BAB II
TINJAUAN PUSTAKA

2.1 PERKEMBANGAN CLIPPER 5.3.

Clipper 5.3. pada waktu pemunculannya hanyalah sekedar program compiler dbase,yaitu untuk mengubah program dbase (berjenis PRG) menjadi program yang berdiri sendiri dalam bahasa mesin (berjenis EXE) , sehingga dapat langsung dijalankan dari sistem operasi dan berjalan lebih cepat. Clipper compiler ini terus berkembang dari versi ke versi terutama terjadi perkembangan yang sangat cepat pada versi winter 85, autumn 85,dan summer 87/88. Tetapi baru pada summer 87/88 itulah program clipper compiler menjadi demikian populer. Banyak programmer yang sangat mengandalkan kemampuan clipper , karena pada versi summer 87/88. Clipper bukan lagi hanya sekedar compiler dbase ,melainkan sudah menjadi program database yang berdiri sendiri, bahkan kedalamnya ditambahkan fungsi-fungsi yang sangat didambakan programmer, seperti user interface yang membuat penampilan lebih baik, menarik serta mudah diatur, programming dalam orientasi tabel, fungsi-fungsi menu dan menginginkan pemakai mendefenisikan fungsi sendiri. Teknologi yang dibawa clipper pada waktu itu merupakan hal yang baru dan merupakan lompatan teknologi dibandingkan program database lain, sehingga membuat clipper makin banyak dipakai.

Perkembangan selanjutnya adalah pada saat diluncurkannya clipper compiler versi 5.0 pada versi baru ini clipper sudah mulai menginjak teknologi

object oriented programming (OOP) yang memang pada saat itu sedang trend .Konsep OOP tersebut semakin memacu daya guna dan kemampuan program clipper . Tidak lama kemudian muncul versi 5.01 sampai versi ini clipper masih dibawah perusahaan nantucket. Barulah pada versi 5.2 clipper diambil alih oleh perusahaan komputer Associated dan diberi nama CA CLIPPER 5.2.

2.2. MENULIS PROGRAM

Untuk menulis program clipper, anda bisa menggunakan program editor apa saja asal berbasis teks (ASCII) misalkan side kick, edit dari dos, QE atau editor yang bawaan clipper sendiri, yaitu program editor (PE) dan lain-lain.

Program yang ditulis sebenarnya hanyalah kumpulan teks yang tidak dimengerti oleh komputer. Program yang sedang ditulis tersebut menggunakan extention atau akhiran *.prg dan kemudian program ini disebut sebagai kode sumber (source kode). Agar kode program ini dimengerti oleh komputer, maka harus melewati tahap kompilasi dan tahap linking.

2.3. PROSEDUR KOMPILASI DAN LINKING

Program sumber yang sudah dibuat supaya bisa dilaksanakan oleh komputer harus diubah kedalam bahasa mesin melalui prosedur kompilasi dan linking.

2.3.1. Prosedur kompilasi

untuk mengkompilasi program sumber yang sudah dibuat, dapat mengikuti cara-cara sebagai berikut:

Dari prompt c:/ ketikan c:/clipper [nama file].

Misalkan c:/clipper coba 1.prg

2.3.2. Prosedur linking

untuk mendapatkan kode dalam bahasa mesin, object kode yang telah terbentuk harus digabungkan dengan kode mesin yang ada dalam daftar clipper. Sintak linking adalah sebagai berikut: c:/rtlink fl coba1 <enter>.

Setelah proses selesai, anda akan mendapatkan file berjenis EXE (executable code) dari kedua prosedur linking tersebut, sebenarnya masih ada cara yang lebih singkat, yaitu dengan cara mengetikan perintah berikut pada prompt dos c:/cl nama file <enter> misalnya c:/ cl coba1 <enter>.

Cl adalah nama file batch yang secara otomatis dibuat saat proses instalasi clipper kedalam hardisk, jika file tidak ada, kita dapat membuatnya sendiri.

2.4. JENIS DATA DALAM CLIPPER

Didalam clipper terdapat beberapa jenis data. Adapun jenis data yang dipunyai clipper adalah.

2.4.1 CHARACTER

Data character digunakan jika data berbentuk string misalnya nama, alamat, telefon, kota dan lain-lain.

2.4.2 DATE

Tipe data date digunakan untuk mewakili data tanggal, clipper dapat mengoperasikan data tanggal mulai tanggal 1 januari 1900 sampai 31 desember 2999 clipper juga dapat membuat tanggal kosong.

2.4.3 LOGICAL

Data logical yaitu data yang mewakili 2 kondisi, benar (true) dan salah (false) dalam clipper 5.3. data logical dapat diindeks.

2.4.4 NUMERIC

Data numeric yaitu data angka yang bisa dioperasikan secara matematik. CA CLIPPER 5.3 dapat menampung data numeric sampai 10 pangkat –308 sampai 10 pangkat 308 dengan ketelitian dibelakang titik desimal sampai dengan 16 digit.

2.4.5 MEMO

Data memo seperti data character, tetapi biasanya untuk data string yang sangat besar dan atau terdiri dari beberapa baris, jika anda membuat data character pada sebuah file DBF, data tersebut mempunyai sama pada setiap recordnya, tetapi untuk data memo, maka masing-masing record bisa mempunyai panjang data yang berlainan.

1.4.6 ARRAY

Data array yaitu satu atau lebih variabel yang mempunyai nama yang sama .Jika satu nama tersebut berisi beberapa variabel maka sebagai pembedanya adalah nomor elemen (index/subcript) variabel tersebut.

2.4.7 NIL

Data nil yaitu untuk menampung sesuatu yang tidak ada isinya. Biasanya dipakai jika anda membutuhkan argumen, tetapi tidak mendefenisikan isinya.

2.4.8 OBJECT

Dengan data baru ini clipper sudah mengarahkan dalam OOP. Clipper menggunakan data object berupa data cllas.

Dari data diatas, yang ada untuk file DBF hanya lima yaitu character, date, logical, memo dan numeric.

2.5 OPERATOR DALAM CLIPPER

CA clipper 5.3 mempunyai operator-operator yang dapat dikelompokan sebagai berikut:

- OPERATOR MATEMATIK
- OPERATOR ASSIGNMENT
- OPERATOR RELASI
- OPERATOR UNTUK GUNAAN KHUSUS

2.5.1. OPERATOR MATEMATIK

Tabel 2.5.1. Operator Matematik

OPERATOR	KEGUNAAN	UNTUK JENIS DATA
**.^	EKSPONENSIAL	N
*	PERKALIAN	N
/	PEMBAGIAN	N
%	MODULUS(sisa pembagian)	N
+	TAMBAH	N,D
-	KURANG	N
-	SAMBUNG	C
++	BERTAMBAH(naik)	N,D
--	BERKURANG(turun)	N,D

KETERANGAN :

C=CHARACTER D=DATE N=NUMERIK

2.5.2. OPERATOR RELASI

Digunakan untuk membandingkan dua buah data dan menghasilkan data logika,true(.T.) berarti benar atau false(.F.) berarti salah.

Tabel Operator Relasi

OPERATOR	KONDISI	UNTUK JENIS DATA
<>!=#	TIDAK SAMA	C,D,L,N,M,NIL
<	LEBIH KECIL	C,D,L,N,M,A,O,NIL
>	LEBIH BESAR	C,D,L,N,M
<=	LEBIH KECIL/SAMA DENGAN	C,D,L,N,M
=	SAMA DENGAN	C,D,L,N,M,NIL
>=	LEBIH BESAR/SAMA DENGAN	C,D,L,M,N
$	ADA DALAM STRING	C,M

KETERANGAN :

C=CHARACTER A=ARRAY

N=NUMERIC

O=OBJECT

D=DATE

L=LOJIK

M=MEMO

NIL=NIL

3 OPERATOR LOGIKA

Digunakan untuk menggabungkan beberapa ekpresi logika dan akan menghasilkan sebuah ekspresi logika dari beberapa ekspresi logika tersebut.

- .And. Ekspresi dan
- .Or. Ekspresi atau
- .Not.,! Ekspresi tidak

2.6 STATEMENT KENDALI

digunakan untuk memilih bagian program yang akan dikerjakan sesuai kondisi yang diberikan, Ada 2 buah statement kondisi yang ada dalam clipper yaitu stsment **IF** dan statement **do case.**

2.6.1 STATEMENT IF

Statement **if** akan mencetak suatu kondisi dan menentukan apakah kondisi tersebut benar atau salah. Kemudian melakukan suatu kegiatan sesuai dengan nilai kondisi tersebut.

Struktur dari **if** dapat berupa:

If..enc if

If..elsa..endif

If…elseif….else…endif

2.6.1.1 STRUKTUR IF..ENDIF

IF ungkapan

statement

ENDIF

2.6.1.2 STRUKTUR IF..ELSEIF..ELSE..ENDIF

IF ungkapan

Statement 1

ELSE

Statement 2

ENDIF

2.6.1.3 STRUKTUR IF..ELSEIF..ELSE..ENDIF

IF ungkapan 1

Statement 1

Else if ungkapan 2

Statement 2

Elseif ungkapan –n

Statement –n

Else

Statement terakhir

Endif

KETERANGAN:

Ungkapan adalah kondisi yang akan diselesaikan oleh statement if ,sampai kondisi yang diseleksi terpenuhi kemudian diproses.

2.6.2 statement DO CASE..case..otherwise..ENDCASE

Bentuk dari struktur if ..ejseif..else..endif yang komplek seperti diatas sebenarnya dapat disusun dengan cara yang lebih sederhana yaitu dengan menggunukan statement DO CASE aturan penulisan

DO CASE

CASE<KONDISI-1>

statement-1

CASE <KONDISI-2>

Statement-2

CASE <KONDISI-N>

Statement-n

OTHERWISE

Statement terakhir

ENDCASE

2.7 MACAM-MACAM PERULANGAN

Salah satu p oses yang hampir selalu ada di dalam pemrograman adalah proses berulang (looping). Proses berulang adalah suatu proses dimana komputer akan mengeksekusi satu atau lebih statement berulang kali sesuai kebutuhan.

2.7.1. Perulangan FOR...NEXT

Perulangan dengan statement for...next digunakan untuk mengulang statement atau satu blok statement berulang kali, sejumlah yang ditentukan .

Aturan penulisan:

FOR <counter>=<awal>TO<akhir>[STEP]<[-]selang>]

<statement>

[EXIT]

[LOOP]

<statement>

NEXT

KETERANGAN :

1. Perintah (Statement) yang ada diantara FOR..NEXT akab diulang mulai dari nilai <awal> sampai nilai<akhir> dengan pertambahan sebesar<selang>.
2. Perintah exit digunakan bila kita akan keluar dari perulangan sebelum mencapai akhir dari perualngan.
3. Perintah loop digunakan bila kita akan mengulang statement pertama pengulangan tapi melewati statement dibawahnya.

2.7.2 PERULANGAN DO ...WHLE ENDDO

Perulangan ini menggunakan statement while..do yang mempunyai struktur dalam penulisannya adalah sebagai berikut:

DO while <kondisi>

<statement>

[EXIT]

[LOCP]

,statement>

ENDDO

KETERANGAN :

1. Perintah <Statement> yang ada diantara do ...enddo akan diulang mulai selama kondisi bernilai benar(.T.)
2. Perintah exit digunakan bila kita akan keluar dari perulangan sebelum mencapai akhir dari perulangan.
3. Perintah loop digunakan bila kita akan mengulang statement pertama perulangan tanpa melewati statement dibawahnya.