

BAB II

LANDASAN TEORI

2.1. KONSEP DATABASE MANAJEMEN SYSTEM

Suatu Database manajemen system (DBMS) berisi satu koleksi data yang saling berelasi dan suatu set program untuk mengakses data tersebut. Jadi DBMS terdiri dari database dan set program pengelola untuk menambah data tersebut. Jadi DBMS terdiri dari database dan set program pengelola untuk menambah data, menghapus data, mengambil dan membaca.

Database adalah kumpulan file-file yang saling berelasi, relasi tersebut bisa ditunjukkan dengan kunci dari tiap-tiap file yang ada. Satu database menunjukkan satu kumpulan data yang dipakai dalam satu lingkup perusahaan atau instansi.

Dalam satu file terdapat record-record yang sejenis, sama besar, sama bentuk, merupakan satu kumpulan entity yang seragam. Satu record terdiri dari field-field yang saling berhubungan untuk menunjukkan bahwa field tersebut dalam satu pengertian yang lengkap dan direkam dalam satu record.

Untuk menyebut isi dari field maka digunakan attribute atau merupakan judul dari satu kelompok entity tertentu, misalnya attribute alamat menunjukkan entity alamat dari mahasiswa. Entity adalah obyek yang nyata dan akan direkam.

Set program pengelolah merupakan satu paket program yang dibuat agar memudahkan dan mengefisienkan pemasukkan atau perekaman informasi dan pengambilan atau pembacaan informasi ke dalam database.

2.1.1. DEFINISI

A. Entity

Entity adalah orang, tempat, kejadian/konsep yang informasinya direkam. Pada bidang administrasi mahasiswa misalnya, entity adalah siswa, buku, pembayaran, nilai test. Pada bidang kesehatan, entity adalah pasien, dokter, obat, kamar, diet.

B. Attribute

Setiap entity mempunyai attribute atau sebutan untuk mewakili suatu entity. Seorang mahasiswa dapat dilihat dari atributnya, misalnya nama, nomor register, alamat, nama orang tua, hobby. Attribute juga disebut sebagai data elemen, data field, data item.

C. Data Value (Nilai atau Isi Data)

Data Value adalah data aktual atau informasi yang disimpan pada tiap data elemen atau attribute. Attribute nama karyawan menunjukkan tempat dimana informasi nama karyawan disimpan, sedang data value adalah Sutrisno, Budiman, merupakan isi data nama karyawan tersebut.

D. Record

Kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entity secara lengkap. Satu record mewakili satu data atau informasi tentang seseorang misalnya nomor karyawan, nama karyawan, alamat kota tanggal masuk.

E. File

Kumpulan record-record sejenis yang mempunyai panjang elemen yang sama, attribute sama namun berbeda data valuenya.

F. Database

Kumpulan file-file yang mempunyai kaitan antara satu file dengan file yang lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu. Bila terdapat file yang tidak dapat dipadukan atau dihubungkan dengan yang lainnya berarti file tersebut bukanlah kelompok dari satu database, ia akan dapat membentuk database sendiri. Database dapat juga diartikan kumpulan data terorganisasi yang digunakan untuk informasi. Meskipun saat ini kita memfokuskan pada penggunaan komputer untuk pengolahan data, database tidak harus berbasis komputer. Buku, alamat dan nomor telepon adalah suatu database.

G. Database Manajemen System (DBMS)

Kumpulan file yang saling berkaitan bersama dengan program untuk pengelolannya disebut sebagai DBMS. Database adalah kumpulan datanya, sedangkan program pengelolanya berdiri sendiri dalam satu paket program yang komersial untuk membaca data, mengisi data, menghapus data serta melaporkan data dalam database.¹

¹ Kristanto Harianto, Ir, Konsep dan perancangan DATABASE, (Yogyakarta : ANDI OFFSET, 1994), hal 1-5.

2.2. KEGUNAAN DATABASE

Penyusunan satu database digunakan untuk mengatasi masalah-masalah pada penyusunan data, yaitu :

- Redundansi dan Inkonsistensi Data
- Kesulitan Pengaksesan Data
- Isolasi Data Untuk Stand arisasi
- Multiple User
- Masalah Keamanan
- Masalah Integrasi
- Masalah Data Independence

2.2.1. Redundansi dan Inkonsistensi Data

Jika file-file dan program aplikasi diciptakan oleh programmer yang berbeda pada waktu yang berselang cukup panjang, maka ada beberapa bagian data mengalami penggandaan pada file-file yang berbeda.

Sebagai contoh : alamat dan nomor telepon dari pelanggan kita tercatat pada file deposito juga pada file rekening koran juga file nasabah.

Penyimpanan di beberapa tempat untuk data yang sama ini disebut sebagai redundansi dan mengakibatkan pemborosan ruang penyimpanan dan juga untuk mengakses jadi lebih tinggi. Penyimpanan data yang sama berulang-ulang di beberapa file dapat mengakibatkan juga inkonsistensi (Tidak konsisten). Hal ini dapat terjadi bila suatu ketika pelanggan tersebut pindah alamat dan nomor teleponnya berubah, maka seharusnya ketiga file

yang memuat data tersebut harus di ubah atau di update. Bila salah satu dari file yang mengandung data tersebut terlewat diupdate maka terjadilah tidak konsisten tadi.

2.2.2. Kesulitan Dalam Pengaksesan Data

Pada suatu saat dibutuhkan untuk mencetak data siapa saja pelanggan yang berada di kode pos 551229 Yogyakarta, padahal belum tersedia program yang telah ditulis untuk mengeluarkan data tersebut. Maka kesulitan tersebut timbul, dan penyelesaian untuk itu adalah kearah DBMS yang mampu mengambil data secara langsung dengan mudah digunakan (User Fricnly).

2.2.3. Isolasi Data Untuk Standarisasi

Jika data tersebar dalam beberapa file dalam bentuk format yang tidak sama, maka ini menyulitkan dalam menulis program aplikasi untuk mengambil dan menyimpan data. Maka haruslah data dalam satu database dibuat satu format sehingga mudah dibuat program aplikasinya. Dapat dibayangkan betapa sulitnya membuat program aplikasi bilamana data dibuatdari format text file Pascal, BASIC, C++ dan juga format dari versi lotus 123 dan lainnya.

2.2.4. Multiple User

Dalam rangka mempercepat semua daya guna sistem dan mendapat responsi waktu yang cepat, beberapa sistem mengijinkan banyak pemakai untuk mengupdate data secara simultan. Salah satu alasan mengapa database dibangun, karena nantinya data tersebut digunakan oleh banyak orang dalam waktu yang berbeda, diakses oleh program program yang sama tapi berbeda orang dan waktu. Semua ini memungkinkan terjadi karena data yang diolah tidaklah tergantung dan menyatu dalam program tapi ia terlepas dalam satu kelompok data.

2.2.5. Masalah Keamanan

Tidak setiap pemakai sistem database diperbolehkan untuk mengakses semua data. Misalkan data mengenai gaji pegawai hanya boleh dibuka oleh bagian keuangan dan personalia, tidak diperkenankan bagian gudang membaca dan mengubahnya.

Keamanan ini dapat diatur lewat program yang dibuat oleh programmer atau fasilitas keamanan dari operating sistem, misalnya Novell Netware untuk Local Area Network.

2.2.6. Masalah Integritas

Database berisi file-file yang saling berkaitan, masalah utama adalah bagaimana kaitan antara file tersebut terjadi. Meskipun kita

mengetahui bahwa file A berkaitan dengan file B, namun secara teknis maka ada field kunci yang mengaitkan kedua file tersebut.

2.2.7. Masalah Data Independence

Pada suatu aplikasi yang kita buat dengan bahasa pemrograman BASIC misalnya, bila program telah dibuat untuk menyelesaikan masalah pembacaan data untuk file pelanggan dengan field no, nama, alamat, maka setelah program jadi dan terdapat perubahan struktur file pelanggan, maka program tersebut haruslah diubah. Hal ini disebut bahwa program yang telah dibuat tidak bebas terhadap database yang ada.

Berlainan dengan paket bahasa yang diciptakan dari DBMS, apapun yang terjadi pada struktur file, setiap kali kita hendak melihat data cukuplah dengan utility LIST, ingin menambah data cukup dengan APPEND. Ini berarti perintah-perintah dalam paket DBMS bebas terhadap database. Apapun perubahan dalam database, semua perintah akan mengalami kestabilan tanpa perlu ada yang diubah.

Namun perlu dipikirkan bagaimana jika ada syarat terhadap database yang ada. Misalnya syarat pengambilan uang tabungan Bank adalah harus disisakan paling minimal 25000. Syarat ini tentu dimasukkan dalam aplikasi program yang dibangun. Namun bagaimanakah jika suatu ketika syarat itu telah bergeser dari syarat 25000 menjadi 50000. Disini

aplikasi program yang telah dibangun haruslah diubah dan ini menunjukkan ketidakbebasan program yang dibuat terhadap database.²

2.3. TEKNIK PERANCANGAN DATABASE

Merancang Database merupakan suatu hal yang sangat penting. Kesulitan utama dalam merancang database adalah bagaimana merancang sehingga database dapat memuaskan keperluan saat ini dan masa mendatang. Dalam hal ini penulis menggunakan teknik Entity Relationship.

Pada model data relation antar file direlasikan dengan kunci relasi (relation key), yang merupakan kunci utama dari masing-masing file. Perancangan database yang tepat akan menyebabkan dBASE III, FOXBASE, Clipper dan paket program lainnya akan bekerja secara optimal.

2.3.1. Entity Relationship Konsep

Relasi antara dua file atau dua tabel dapat dikategorikan menjadi tiga macam. Demikian pula untuk membantu gambaran relasi secara lengkap terdapat juga tiga macam relasi dalam hubungan attribute dalam satu file.

A. One to one Relationship 2 file

Hubungan antara file pertama dengan file kedua adalah satu berbanding satu. Seperti pada pelajaran privat dimana satu guru mengajar satu siswa dan satu siswa hanya diajar satu guru pula.

² Ibid, hal 5-7

B. One to many Relationship 2 file

Hubungan antara file pertama dengan file kedua adalah satu berbanding banyak. Seperti pada sistem pengajaran di sekolah dimana satu guru mengajar banyak siswa dan banyak siswa hanya diajar satu guru pula.

C. Many to many Relationship 2 file

Hubungan antara file pertama dengan file kedua adalah banyak berbanding banyak. Seperti pada sistem pengajaran di perguruan tinggi dimana satu Dosen mengajar banyak Mahasiswa dan banyak Mahasiswa hanya diajar banyak Dosen pula.

D. Relasi one to 2 attribute dalam 1 file

Hubungan antara satu attribute dengan attribute yang lain dalam satu file yang sama mempunyai hubungan satu lawan satu. Misalnya attribute nomor pegawai yang unik dan attribute nomor ktp pegawai tersebut mempunyai hubungan satu lawan satu. Satu pegawai hanya satu nomor ktp, tidak ada yang berganda.

E. Relasi many to one 2 attribute dalam 1 file

Hubungan antara satu attribute dengan attribute yang lain dalam satu file yang sama mempunyai hubungan satu lawan banyak. Misalnya pada satu alamat pegawai yang terdapat puluhan pegawai

kantor tersebut, maka hubungan antara attribute alamat pegawai dengan nomor pegawai adalah satu alamat menunjukkan banyak pegawai

F. Relasi many to many 2 attribute dalam 1 file

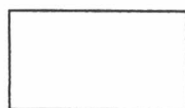
Hubungan antara satu attribute dengan attribute yang lain dalam satu file yang sama mempunyai hubungan banyak lawan banyak. Misalnya pada satu alamat pegawai yang terdapat puluhan pegawai kantor tersebut, maka hubungan antara attribute alamat pegawai dengan nomor pegawai adalah satu alamat menunjukkan nama pegawai adalah beberapa alamat menunjukkan nama pegawai yang sama dan sebaliknya.

2.4. RANCANG SISTEM

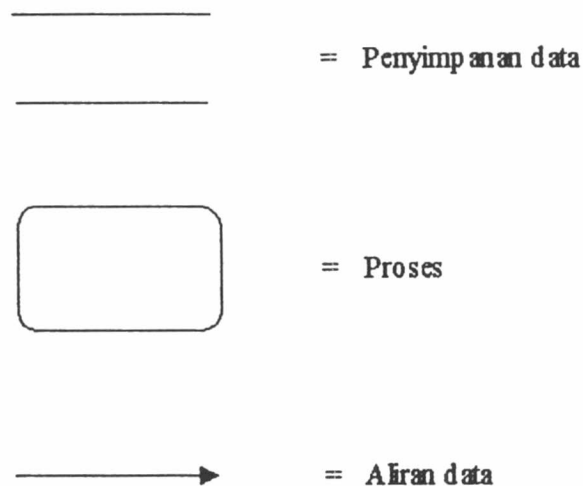
2.4.1. Data Flow Diagram

Data Flow diagram adalah gambaran sistem secara logikal, gambaran ini tidak tergantung pada perangkat lunak, perangkat keras, struktur data maupun organisasi file.

Simbol – simbol yang digunakan adalah :



= Sumber atau Tujuan data



Gambar 2.1. Simbol-simb ol aliran Data Flow Diagram

Proses pada Data Flow Diagram dapat merupakan sekumpulan program, satu program atau model sub program, dapat juga merupakan transformasi data secara manual. Penyimpanan data merupakan file, elemen dari satu database, atau satu bagian dari record. Penyimpanan data dapat berupa disket, magnetic drum dan magnetic tape.

2.4.2. Data Dictionary

Data Dictionary merupakan kumpulan data mengenai data-data. Tujuannya adalah untuk memberikan informasi mengenai definisi, struktur, pemakai dan masing-masing elemen.

Contoh Data Dictionary :

Data Karyawan = Nama +

Alamat +

Status +

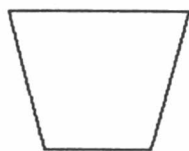
Tempat Lahir +

Tanggal Lahir

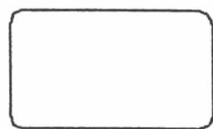
2.4.3. Sistem Flow Chart

Sistem FlowChart merupakan alat bantu yang banyak digunakan untuk menggambarkan sistem secara psikal. Sistem FlowChart ini sangat membantu dalam memahami langkah-langkah pengerjaan suatu program.

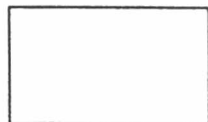
Simbol- simbol yang digunakan dalam Sistem FlowChart adalah :



= Manual Operation



= Alternate proses



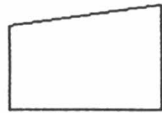
= Process



= Chard



= Sequential access storage



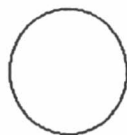
= Manual Input



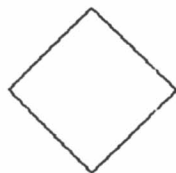
= Display



= Document



= Connector



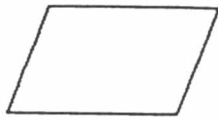
= Decision



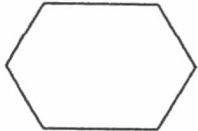
= Off- Page Connector



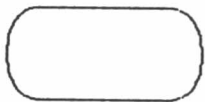
= Magnetic disk



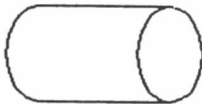
= Data



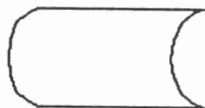
= Preparation



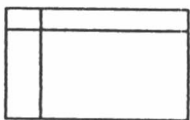
= Terminator



= Direct Acces Storage



= Storage Data



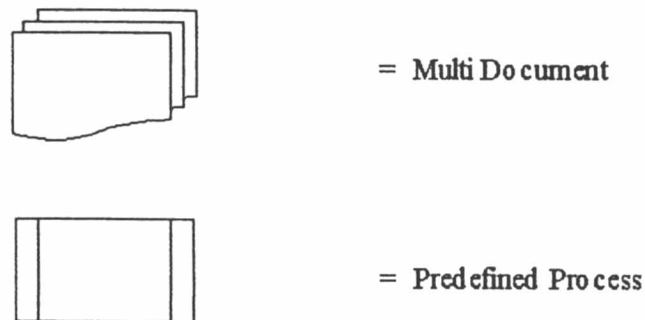
= Internal Storage



= Punched Tape



= Delay



Gambar 2.2. Simbol-simbol Sistem FlowChart

2.5. BAHASA PEMROGRAMAN

2.5.1. CA Clipper 5.3

Program-program untuk sistem ini menggunakan bahasa pemrograman CA Clipper 5.3. Dalam hal ini penulis menggunakan bahasa pemrograman Clipper karena bahasa pemrograman ini masih serumpun dengan bahasa pemrograman Database. Bahasa pemrograman Clipper ini sebagai bahasa pemrograman yang menjembatani bahasa pemrograman antara FOXBase dan FoxPro, perintah-perintah yang digunakan juga tidak jauh berbeda, hanya saja ada fasilitas-fasilitas yang hanya ada pada Clipper. Seperti fasilitas Compiler ataupun Linker yang tidak ada pada dBASE maupun FoxBase.

Clipper merupakan salah satu software Sistem Manajemen Database yang mempunyai banyak kemampuan. Selain sebagai compiler, Clipper dapat juga dipakai sebagai bahasa pemrograman yang pada intinya mirip dengan dBASE III Plus. Hampir semua instruksi yang ada pada dBASE III Plus dapat dipakai dalam

Clipper. Di samping itu, Clipper juga menyediakan tambahan instruksi-instruksi yang dapat digunakan untuk membantu pemakai dalam menangani masalah :

- Pengaturan Layar
- Low Level File Access (File Handle)
- Manipulasi Data
- Proses Kontrol
- Interface UDF melalui C atau Assembler dan kemampuan untuk membentuk fungsi kreasi sendiri³

2.5.2. Jenis Data

Jenis – jenis data yang dipakai Clipper adalah :

Date

Data date digunakan untuk mewakili data tanggal. Clipper bisa mengoperasikan data tanggal mulai tanggal 1 Januari 1900 sampai dengan Desember 2999. Clipper juga bisa membuat tanggal kosong.

Format tanggal yang dimiliki Clipper sebagai berikut :

<u>Format Tanggal</u>	<u>Bentuk Keluaran</u>
Amerika	mm/dd/yy
Ansi	yy.mm.dd
British	dd/mm/yy
French	dd/mm/yy

³ Gunawan Ganiadi, Teknik Pemrograman dan Trik untuk Clipper (Jakarta, PT. Elex Media Komputindo), hal VII

German	dd.mm.yy
Italian	dd-mm-yy
Japanese	yy/mm/dd
USA	mm-dd-yy

Character

Data Character digunakan jika data berbentuk string.

Misalnya :

Nama, alamat, telepon dan lain-lain. Data Character bisa menampung Null string sampai string berukuran 64 KB.

Logical

Data Logical yaitu data yang mewakili dua kondisi, yaitu benar/ true (.T.) dan salah/false (.F.). Data logical dalam Clipper 5.3 bisa diindex.

Numerik

Data Numerik yaitu data angka yang bisa dioperasikan secara matematik. CA Clipper 5.3 dapat menampung data numerik 10 pangkat -308 sampai 10 pangkat 308 dengan ketelitian dibelakang titik desimal sampai 16 digit.

Memo

Data memo seperti data Character, tetapi biasanya untuk data string yang sangat besar dan terdiri dari beberapa baris. Jika ingin membuat data

character pada sebuah file DBF, data tersebut mempunyai panjang yang sama pada setiap record, tetapi untuk data memo masing-masing record bisa mempunyai panjang data yang berlainan. Seperti data character, satu data memo maksimum berisi 64 KB, tetapi file data memo (berjenis memo) bisa berukuran lebih dari 64 KB, karena merupakan kumpulan dari beberapa memo.

Array

Data Array yaitu satu atau lebih variabel yang mempunyai nama sama. Jika satu nama tersebut berisi beberapa variabel, maka sebagai pembedanya adalah nomor elemen (index atau sub script) variabel tersebut.

Contoh :

```
Local Paket[3]
```

```
Paket[1] := 'dBASE III Plus'
```

```
Paket[2] := 'Wordstar'
```

```
Paket[3] := 'Lotus 123'
```

```
? Paket[1] // dBASE III Plus
```

Sebuah array sebelum bisa dipakai harus dideklarasikan dulu. Cara mendeklarasikan array adalah sebagai berikut :

```
Declare Nama_array [jumlah elemen]
```

```
Local Nama_array [jumlah elemen]
```

Privat Nama_array [jumlah elemen]
Publik Nama_array [jumlah elemen]
Static Nama_array [jumlah elemen]

Selain itu, array bisa juga digunakan dengan menggunakan karakter {} untuk mengapit, tanpa harus dideklarasikan dulu.

Contoh :

```
Paket :={ 'dBASE III Plus', 'Wordstar', 'Lotus 123' }
```

Sebuah array bisa mengandung elemen-elemen dari jenis data yang berlainan.

```
Array :={ 'Budî', 'Donny', 'Tonny', c to d ('03/15/96'), 85 }
```

Code Blok

Yaitu data berisi kode yang terkompilasi. Bentuk umum Code Blok sebagai berikut :

```
{<daftar argumen>|<daftar ekspres> }
```

Nil

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

Contoh :

```
Local Test [4]
```

?Test[1]/Nil
?Test[2]/Nil
?Test[3]/Nil
?Test[4]/Nil
?Test[5]/Error!

Object

Dengan data baru ini Clipper sudah mengarah dalam OOP (Object Oriented Programming). Clipper menggunakan data object berupa data Class.

Data Class yang disediakan oleh Clipper adalah :

- Error Class
- Get Class
- Tbrowse Class
- TBColumn Class

2.5.3. Jenis Operator

CA Clipper 5.3 mempunyai operator-operator yang dapat dikelompokkan sebagai berikut :

Operator Logik

Operator logik digunakan untuk menggabungkan beberapa ekspresi logik dan akan menghasilkan sebuah ekspresi logik dari beberapa ekspresi logik tersebut.

Operator	Test	DiGunakan pada data
.and.	Logik dan	logik
.or.	Logik atau	logik
.not.,!	Logik tidak	logik

Operator Untuk kegunaan Khusus

Operator-operator berikut digunakan untuk keperluan khusus yang tidak masuk dalam kategori operator-operator di atas.

Operator	Kegunaan
&	Operator Makro
⇒	Operator alias
@	Mengirim alamat
{ }	Pembuatan Array
[]	Elemen Array
()	Fungsi atau pengelompokan

Operator Relasi

Operator relasi digunakan untuk membandingkan dua buah data dan menghasilkan data logik, True (.T.) berarti benar atau False(.F.) berarti salah .

Operator	Test	Digunakan Pada Data
<> != #	Tidak sama	C,D,L,M,N,NIL
>	Lebih Besar	C,D,L,M,N
<	Lebih Kecil	C,D,L,M,N
<=	Lebih Kecil atau sama dengan	C,D,L,M,N
=	Sama dengan	C,D,L,M,N,NIL
==	Sama persis	C,D,L,M,N,NIL
>=	Lebih Besar atau sama dengan	C,D,L,M,N
\$	Ada dalam String	C,D,L,M,N

Keterangan :

C = karakter	M = Memo
A = Array	D = Tanggal
L = Logik	N = Numerik
O = Object	NIL = Nil

Operator Assignment

Operator Assignment digunakan untuk memberi harga pada variabel.

Operator	Kegunaan	Digunakan Pada Data
=	Simple Assignment	Semua
:=	In-Line Assignment	Semua
<Op>	Compound Assignment	C,D,L,M,N

Operator Matematika

Operator	Kegunaan	Digunakan pada Data
** , ^	Eksp onensial	N
*	Perkalian	N
/	Pembagian	N
%	Modulus	N
+	Tambah	N,D
+	Sambung	C
-	Kurang	N,D
-	Sambung	C
++	Bertambah (Naik)	N,D
--	Bertambah(Turun)	N,D

Keterangan :

C = Karakter N = Numerik

D = Tanggal

2.5.4. Variabel Memori

Variabel memori adalah suatu tempat dalam memori komputer yang mempunyai nama (sebagai pengenal) dan mempunyai nilai. Nilai dalam variabel bisa diganti setiap saat. Suatu variabel akan hilang dari memori komputer jika menemui pernyataan RETURN pada program/procedure/fungsi yang bersangkutan selesai dilaksanakan. Variabel juga akan hilang jika komputer dimatikan atau sengaja dibenkan perintah menghapus variabel tersebut.

Suatu variabel jika sudah di buat akan dapat diakses secara acak. Nama Variabel maksimum panjangnya 10 karakter, dengan karakter pertama tidak boleh berupa angka. Nama variabel juga tidak boleh mengandung spasi atau tanda baca serta karakter yang dipakai Clipper sebagai "Operator".

Pada CA Clipper 5.0, CA Clipper 5.2, CA Clipper 5.3 ada beberapa tambahan dalam jenis variabelnya. Jenis-jenis variabel yang dimiliki Clipper sebagai berikut :

Variabel Publik

Variabel Publik yaitu variabel yang bisa dikenali oleh semua prosedur dan fungsi yang ada. Biasanya dipakai untuk variabel yang bisa digunakan di semua tempat dan bisa diedit disemua tempat.

Variabel Private

Jika kita mendefinisikan suatu variabel private, maka variabel tersebut :

1. Hanya dikenali oleh procedure/fungsi ditempat variabel didefinisikan atau procedure/fungsi pada level yang lebih bawah (yang dipanggil)
2. Tidak mempengaruhi variabel pada level diatasnya jika ada yang bernama sama.
3. Tidak mengenali variabel dengan nama yang sama pada level diatasnya. Hal ini bisa terjadi karena variabel berjenis private menggunakan memori tersendiri.

Variabel Local

Variabel tersebut mempunyai sifat yang sama dengan private, dengan perkecualian variabel lokal hanya dikenali pada procedure atau fungsi ditempat variabel dideklarasikan. Jadi variabel tersebut tidak akan dikenali pada procedure/ fungsi pada level yang lain.

Variabel Static

Variabel Static mempunyai sifat variabel lokal, dengan kekhususan jika dideklarasikan dan diinisialisasi suatu variabel static dengan nilai tertentu dalam sebuah fungsi atau procedure, maka dia akan diinisialisasikan sekali, walaupun fungsi tersebut dipanggil lagi.