not null,

area char(6) not null,

2.4.2 *Insert*

Adalah perintah untuk memasukkan sebuah *record* ke dalam tabel yang ada. Penulisannya adalah sebagai berikut :

INSERT INTO nama_tabel VALUES (data_kolom1, data_kolom2,…)

Contoh :

INSERT INTO petugas_tlp VALUES (‘810029’, ‘alfon’, ‘Area_3’, ‘+628128282977);

2.4.3 *Select*

Isi tabel dapat ditampilkan dengan menggunakan perintah *select*. Biasanya di dalam menampilkan isi dari tabel dilakukan pemilihan kriteria data yang akan ditampilkan, sehingga perintah select sering kali dikombinasikan dengan perintah where. Cara penulisan perintah select adalah :

SELECT kolom FROM nama_tabel WHERE kriteria_pemilihan

Contoh :

SELECT * FROM petugas_tlp WHERE id_ptgs=’810029’

2.4.4 *Delete*

Untuk menghapus suatu *record* sesuai dengan kriteria tertentu, digunakan perintah ini. Cara penulisan perintah delete adalah :

DELETE FROM nama_tabel WHERE kiteria_pemilihan.

Contoh :

DELETE FROM petugas_tlp WHERE id_ptgs like ‘53%’ AND area = ’area_3’’;

Artinya, menghapus record petugas yang id_ptgsnya sama dengan ‘53%’ (53 kebawah) dan yang bertugas`di area-3.

2.4.5 *Update*

Adalah perintah untuk merubah atau memodifikasi isi record tertentu. Syntaxnya sebagai berikut :

UPDATE nama_table SET kolom1 = nilai_baru1, kolom2 = nilai_baru2 WHERE kriteria;

Contoh :

UPDATE petugas_tlp SET no_hp = '+622168537697, nama = 'joni' WHERE id_ptgs='810029';

Artinya, rubah no_hp dan nama petugas yang memiliki id_ptgs = '810029' menjadi no_hp baru yaitu '+622168537697' dan nama yang baru adalah 'joni'.

2.5 *Microsoft Access 2003*

Microsoft Access adalah suatu relational sistem *database management* dari *Microsoft,* terpaket dengan *Microsoft Office Profesional* yang mengkombinasikan relasional *Microsoft Jet Database Engine* dengan *Graphical User Interface*.

Microsoft Access dapat menggunakan data yang disimpan di dalam *Access/Jet, SQL Server, Oracle, atau ODBC-compliant data container* manapun. Microsoft Access mendukung teknik *Object Oriented*. Microsoft Access versi 1.0 di perkenalkan pertama kali pada November 1992.

2.5.1 *Object*

Ada empat *object* penting dalam Access yaitu :

1) Tabel

Tabel adalah sebuah object utama dalam sebuah database. Tabel terdiri dari baris dan kolom. Baris pertama dari tiap kolom disebut heading. Baris kedua, ketiga dan seterusnya dalam sebuah tabel adalah data sebenarnya dari tabel tersebut. Tiap unit data dari sebuah tabel juga disebut record. Kolom dalam terminologi Access disebut juga *field*.

2) Form

Form digunakan untuk menampilkan data yang tersimpan di dalam tabel. Dengan menggunakan form aktifitas memasukkan dan menampilkan informasi tertentu menjadi lebih mudah. Form tidak digunakan untuk menyimpan data, karena data dalam sebuah form berasal dari tabel atau query.

3) Query

Query berguna untuk menampilkan, mengubah dan menganalisis data. Query merupakan satu dari dua sumber data selain form. Data dari query dan tabel digunakan pula untuk form dan report. Query mirip sebuah tabel, namun query berbeda dari tabel. Query menampilkan data dari tabel dalam bentuk yang dinamis. Query yang paling umum adalah select query. Query ini menampilkan data dari satu atau beberapa tabel berdasarkan kriteria yang ditentukan oleh user. Query juga bisa menampilkan data dengan urutan yang ditentukan

4) Report

Report digunakan untuk merangkum dan mencetak data secara efektif. Dengan data dari tabel yang sama, report dapat menyajikannya dalam tampilan yang berbeda-beda berdasarkan field yang diinginkan untuk disajikan.

2.5.2 Tipe data *field*

Access menggunakan berbagai macam tipe data field untuk jenis-jenis data yang berbeda-beda. Field-field dalam sebuah tabel haruslah diberi tipe data yang cocok agar Access tahu bagaimana harus mengolah dan memperlakukan data dalam field itu. Berikut tipe-tipe data pada Access:

1) Text

Ini adalah tipe data yang paling umum digunakan. Tipe data ini bisa digunakan untuk field alfanumeric seperti nama, alamat, kode pos dan nomor telpon. Sampai dengan 255 karakter bisa dimasukkan ke field dengan tipe data ini meskipun defaultnya 50 karakter.

2) Memo

Tipe data ini mirip dengan text namun bisa menampung hingga 64000 karakter. Penggunaan yang tidak tepat dengan tipe data ini bisa mengakibatkan besarnya ukuran file database yang tentu saja akan memberatkan sistem sendiri.

3) Number

Tipe data ini digunakan untuk menyimpan data numeris yang akan digunakan untuk perhitungan matematis (kecuali perhitungan uang atau yang memerlukan tingkat akurasi yang tinggi).

4) Date/Time

Tipe data ini digunakan untuk nilai tanggal dan waktu dalam periode 100 dan 9999.

5) Currency

Tipe data ini terutama digunakan untuk nilai mata uang.

6) AutoNumber

Ini adalah angka urut atau angka acak yang ditetapkan Access untuk record baru yang ditambahkan ke sebuah tabel. Field dengan tipe data ini tidak bisa di-edit. Tipe data ini digunakan untuk field yang harus unik dimana tidak boleh ada nilai yang sama dalam field tersebut

7) Yes/No

Tipe data ini hanya bisa mempunyai dua nilai : Yes atau No. Ini digunakan untuk field yang hanya punya dua nilai.

8) OLE Object

Tipe data ini digunakan untuk eksternal object, seperti bitmap atau file suara.

9) Hyperlink

Ini merupakan gabungan dari teks dan bilangan yang disimpan sebagai alamat hyperlink. Hyperlink digunakan untuk menyimpan pointer ke sebuah situs internet atau path ke sebuah file di harddisk.

10) Lookup Wizard

Bila menggunakan tipe data ini untuk sebuah field, maka dimungkinkan memilih sebuah nilai dari tabel lain atau dari sebuah daftar nilai yang ditampilkan dalam sebuah kotak daftar atau kotak combo