UNIT PELAKSANA TEKNIS PERPUSTAKAAN
UNIKOM

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian & Fungsi EPROM

EPROM (Erasable Programable Read Only Memory) adalah sebuah chip yang bisa diprogram dan mampu menyimpan data. Chip yang isinya data (program-program) yang disikan user ke EPROM tersebut. Satu-satunya cara untuk menghapus data pada EPROM adalah dengan menggunakan sinar ultraviolet.

EPROM merupakan jembatan/penengah antara hadware dan software, sebagai contoh EPROM salah satunya adalah chip BIOS (Basic Input Output System) pada motherboard komputer.

2.2 Type, Ukuran/Kapasitas EPROM

EPROM yang beredar/yang ada dipasaran bermacam-macam, mulai dari type dan ukuran. Dibawah ini adalah beberapa type dan ukuran dari EPROM :

2.2.1 NM27C020 2,097,152 – Bit (256K x 16) UV Erasable CMOS EPROM

Chip EPROM ini mempunyai kapasitas sebesar 2 MB. Menggunakan AMG arsitektur, chip EPROM ini menghantarkan proses dengan kecepatan tinggi dengan mengkonsumsi tegangan rendah.

Chip EPROM ini menyediakan sistem dasar mikroprosesor yang luas kapasitasnya untuk partisi yang besar pada operasi sistem dan aplikasi software. Chip ini memberikan/mempunyai waktu akses

sebesar 100 ns (no-wait-state operasi) dengan CPU yang mempunyai kemampuan tinggi.

Chip NM27C020 menawarkan single chip solusi untuk tempat penyimpanan kode yang dibututhkan dengan 100% firmware – base equipment.

➲ Ciri-ciri :

- CMOS berkemampuan tinggi ; 90 ns waktu akses.
- Kesesuaian mikroprosesor untuk cepat mati (Vpp & PGM adalah "Don't care" selama pembacaan pada operasi normal).
- Setara/sesuai dengan EPROM 27210 & 27C210.
- Dibuat dengan identifikasi kode.
- Pengkodean yang cepat.
- JEDEC standar konfigurasi pin :
  - 40 – pin CDIP paket
  - 40 – pin PDIP paket
  - 40 – pin PLCC paket

2.2.2 NM27C040 4,194,364 – Bit (512 x 8) High Perpormance CMOS EPROM

Chip EPROM ini mempunyai kemampuan yang tinggi, terorganisir atas 512 K dari 8 bit bagian. Chip ini kompabilitas dengan byte-wide-JEDEC EPROM memungkinkan untuk di"upgrade" sampai 8 Mbit.

Ciri-ciri/keistimewaan "Don't care" pada Vpp selama pembacaan operasi mengijinkan perluasan/penambahan memori dari 1 M samapai 8 Mbit tanpa papan sirkuit diganti.

Chip NM27C040 ini menyediakan sistem dasar mikroprosesor yang luas kapasitasnya untuk partisi yang besar pada operasi sistem dan aplikasi software. Chip ini memberikan/mempunyai waktu akses sebesar 120 ns dengan CPU yang mempunyai kemampuan tinggi.

Chip NM27C040 menawarkan single chip solusi untuk tempat penyimpanan kode yang dibututhkan dengan 100% firmware – base equipment. Penggunaan software secara rutin pengaksesannya sangat cepat dari penyimpanan EPROM, kehebatanya meningkatkan sistem utility.

➲ Ciri-ciri :

- CMOS berkemampuan tinggi ; 120 ns, 150 ns waktu akses.
- Kemudahan upgrade path ; Vpp adalah "Don't care" selama pembacaan operasi normal.
- Dibuat dengan identifikasi kode.
- Konfigurasi pin JEDEC standard :
  - 32 – pin PDIP
  - 32 – pin PLCC
  - 32 – pin CERDIP

2.2.3 NM27C128 131,072 – Bit (16 K x 8) High Perpormance CMOS EPROM

Chip ini mempunyai kemampuan yang tinggi 128 K UV EPROM. CMOS terbaru dengan pembagian teknologi pembagian gerbang EPROM yang mana mengijinkan pembagian gerbang EPROM untuk

dioperasikan secepat mungkin, secepat 90 ns waktu akses untuk jarak pengoperasian penuh.

Chip NM27C128 ini menyediakan sistem dasar mikroprosesor yang luas kapasitasnya untuk partisi yang besar pada operasi sistem dan aplikasi software. Chip ini memberikan/mempunyai waktu akses sebesar 90 ns dengan CPU yang mempunyai kemampuan tinggi.

Chip NM27C128 menawarkan single chip solusi untuk tempat penyimpanan kode yang dibututhkan dengan 100% firmware – base equipment. Penggunaan software secara rutin pengaksesannya sangat cepat dari penyimpanan EPROM, kehebatanya meningkatkan sistem utility.

Chip NM27C128 dibuat dengan standar pinout EPROM menyediakan peng"upgrade"dan path yang mudah untuk sistem yang mana menggunakan arus standar EPROM.

Chip NM27C128 merupakan salah satu bagian dari EPROM Family dengan kepadatan yang tinggi yang mana jarak dalam kepadatannya samapai 4 Mb.

➲ Ciri-ciri :

- CMOS berkemampuan tinggi ; 90 ns waktu akses.
- Kesesuaian Mikroprosesor untuk cepat mati.
- Konfigurasi pin JEDEC standard :
  - 32 – pin PDIP paket
  - 32 – pin chip carrier
  - 32 – pin CERDIP paket.
- Penggantian clock untuk 27C128 atau 27128

- Waktu pemrograman 40% lebih cepat dengan turbo algorithm Fairchild.

2.2.4 NM27C210 1,048,576 – Bit (64 K x 16) High Perpormance CMOS EPROM.

Chip ini mempunyai kemampuan yang tinggi. Chip ini memuat 1,048,576 bit terorganisir atas 64 K x 16 bit. Chip ini menawarkan kedua versi erasable untuk pembentukan dasar dan pembuatan awal aplikasi selama non erasable, dipaket dalam versi plastik yang ideal untuk volume tinggi dan aplikasi assembly otomatis.

Chip NMC128 dioperasikan dari 5 volt +100% supply dari mode baca. Chip NMC128 ditawarkan dalam dua paket DIP dan surface mount. Paket DIP adalah 40-pin dual-in-line keramik dengan window kwarsa untuk mengijinkan penghapusan. Paket surpace mount adalah 44-pin PLCC yang ditawarkan dalam OTP.

➲ Ciri-ciri :

- CMOS berkemampuan tinggi ; 90 ns waktu akses.
- Kesesuaian Mikroprosesor untuk cepat mati.
- Kemudahan upgrade path ; Vpp adalah "Don't care" selama pembacaan operasi normal.
- Setara/sesuai dengan EPROM 27210 & 27C210.
- Dibuat dengan identifikasi kode.
- Kecepatan pemrograman
- Konfigurasi pin JEDEC standard :
  - 40 – pin CDIP paket

- 40 – pin PDIP paket
- 44 – pin PLCC paket

Semakin besar ukuran EPROM harganya juga semakin mahal tergantung kapasitas pada EPROM tersebut.

2.3 Bahasa Pemrograman Yang Digunakan

Dibawah ini adalah bahasa pemrograman yang biasa digunakan untuk membuat program/data kedalam chip EPROM :

2.3.1. Bahasa Assembly

Bahasa Assembly merupakan bahasa pemrograman tingkat rendah. Bahasa Assembly mempunyai keunggulan yang tidak mungkin diikuti oleh bahasa tingkat apapun dalam hal kecepatan, ukuran file yang kecil serta kemudahan dalam manipulasi sistem komputer. Dibawah ini adalah langkah-langkah/cara untuk memulai dengan bahasa Assembly :

1. Text Editor

   Untuk menuliskan source file bagi program assembly kita dapat gunkan berbagai editor, misalkan SideKick, Wordstar dan WordPerfect. Source file yang diketikan harus berupa file ASCII, file ini dapat kita hasilkan melalui Wordstar dengan file 'NON DOCUMENT', atau dengan SideKick. Source file untuk assembly umumnya berektensi .ASM.

2. Compiler

Source file ASCII yang telah kita ketikan perlu di-compile kebentuk file objek dengan ektensi .OBJ, dari file objek inilah nantinya dapat dijadikan kebentuk file .EXE atau .COM. Untuk meng-compile source file, misalnya file COBA.ASM menjadi file object dengan ektensi .OBJ dapat kita gunakan file TASM.EXE (dengan mengetikan : C:\>tasm coba).

3. Linking

Fille object yang telah terbentuk dengan TASM, belum dapat dieksekusi secara langsung. Untuk membuat file object kebentuk file yang dapat dieksekusi (ektensi .Com atau .EXE), kita dapat menggunakan file TLINK.EXE. Bile source program yang kita buat dalam bentuk .EXE (maka untuk membentuk file dengan ektensi EXE dapat kita ketikan C:\tlink coba). Bila source program yang dibuat adalah file COM (dapat kita ketikan c:\tlink /t coba).

4. Perbedaan Program COM dan EXE

Program COM :
- ➲ Relatif lebih kecil dari file EXE
- ➲ Lebih cepat dibanding file EXE
- ➲ Hanya dapat menggunakan 1 segmen
- ➲ Ukuran file maksimum 64 KB (ukuran satu segmen)
- ➲ Sulit untuk mengakses data atau procedure yang terletak pada segmen yang lain.
- ➲ 100h byte pertama merupakan PSP (Program Segment Prefix) dari program tersebut.
- ➲ Dapat dibuat dengan DEBUG.

Program EXE :
- Relatif lebih besar dari file COM.
- Lebih lambat disbanding file COM.
- Bisa menggunakan lebih dari 1 segmen.
- Ukuran file tidak terbatas sesuai dengan ukuran memory.
- Mudah mengakses data atau procedure pada segmen yang lain.
- Tidak dapat dibuat dengan DEBUG.

5. Bentuk Angka

Assembler mengizinkan penggunaan beberapa penggunaan angka, yaitu :

1. Desimal

   Untuk menuliskan angka dalam bentuk desimal, dapat digunakan tanda 'D' pada akhir angka tersebut atau dapat juga tidak diberi tanda sama sekali, contoh:298D atau 298 saja.

2. Biner

   Untuk menuliskan angka dalam bentuk biner (0..1), harus ditambahkan tanda 'B' pada akhir angka tersebut, contoh: 0110011B.

3. Hexadesimal

   Untuk menuliskan angka dalam bentuk hexadesimal (0..9,A..F), harus ditambahkan tanda 'H' pada akhir angka tersebut. Perlu diperhatikan bahwa bila angka pertama dari hexa berupa karakter (A..F) maka angka nol harus ditambahkan didepannya. Bila hal ini tidak dilakukan,

assembler tidak akan menganggapnya sebagai suatu label, bukannya sebagai nilai nilai hexa. Contoh penulisan yang benar: 0A12H, 2A02H.

4. Karakter

   Penulisan karakter atau string diapit oleh tanda petik dua (") atau tanda petik satu ('). Contoh: 'Ini adalah karakter'.

6. Label

   Label dapat kita definisikan dengan ketentuan akhir dari nama label tersebut harus dapat berupa tanda titik dua (:). Pemberian nama label dapat digunakan :

   - ❖ Huruf : A..Z (Huruf besar dan kecil tidak dibedakan)
   - ❖ Angka : 0..9
   - ❖ Karakter khusus : @ . _ $

   Nama pada label tidak boleh terdapat spasi dan didahului oleh angka. Contoh dari penulisan label yang benar : mulai: MOV CX,7. Nama label yang terpanjang dapat kita kenali oleh assembler adalah nama label yang terdiri dari 31 karakter.

7. Komentar

   Untuk memberikan komentar pada source file, digunakan tanda ';'. Apapun yang dituliskan dibelakang tanda ':' akan dianggap sebagai komentar. Contoh : MOV BX,7 **; berikan nilai7 pada BX.**

8. Perintah MOV

   Perintah MOV digunakan untuk meng-copy nilai atau angka menuju suatu register, variable atau memori. Adapun syarat untuk perintah MOV ini adalah :

   MOV Tujuan,Asal

Sebagai contohnya :

```
MOV AL,9   ; masukan nilai 9 pada AL
MOV AH,AL  ; nilai AL=9 dan AH=9
MOV AX,9   ; AX=AH+AL hingga AH=0 dan
             AL:=9
```

Perintah MOV akan meng-copy-kan nilai pada sumber untuk dimasukan ke Tujuan, nilai sumber tidaklah berubah. Inilah sebabnya MOV(E) akan kita terjemahkan disini dengan mengkopi, dan bukannya memindahkan.

9. Perintah INT

Di dalam pemrograman assembly, kita akan banyak sekali menggunakan interupsi untuk membantu kita dalam mengerjakan suatu pekerjaan. Untuk menghasilkan suatu interupsi digunakan perintah INT dengan syntax : INT NoInt. Dengan NoInt adalah nomor interupsi yang ingin dihasilkan. Sebagai contohnya bila kita ingin menghasilkan interupsi 21h, dapat dituliskan dengan : **INT 21h**, maka interupsi 21h akan segera terjadi.

2.3.2. Bahasa C++

Bahasa C++ merupakan pengembangan dari Bahasa C dan merupakan bahasa tingkat menengah. Bahasa C++, suatu bahasa pemrograman yang berorientasi pada objek (*Object Oriented Programing- OPP*). OPP memungkinkan pembuatan program menjadi lebih mudah dan cepat untuk menciptakan software yang berkemampuan tinggi. Fasiilitas penting yang ada pada C++ adalah pembentukan class yang akan mampu melakukan operasi-operasi pemrograman *enkapsulasi* (perlindungan atas data dan fungsi class),

inheritance (pewarisan data dan fungsi class), dan *polymorphism* (menggunakan satu nama fungsi untuk beberapa tujuan operasi program).

Pemrograman berorientasi objek adalah suatu teknik yang memungkinkan kita untuk menulis program secara modul dan terstruktur sehingga memudahkan untuk melakukan pemeliharaan dan perubahan.

Program yang dibuat dengan bahasa C++ dapat memiliki beberapa fungsi tetapi setidak-tidaknya memiliki fungsi utam, yaitu main () dimana suatu program akan dimulai dan berakhir disini.

Di dalam bahasa pemrograman lainya penulisan/pembuatan program harus terstruktur dan terperinci sesuai dengan aturan bahasa pemrograman yang kita gunakan. Dibawah ini adalah aturan/cara pendefinisian pembuatan program pada Bahasa C++ :

1. Tipe Data Terdefinisikan (Predefined Data Type)

   Informasi atau data yang kita gunakan dalam suatu program kita kategorikan berdasarkan tipe datanya. Tipe data dari suatu variable akan memberitahu kita jenis data yang bagaimana yang dikandung dalam variable tersebut, dan juga cukupan nilai yang dapat disimpan oleh variable bertipe data tertentu tersebut. Singkatnya, untuk membuat program dengan efisiensi dan efektif kita perlu mengetahui tipe data yang akan kita proses. Tipe-tipe data tersebut dituliskan sebagai berikut dengan urutan singkatanya diikuti kepanjangannya (yang untuk selanjutnya jika kita menyebut tipe data kita hanya perlu dengan menyebut nama singkatannya) : bool

(Boolean), int (integer), char (charakter), float (single-precision floating point numbers), double (double-precision floating), void (valueless data).

2. Memberi Nama Item-Item Pemrograman

   Aturan pemberian nama item pemrograman adalah sebagai berikut :

   - Huruf atau karakter pertama haruslah berupa abjad atau garis bawah/under score (_).
   - Karakter-karakter berikutnya dapat berupa abjad, angka, atau garis bawah tanpa disertai spasi.
   - Secara default suatu nama memiliki panjang paling banyak 32 karakter.
   - Nama-nama adalah bersifat *case sensitive* artinya membedakan huruf besar & kecil.
   - Nama-nama tidak boleh sama dengan nama-nama yang sudah digunakan dalam item-item pemrograman yang didefinisikan (reserved word) dalam C++.

3. Preprocessor Directive

   Maksud diciptakannya *preprocessor directive* adalah untuk membuat suatu file perantara untuk kemudian dikompilasi sebelum kompiler mengkompilasi program yang kita buat. Pendeklarasian dimulai dengan tanda # diikuti dengan nama preprocessor directive dan biasanya ditempatkan sebelum baris-baris program yang kita buat. Berikut ini adalah contoh nama preprocessor directive :

Nama : #include

Deklarasi : #include<file_name>
#include "file_name"

Contoh : #Include<stdio.h>
#include "global.h"

4. Mendeklarasikan Variabel

Variabel adalah suatu nama identifikasi yang digunakan untuk menampung dan mendapatkan informasi. Mendeklarasikan variable adalah membuat suatu pernyataan program tentang tipe data yang dikandung oleh variable serta memberi nama variable tersebut. Cara pendeklarasian variable adalah sebagai berikut :

- tipedata namavariabel;
- tipedata namavariable = nilai awal;
- tipedata namavariable1, namavariable2,.....;

5. Mendeklarasikan Konstanta

Konstanta adalah suatu nama identifikasi (*identifier*) yang diasosiasikan dengan nilai-nilai tetap, artinya bahwa setiap kali kita ingin memanggil/menggunakan nilai tersebut maka kita cukup memanggil nama konstantanya saja yang lebih deskriptif.

Ada dua cara pendeklarasian konstanta : pendeklarasian formal dan pendeklarasian dengan makro. Pendeklarasian formal didahului dengan kata (key word) **const** diikuti tipe data, nama konstanta dan nilai tetapnya sebagai berikut :

const tipedata namakonstanta = nilaitetap;

Pendeklarasian dengan makro dimaksudkan untuk mendefinisikan nilai tetap konstanta dengan preprocessor directive #define.

2.3.3. Turbo Pascal

Turbo Pascal merupakan bahasa pemrograman tingkat tinggi (High Level Language) dan sangat baik untuk dipelajari karena program kompiler yang mempunyai sifat struktur dan prosedural. Pascal adalah bahasa tingkat tinggi yang orientasinya kepada segala tujuan, bahasa ini dirancang oleh Prof. Niklaus Wirth dari Techincal University di Zurich, Switzerland. Turbo Pascal merupakan versi bahasa Pascal yang paling populer dan paling banyak digunakan. Compiler Turbo Pascal banyak digemari karena bersifat interaktif. Bahasa Pascal merupakan bahasa pemrograman propesional juga merupakan sarana yang ideal untuk mengembangkan program-program pascal.

Paket Turbo Pascal meliputi beberapa fasilitas pemrograman yang sangat berkaitan :

- Perpustakaan unit yang menyediakan sejumlah besar prosedur (prosedure) dan fungsi (function).
- Compiler yang menghasilkan program bahasa mesin yang cepat dari kode pascal.
- Debuger yang memungkinkan untuk menelusuri penampilan program baris demi baris, meneliti perubahan nilai variable dan ekspresi sewaktu penampilan berlangsung.

- Fasilitas lingkungan pengembangan dalam bentuk menu yang memadukan elemen dalam memperlancar proses penyuntingan, pengujian dan penyempurnaan program pascal.
- Dukungan grafis dan animasi yang sangat propesional.
- Versi bahasa Pascal yang cepat dan interakif.

Program Pascal adalah program yang terstruktur, ada aturan khusus dalam penulisan program pascal, secara umum struktur program pascal adalah sebagai berikut :

```
Program Struktur_Pascal ; {nama program}
Uses                      {deklarasi peranti}
Label                     {deklarasi peranti}
Const                     {deklarasi konstanta}
Type                      {deklarasi type data}
Var                       {deklarasi variable}

Procedure satu;
 Begin
     {Pernyataan/statemen}
 end;
Function satu;
 Begin
     {Pernyataan/statemen}
   end;
Begin  {awal program utama}
 {pernyataan/statemen-statemen program}
 end;   {akhir program}
```

Dalam Pascal semua variable yang akan dipakai tipe datanya harus sudah ditentukan. Tipe data dalam Turbo Pascal dapat digambarkan sebagai berikut :

1. Tipe Sederhana

   Tipe sederhana maupun tipe data skalar yang bias diartikan bahwa dalam sebuah peubah hanya dimungkinkan untuk menyimpan sebuah nilai data. Yang termasuk kedalam tipe data sederhana adalah :

   1.1. Tipe Ordinal

      Tipe ordinal adalah subset dari tipe sederhana. Dibawah ini adalah yang termasuk kedalam tipe ordinal :

      1. Tipe Integer

         Tipe data yang nilainya tidak mempunyai titik desimal. Titik desimal yang termasuk kedalam kelompok ini yaitu : shortint, integer, longint & word. Masing-masing tipe mempunyai batas nilai [2].

      2. Tipe Boolean

         Data tipe Boolean mempunyai dua nilai true & false.

      3. Tipe Char

         Tipe char digunakan untuk mendefinisikan tipe data yang nilainya merupakan himpunan karakter yang dikenal komputer.

      4. Tipe Terbilang

         Disebut tipe terbilang karena semua nilai disebut satu persatu.

5. Tipe Data Subjangkauan

   Penggunaan tipe subjangkauan terutama untuk meningkatkan kemudahan pemahaman program dan untuk memperoleh cek jangkauan secara otomatis.

1.2. Tipe Real

Tipe real adalah bilangan-bilangan yang berisi titik desimal, mempunyai jangkauan dari 1E – 38 sampai dengan 1E + 38 . E menunjukan niali sepuluh dari pangkat.

2. Tipe String

Data yang berisi sederetan karakter yang banyaknya bias berubah-rubah yaitu dari 1 sampai 255 karakter.

3. Tipe Larik

Larik atau array adalah tipe terstruktur yang mempunyai komponen dalam jumlah yang tetap dan setiap komponen mempunyai tipe data yang sama. Persis masing-masing komponen dalam larik dinyatakan sebagai nomor indeks.

4. Tipe Rekaman

Sama halnya dengan larik, tipe rekaman (record) adalah kumpulan data yang membedakan antara larik dan rekaman adalah bahwa larik sama elemennya harus bertipe sama, tetapi dalam rekaman setiap elemen bias mempunyai tipe data yang berbeda satu sama lain.

5. Tipe Himpunan

Himpunan (set) adalah himpunan objek yang mempunyai tipe data yang sama dan urutan penulisannya tidak diperhatikan.

6. Tipe Pointer

   Tipe suatu variable yang berisi alamat dimemori dimana suatu data yang disimpan bukannya berisi data itu sendiri, dengan kata lain pointer akan menunjukan letak dari data memori.

7. Tipe Berkas

   Berkas/file adalah kumpulan sejumlah komponen yang bertipe data sama, yang jumlahnya tidak tentu dan biasanya tersimpan dalam satu media penyimpanan luas.

2.3.4. Bahasa Power BASIC (BASIC)

Bahasa BASIC (*Beginner's All Purpose Symbolic Instructions Code*) merupakan bahasa tingkat tinggi. Bahasa ini diciptakan oleh Thomas E. Kurtz dan Jhon G. Kemeny, seorang Profesor dari Darthmouth College di Hanover, New Hampshire, New England pada tahun 1964.

Bahasa Power BASIC merupakan sebuah software pemrograman yang menggunakan compiler sebagai penerjemah. Bahasa Power BASIC mempunyai beberapa keunggulan dibandingkan dengan Bahasa BASIC lainnya, antara lain :

- Power BASIC memiliki kemampuan akses Port, Interrupt, dan memory yang lebih mudah dan efisien dibandingkan dengan Bahasa BASIC lain.
- Menyediakan fasilitas pemrograman Array yang lebih lengkap (instruksi ARRAY).

- Dapat menghasilkan *file executable* yang lebih kecil dibandingkan dengan Bahasa BASIC lainnya seperti Quic BASIC.
- File-file yang diperlukan untuk bekerja lebih sedikit dan lebih kecil. Bahkan untuk melakukan kompilasi, Power BASIC tidak memerlukan file Bantu lainya;jadi Power basic merupakan Bahasa Compiler yang murni.
- Memori yang dibutuhkan lebih kecil dibandingkan dengan Quic BASIC sebagai saingan utamanya.
- Power BASIC mempunyai instruksi-instruksi *meta command* yang dapat dilakukan untuk mengatur kompilasi *source* programnya.

1. Menulis Program Sederhana

   a. Metoda Penulisan Program

   - Metode Program dengan Nomor Baris

     Pada metode ini setiap penulisan baris program diawali dengan menuliskan nomor baris instruksi.

     Contoh :

```
10 CLS
20 PRINT "Selamat Datang"
30 RINT "Di Power BASIC"
40 END
```

   - Metoda Program tanpa Nomor Baris

     Pada metoda ini setiap penulisan baris program tanpa diawali dengan menuliskan nomor baris instruksi. Contoh :

```
CLS
PRINT "Selamat Datang"
RINT "Di Power BASIC"
END
```

Cara penulisan yang pertama merupakan cara penulisan penulian Bahasa Basic Konvensional (Interperter), sedangkan pemrograman dengan Power BASIC menggunakan metode yang kedua karena lebih hemat dalam penyimpanannya di memori eksternal.

b. Dokumentasi Program

Pemrograman dapat menulis dokumentasi program dengan instruksi **REM** atau tanda Apostrof ('). Instruksi tersebut berfungsi untuk memberikan catatan atau keterangan-keterangan yang dianggap penting oleh pemrogram. Keterangan tersebut tidak akan dieksekusi seperti instruksi – instruksi yang lain.

Contoh :

```
REM   ************************
REM   *  LUAS SEGITIGA  *
REM   ************************
REM   Budi Sutedjo
REM   Membersihkan Layar
CLS
REM   Menginputkan Alas dan Tinggi
INPUT "Ketikanlah Alas          : ", Alas
INPUT "Ketikanlah Tinggi        :", Tinggi
REM   Menghitung Luas
LET   Luas = Alas * Tinggi / 2
REM   Menampilkan Luas
PRINT "Luas =  '; Luas
END
```

Dalam menuliskan program di atas dapat dilakukan dengan dua mode pengetikan, yaitu mode *overwrite* dan mode *insert*. Untuk memilih mode tersebut, dapat digunakan tombol **Insert** atau **Ctrl V.**

2. Baris program

   Suatu baris program pada Power BASIC terdiri dari maksimal **248** karakter. Untuk baris-baris instruksi yang panjangnya melebihi 248 karakter sebaiknya dipisah-pisah menjadi beberapa baris yang saling berhubungan. Nomor baris bisa sembarang bilangan positif dari 1 – 65535, dan bias berupa karakter alphanumeric yang sering disebut sebagai label, yang merupakan kombinasi dari huruf A – Z, a –z, dan angka 0 – 9 yang ditandai dengan tanda (:).

3. Instruksi

   Instruksi merupakan unit terkecil yang merupakan implementasi sederhana dalam bahasa pemrograman dari suatu proses dasar komputer. Penulisan instruksi ini tidak membedakan antara huruf kecil dan huruf kecil. Instruksi tersebut hanya terdiri karakter alphabet (pada Power BASIC terdapat lebih dari **130** instruksi)

4. Data, Konstanta dan Karakter

   - **Data Numerik**, yaitu jenis data yang digunakan dalam proses aritmatika atau proses matematis lainnya.
   - **Data String**, yaitu jenis data yang dapat terdiri dari berbagai macam karakter apapun. Digunakan untuk proses yang non-matematis.
   - **Data Logika**, yaitu jenis data yang hanya terdiri dari dua satuan, yaitu Benar (true) atau Salah (false). Digunakan

dalam suatu proses logika yang terdiri dari persamaan boolean.

5. Variabel

Variabel (pengubah) adalah suatu lambing dari suatu daerah di memori utama sebuah komputer yang dapat berisi suatu nilai. Variabel merupakan nama yang mewakili nilai data dimana data tersebut dapat berubah pada saat program dieksekusi.

Dalam pemrograman dengan Bahasa BASIC didepan penulisan variable diberikan simbol untuk tiap jenis data yang diwakilinya. Macam-macam simbol yang terdapat dalam Power BASIC adalah :

$	itring/untai	#	Double Precision
%	Integer	##	Extended Precision
&	long integer	@	BCD Fixed point
&&	Quad Integer	@@	BCD Floating point
!	Single Precision	$$	Flex String

6. Operator

Operator adalah simbol-simbol khusus yang digunakan untuk mengoperasikan suatu nilai data (operand). Ada beberapa jenis operator, yaitu :

a. Operator Aritmatika, digunakan untuk mengoperasikan data-data numeric, seperti penjumlah, pengurangan, perkalian, pembagian dan lainnya.

Valensi	Operator	Fungsi
2	^	Pemangkatan
1	*	Perkalian
1	/	Pembagian
1	\	Pembagian integer
1	MOD	Modulo (sisa pembagian)
0	+	Penjumlahan
0	-	Pengurangan atau negasi

b. Operator Relasi digunakan untuk mewakili sebuah nilai logika (nilai Boolean), dari suatu persamaan atau nilai.

Operator-operator yang terlibat adalah :

= : Sama dengan
> : Lebih dari
< : Kurang dari
<> : Tidak sama dengan
>= : Lebih atau sama dengan
<= : Kurang atau sama dengan

c. Operasi Logika, digunakan untuk mengoperasikan operand (konstanta, variable, atau suatu ekspresi logis) secara logis. Operator-operator logika yang umum dalam bahasa pemrograman :

AND : Operasi AND akan menghasilkan benar hanya jika kedua buah operatornya bernilai benar.

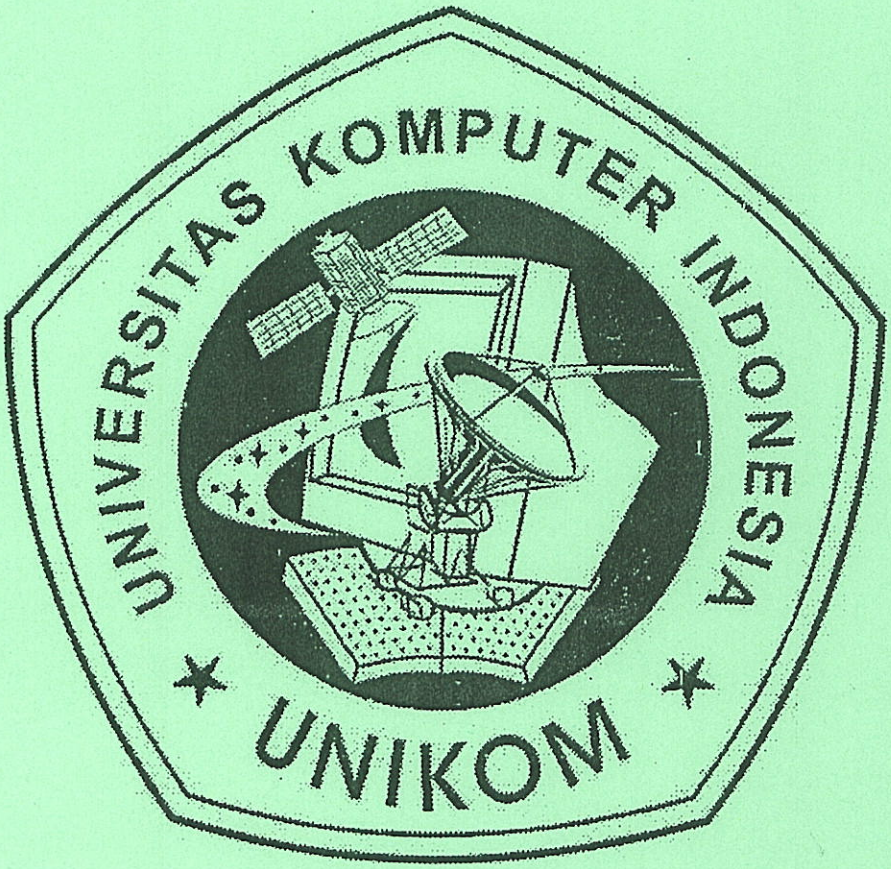
OR : Operasi OR akan menghasilkan nilai benarjika salah satu operandnya bernilai benar.

XOR : Operasi XOR akan bernilai benar jika kedua operandnya berbeda (satunya benar dan satunya salah)

EQV : Operasi EQV akan bernilai benar jika kedua operandnya bernilai sama.

IMP : Operasi Implikasi akan bernilai benar jika dan hanya jika operand keduanya bernilai benar dan operand keduanya bernilai salah.

NOT : Operasi ini hanya akan membalikan nilai dari sebuah nilai dari sebuah operand dan sifatnya unary.


UNIVERSITAS KOMPUTER INDONESIA
UNIKOM