Spesifikasi:

Ukuran: 14x21 cm
Tebal: 218 hlm
Harga: Rp 30.800
Terbit pertama: Juni 2004
Sinopsis singkat:

Dengan hadirnya buku ini, diharapkan pembaca yang ingin mendalami teknik interfacing komputer dan mikrokontroler memperoleh informasi yang lengkap dan praktis sebagai panduan utama untuk percobaan-percobaan. Buku ini juga dilengkapi berbagai contoh program menggunakan Bahasa C/C++ hingga Visual Basic .NET. Disertakan juga rangkaian lengkap sehingga memperkaya khazanah pembaca dengan berbagai rangkaian terbaru dan canggih. Buku ini sangat berguna bagi mahasiswa yang mengambil mata kuliah Mikroprosesor, Interfacing Komputer, dan Elektronika Digital atau bagi masyarakat umum yang tertarik untuk mempelajari mikrokontroler AT89C51.

BAB 5
BAHASA C/C++ UNTUK INTERFACING KOMPUTER

5.1 Pengenalan Bahasa C/C++

Bahasa C dikembangkan di Laboratorium Bell (USA) sekitar tahun 1972 oleh Dennis Ritchie yang adalah seorang pakar pemrograman. Bahasa C++ yang merupakan penyempurnaan dan pengembangan dari C dibuat oleh Bjarne Stroustrup. C dan C++ ialah *compiler* untuk membuat aplikasi yang umum dan juga merupakan bahasa tingkat menengah yang sering digunakan untuk membuat aplikasi interfacing komputer maupun mikrokontroler. Kedua bahasa ini dikategorikan sebagai bahasa tingkat menengah karena terkadang kita perlu mengetahui juga Bahasa Assembly untuk pemrograman yang berhubungan dengan perangkat keras atau peralatan komputer. Kita dapat menyisipkan Bahasa Assembly ke C/C++ untuk keperluan pemrograman perangkat keras komputer menggunakan perintah **asm**.

Untuk mempelajari C/C++, Anda dapat menggunakan perangkat lunak Turbo C, Turbo C++, Borland C++, C++ Builder atau Visual C++, sedangkan Visual C++ .NET sudah memiliki banyak perbedaan dari segi tampilan GUI dan beberapa model pemrogramannya yang dikenal dengan istilah *managed C++*. Akan tetapi, membuat aplikasi

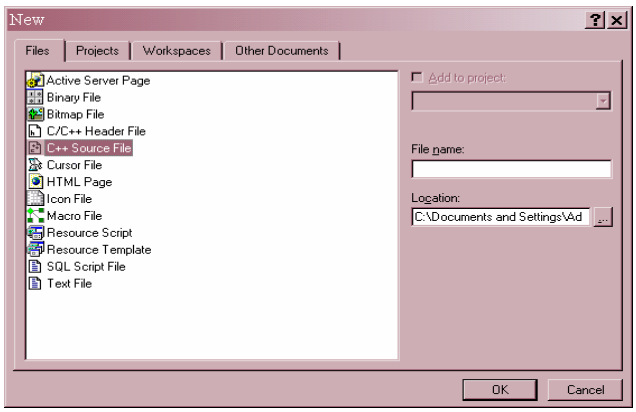
interfacing menggunakan Visual C++ .NET maupun Visual C# .NET tetap saja merupakan sesuatu yang menarik dan merupakan tantangan. Turbo C++ ialah kompiler yang berbasiskan DOS, sedangkan Borland C++ versi 4 ke atas berjalan di bawah Windows. Anda juga dapat membuat program C++ berbasiskan Linux dan dikompilasi menggunakan *gcc*.

5.2 Memulai Pemrograman C/C++

Anda dapat men-download kompiler Turbo C++ beserta contoh program yang digunakan dalam buku ini pada alamat www.widodo.com/downloads/downloads.html. Instal program Turbo C++ yang telah Anda download, lalu set direktori sesuai dengan tempat penyimpanan file Anda. Jika Anda menggunakan Visual C++ 6, tidak ada seting yang harus Anda lakukan. Berikut contoh kode program sederhana yang penulis buat menggunakan C dan C++ sebagai perbandingan. Tampilan berikut adalah jika Anda ingin membuat program di Visual C++ 6.

Langkah–langkah menjalankan program Visual C++ 6:

1. Klik menu **Start** | **Program** | **Microsoft Visual C++ 6**.
2. Klik menu **File** | **New** dan isi kotak **File Name** dengan *hello* seperti tampak pada Gambar 5.1 di bawah ini.



Gambar 5.1 Membuat File C/C++

3. Klik button **OK** dan akan tampil editor tempat Anda mengetikkan kode. Untuk menjalankan program, tekan **F5** atau gunakan menu **Run**.

5.2.1 Konstruksi Program C/C++

Konstruksi dari suatu program C/C++ harus mengikuti aturan sebagai berikut:

```
/* Ini berupa komentar sebagai penjelasan dari program yang
dibuat. Bisa berisi tujuan program, dibuat oleh siapa dan
tanggal berapa */

Include file
Deklarasi variabel global
fungsi Utama(main)
{
   Isi program
}
```

Berikut contoh program untuk menerima input dan menampilkannya ke layar monitor menggunakan bahasa C:

```
/* Program C untuk menerima input data dari user lalu
   menampilkannya.
   Dibuat oleh Widodo Budiharto
   Tanggal 20 Februari 2004 */
// Digunakan untuk library fungsi yang akan digunakan, dikenal
   sebagai preprocessor //directive
#include <stdio.h>
#include <conio.h>

int main()//fungsi utama bernama main bertipe int
{
//deklarasi variable integer dan karakter
   int umur;
   char nama[30];

//Menampilkan pesan menggunakan fungsi printf
printf ("Selamat menggunakan bahasa C untuk Interfacing
Komputer\n");
   printf ("Masukkan nama Anda \n");
//menerima data menggunakan fungsi scanf, %s untuk string, %c
hanya untuk 1 karakter scanf ("%s",&nama);
   printf ("Masukkan umur Anda \n");
//menerima data integer menggunakan %d scanf ("%d", &umur);
   printf ("Dear %s, umur Anda %d \n",nama,umur);
   printf ("Tekan sembarang tombol untuk keluar !");
//menunggu aksi penekanan sembarang tombol untuk keluar
//fungsi getch() ini menggunakan library conio.h
   getch();
//mengembalikan nilai integer karena fungsi main bertipe integer
   return 0;
}
```

Untuk menjalankan program tersebut menggunakan Visual C++ 6, Anda dapat menekan **F5**, atau menjalankan menu **Run**. Jika menggunakan Turbo C++, tekan menu **Alt+R**. Program di atas membuat dua buah variabel bernama *nama* yang bertipe char yang dapat menampung sekitar 30 karakter, dan variabel *umur* yang dapat menampung bilangan bulat. Untuk menerima input, digunakan fungsi *scanf* dengan dua parameter, yaitu tipe data penerima dan nama variabel penampung. Fungsi *printf* digunakan untuk menampilkan string dan data yang ingin ditampilkan dengan menggunakan variabel yang ada. Fungsi *getch()* akan mendeteksi penekanan sembarang input dari user untuk keluar. Gambar 5.2 menampilkan hasil output program.

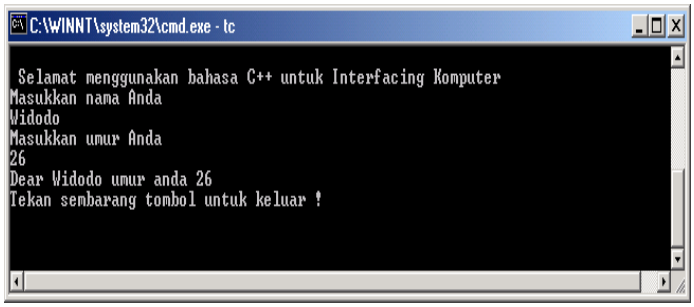
Gambar 5.2 Hasil Program C

Jika Anda menggunakan C++, struktur tubuh programnya tidak jauh berbeda dari struktur berikut.

```
/* Program C++ untuk menerima input dari user lalu menampilkan
   datanya
   Dibuat oleh Widodo Budiharto
   Tanggal 20 Februari 2004 */

#include <iostream.h>
#include <conio.h>

int main()
{

int umur;
char nama[30];
cout<<"\n Selamat menggunakan bahasa C++ untuk Interfacing
Komputer"<<endl;
cout<<"Masukkan nama Anda"<<endl;
cin>>nama;
cout<<"Masukkan umur Anda"<<endl;
cin>>umur;
cout<<"Dear "<<nama<<" umur Anda "<<umur<<endl;
```

```
cout<<"Tekan sembarang tombol untuk keluar !"<<endl;
getch();
return 0;
}
```

Simpan program ini dan beri nama ***hello.cpp.*** Pada program C++ di atas, C++ menggunakan header file yang umum, yaitu *iostream.h* untuk menerima dan menampilkan data, berbeda dengan C yang menggunakan header file *stdio.h*. Untuk menerima data digunakan fungsi *cin* dan untuk menampilkan data digunakan fungsi *cout*. Fungsi *\n* digunakan untuk mengganti baris, begitu juga fungsi *endl*. Beberapa fungsi lain yang cukup terkenal adalah *\t* untuk tabulasi dan *\b* untuk bell. Jika program dijalankan, tampilan dengan Turbo C++ akan terlihat seperti gambar berikut.



Gambar 5.3 Tampilan Program menggunakan C++

Jika Anda membuatnya melalui editor di Linux, kompilasi dapat dilakukan melalui terminal menggunakan *gcc* sebagai berikut:

```
gcc -o hello hello.cpp
```

Kode di atas akan menghasilkan output *hello* dari file sumber bernama *hello.cpp*. Anda cukup mengetikkan *./hello* saja dan program akan tampil.

5.2.2 Pengulangan *(Looping)*

Untuk melakukan pengulangan eksekusi kode pada program, dapat digunakan fungsi *for* dengan sintaks sebagai berikut:

```
for (int a=0;a<10;a++)
{
```

```
printf (“Ini pengulangan ke: %d”, a);
}
```

Kode di atas akan menampilkan pesan sebanyak sepuluh kali. Selain menggunakan *for*, dapat digunakan juga fungsi *while*. Program akan mendeteksi apakah kondisi memenuhi kondisi *true* dan jika memenuhinya maka kode di dalam blok eksekusi akan dijalankan.

Contoh:

```
char answer='y'
while (answer ='y')
{
…
printf (“Ingin melanjutkan (y/n) ?”);
scanf (“%d”, &answer);
}
```

5.3 Fungsi

Jika kita menulis keseluruhan kode di dalam fungsi *main*, kita akan sangat repot dalam proses pelacakan kesalahan atau pembacaan program. Oleh karena itu idealnya kode-kode untuk tujuan-tujuan tertentu dimasukkan ke dalam fungsi yang terpisah-pisah. Fungsi mempunyai tipe data tersendiri dan dapat/tidak mengembalikan harga, bergantung tipe datanya. Fungsi yang diharapkan mampu mengembalikan harga dari hasil proses dalam fungsi tersebut harus menggunakan tipe data selain void. Jika diinginkan agar fungsi tidak mengembalikan data, dapat digunakan tipe data void. Berikut ini contoh kode yang menggunakan fungsi bertipe data void dan integer.

```
//Program contoh menulis Fungi
//Dibuat oleh Mr. Widodo
#include <stdio.h>
#include <conio.h>
#include <dos.h>

void bunyi(); //deklarasi fungsi
int hitung(int,int); //deklarasi fungsi
int main()
{
int a,b,hasil;
clrscr();
bunyi();//memanggil fungsi bunyi
printf ("Masukkan bilangan 1");
scanf("%d" , &a);
printf ("Masukkan bilangan 2");
scanf ("%d", &b);
```

```
hasil=hitung(a,b);//assignment , hasil perhitungan dari fungsi
hitung dikirim ke hasil
bunyi();
printf ("\nHasil perkalian ialah %d" , hasil);
getch();
}
//fungsi bunyi tipe datanya void, oleh karena itu tidak
mengembalikan harga
void bunyi() //definisi fungsi bunyi
{
sound(5000); //menghasilkan bunyi 5000 Hz
delay(1000);// selama 1 detik
nosound(); //matikan bunyi
}
//fungsi htiung tipe datanya integer, oleh karena itu
mengembalikan harga
int hitung(int x, int y)//definisi fungsi bunyi dengan 2
parameter/argumen
{

int hasil_perkalian=x*y; //mengembalikan hasil perkalian dengan
tipe integer
}
```

Program di atas akan membersihkan layar menggunakan fungsi *clrscr()*, setelah itu meminta user memasukkan data. Program juga memanggil fungsi dan membunyikan suara serta melakukan perhitungan, lalu mengembalikan lagi hasil perhitungan tersebut ke pemanggil *(caller)*.

5.4 Pemrograman Grafik

Turbo C++ menggunakan BGI (Borland Graphics Interface) untuk menampilkan gambar berwarna. Beberapa file pendukung grafik yang penting ialah:

- ***Graphics.h***, berisi deklarasi variable-variable penting grafik
- ****.bgi***, berisi data pengendali kartu grafik. Pada komputer yang memakai mode grafik VGA, pengendali grafiknya ialah *egavga.bgi*.
- ****.chr***, berisi data huruf (font) misalnya *gothic.chr* yang berisi data huruf model gothic.

Program berikut merupakan contoh pemrograman grafik Turbo C++ (dari folder *examples*) yang akan menerima data temperatur dari

user, lalu dapat disimpan ke disk serta dibaca dan ditampilkan dalam bentuk grafik bar.

```
/*Program membaca temperatur lalu menampilkan Bar
Dimodifikasi oleh Widodo Budiharto
Tanggal 21 Februari 2004 */

//Program membaca temperatur lalu menampilkan Bar
#include <conio.h>
#include <ctype.h>
#include <stdio.h>
#include <stdlib.h>
#include <graphics.h>

/* Prototipe / deklarasi fungsi */
void get_temps(void);
void table_view(void);
void min_max(int num_vals, int vals[], int *min_val, int
   max_val);
float avg_temp(int num_vals, int vals[]);
void graph_view(void);
void save_temps(void);
void read_temps(void);

/* definisi konstanta global*/
#define TRUE      1
#define READINGS  8
/* struktur data Global */
int temps[READINGS];
int main(void)
{
   while (TRUE)
   {
   printf("\nMenu Program Plotting Temperatur \n");
   printf("\tE - Masukkan data temperatur \n");
   printf("\tS - Simpan data ke disk\n");
   printf("\tR - Baca file disk \n");
   printf("\tT - Melihat data dalam format tabel \n");
   printf("\tG - Tampilan grafik \n");
   printf("\tX - Keluar dari program \n");
   printf("\nTekan salah satu kunci di atas: ");

   switch (toupper(getche()))
   {
   case 'E': get_temps(); break;
   case 'S': save_temps(); break;
   case 'R': read_temps(); break;
   case 'T': table_view(); break;
   case 'G': graph_view(); break;
   case 'X': exit(0);
      }
   }
}

/* Definisi Fungsi */
void get_temps(void)
{
   char inbuf[130];
   int reading;
   printf("\nMasukkan temperatur-temperatur\n");
```

```
   for (reading = 0; reading < READINGS; reading++)
   {
   printf("\nMasukkan bacaan ke # %d: ", reading + 1);
   gets(inbuf);
   sscanf(inbuf, "%d", &temps[reading]);
   }
}
void table_view(void)
{
   int reading, min, max;
   clrscr(); /* Bersihkan layar */
   printf("Membaca\t\tTemperatur(F)\n");
   for (reading = 0; reading < READINGS; reading++)
      printf("%d\t\t\t%d\n", reading + 1, temps[reading]);

   min_max(READINGS, temps, &min, &max);
   printf("Temperatur minimum: %d\n", min);
   printf("Temperatur maksimum: %d\n", max);
   printf("Temperatur rata-rata: %f\n", avg_temp(READINGS,
      temps));
}
void min_max(int num_vals, int vals[], int *min_val, int
*max_val)
{
   int reading;
   *min_val = *max_val = vals[0];
   for (reading = 1; reading < num_vals; reading++)
   {
      if (vals[reading] < *min_val)
         *min_val = vals[reading];
      else if (vals[reading] > *max_val)
         *max_val = vals[reading];
   }
}
float avg_temp(int num_vals, int vals[])
{
   int reading, total = 0;
   for (reading = 0; reading < num_vals; reading++)
      total += vals[reading];
      return (float) total/reading; //casting
}
void graph_view(void)
{
   int graphdriver = DETECT, graphmode;
   int reading, value;
   int maxx, maxy, left, top, right, bottom, width;
   int base; /* sumbu x */
   int vscale = 1.5; /* nilai skala vertikal */
   int space = 10; /* spasi antara bar */
   char fprint[20]; /* teks terformat untuk sprintf */
   initgraph(&graphdriver, &graphmode, "..\\bgi");//direktori bgi
   if (graphresult() < 0) /* Pastikan inisialisasi driver ok
   */
   return;
   maxx = getmaxx(); /* daerah terjauh */
   width = maxx /(READINGS + 1); /* skala dan spasi */
   maxy = getmaxy() - 100; /* ruang untuk teks */
   left = 25;
   right = width;
   base = maxy / 2; /* untuk nilai negatif */
```

```
   for (reading = 0; reading < READINGS; reading++)
   {
      value = temps[reading] * vscale;
      if (value > 0)
      {
         top = base - value; /* layar atas */
         bottom = base;
         setfillstyle(HATCH_FILL, 1);
      }
      else
      {
         top = base;
         bottom = base - value; /* layar bagian bawah */
         setfillstyle(WIDE_DOT_FILL, 2);
      }
      bar(left, top, right, bottom);
      left +=(width + space); /* spasi kiri untuk bar
         berikutnya */
      right +=(width + space); /* ujung kanan untuk bar
         berikutnya*/
   }
   outtextxy(0, base, "0 -");
   outtextxy(10, maxy + 20, "Plot dari Pembacaan temperatur ");
   for (reading = 0; reading < READINGS; reading++)
   {
      sprintf(fprint, "%d", temps[reading]);
      outtextxy((reading *(width + space)) + 25, maxy + 40,
         fprint);
   }
   outtextxy(50, maxy+80, "Tekan sembarang tombol");
   getch(); /* menunggu penekanan key*/
   closegraph();
}
void save_temps(void)
{
   FILE * outfile;
   char file_name[40];
   printf("\nSimpan sebagai nama file apa ? ");
   while (kbhit()); /* jika ditekan*/
   gets(file_name);
   if ((outfile = fopen(file_name,"wb")) == NULL)
   {
      perror("\nMembuka file gagal ! ");
      return;
   }
   fwrite(temps, sizeof(int), READINGS, outfile);
   fclose(outfile);
}
void read_temps(void)
{
   FILE * infile;
   char file_name[40] = "test";
   printf("\nBaca dari file mana? ");
   while (kbhit());
   gets(file_name);
   if ((infile = fopen(file_name,"rb")) == NULL)
   {
      perror("\nBuka file gagal! ");
      return;
   }
```

```
   fread(temps, sizeof(int), READINGS, infile);
   fclose(infile);
}
```

Perhatikan bahwa pada program di atas, fungsi terlebih dahulu harus dideklarasikan dengan menuliskan nama fungsi dan dapat diikuti dengan tipe data parameter yang diinginkan. Setelah fungsi dideklarasikan, definisikanlah fungsi. Anda juga dapat langsung mendefinisikan fungsi asalkan fungi-fungsi tersebut ditulis di atas fungsi *main*.

Untuk deteksi dan inisialisasi driver grafik, digunakan fungsi yang ada di BGI (*Borland Graphics Interface*)sebagai berikut.

```
void graph_view(void)
{
 int graphdriver = DETECT, graphmode;
….
 initgraph(&graphdriver, &graphmode, "..\\bgi");//direktori bgi
```

Beberapa fungsi yang berhubungan dengan pembacaan dan penulisan file pada disk ialah *fopen*, *fwrite*, *fread,* dan *fclose*.

```
void save_temps(void)
{
 FILE * outfile;
 char file_name[40];
 printf("\nSimpan sebagai nama file apa ? ");
 while (kbhit()); /* jika ditekan*/
 gets(file_name);
 if ((outfile = fopen(file_name,"wb")) == NULL)
 {
 perror("\nMembuka file gagal ! ");
 return;
 }
 fwrite(temps, sizeof(int), READINGS, outfile);
 fclose(outfile);
}
```

Program di atas akan mencoba menyimpan suatu file dengan meminta nama file tertentu dari user. Fungsi *fwrite* digunakan untuk menulis ke dalam disk, lalu ditutup dengan perintah *fclose*.

```
void read_temps(void)
{
 FILE * infile;
 char file_name[40] = "test";
 printf("\nBaca dari file mana? ");
 while (kbhit());
 gets(file_name);
 if ((infile = fopen(file_name,"rb")) == NULL)
 {
 perror("\nBuka file gagal! ");
 return;
```

```
 }
 fread(temps, sizeof(int), READINGS, infile);
 fclose(infile);
}
```

Program di atas akan mencoba membuka file menggunakan perintah *fopen* dan jika file dibuka dengan sukses, file akan dibaca menggunakan perintah *fread*.

Latihan:

1. Buatlah program kalkulator, lengkap dengan fungsi trigonometrinya. Jangan lupa menggunakan header file *math.h*.
2. Buat program untuk menampilkan lingkaran bewarna dengan diameter 300 pixel.
3. Jalankan contoh program *bgidemo.c* yang menampilkan contoh pemrograman warna dan games.
4. Pelajari pointer, template dan pemrograman OOP *(Object Oriented Programming*). Anda dapat melihat materi pemrograman OOP C++ di www.widodo.com/tutorial/ti.html.