BAB IV

DESKRIPSI PROGRAM

4.1. STRUKTUR PROGRAM

Untuk mempermudah pengembangan, perbaikan, dan kompilasi, program aplikasi musikal yang dibuat dipecah menjadi 3 bagian, yaitu Program Utama.C, Program Song.C, dan program Tuts.C.

- Program UTAMA.C adalah program yang berisi fungsi-fungsi untuk membuat menu pilhan, daftar perintah, dan daftar informasi program.
- Program SONG.C adalah program yang berisi fungsi-fungsi untuk membuat lagu, menyimpan data, mengubah data, dan memainkan lagu
- Program TUTS.C berisi fungsi-fungsi untuk menggambar piano dan memvisualisasikan gerankannya pada saat lagu dimainkan.

Untuk menghasilkan program UTAMA.EXE, program UTAMA.C dikompilasi bersama-sama dengan program SONG.C dan TUTS.C. Di dalam bahasa C, untuk mengkompilasi lebih dari 1 program secara bersamaan diperlukan suatu penghubung antara file-file program yang akan dikompilasi, yaitu file header (*header file*). File Header adalah file yang berisi nama-nama fungsi yang akan dipakai bersama serta parameter-parameter yang dikirimkan.

Header file untuk SONG.C adalah SONG.H, header file untuk TUTS.C adalah TUTS.H, sedangkan Program UTAMA.C tidak mempunyai header file karena merupakan program induk. UTAMA.C justru punya file proyek (*project file*) yaitu file yang berisi daftar program anak yang akan dipanggil.

4.1.1. Listing file UTAMA.PRJ

```
Utama(song.h)
song(tuts.h)
tuts
```

Dari file UTAMA.PRJ ini, program UTAMA.C sebagai file induk akan memanggil fungsi yang berada pada program SONG.C atau TUTS.C, demikian juga program SONG.C akan memanggil fungsi yang berada pada program TUTS.C.

4.1.2. Listing file header SONG.C

```
void MainLagu();
void ProsesLagu(int tempo);
void InfoLagu(int tempo_info);
void BuatLagu();
void EditLagu();
int BacaLagu(char *kode);
char Baca(int X,int Y, int Color);
void BacaNada(char Kata[80]);
void Cursor(int X, int Y, int Status, int Color);
int MuatLagu();
void CetakLagu(int awaly,int akhiry);
void FrameLagu();
```

4.1.3. Listing file TUTS.C

```
void tsatu(int kiri, int atas);
void lsatu(int kiri, int atas);
void tdual(int kiri, int atas);
void ldual(int kiri, int atas);
void tduar(int kiri, int atas);
void lduar(int kiri, int atas);
void ttiga(int kiri, int atas);
void ltiga(int kiri, int atas);
void InisialGraph();
void GambarPiano();
void BorderInfo(int warna_grs,int warna_win,int pola);
void BorderPiano();
void ClearBawah();
void ClearPiano();
void ClearKiri();
void Massage (char *Tanya);
void Massage1 (char *Tanya);
void FrameMenu(int color_kop,int pola_kop,int bolor_kop,
               int color_win,int pola_win,int bolor_win,
               int color_judul,char *judul, int posx,int posy);
```

4.2. DEKRIPSI FUNGSI-FUNGSI POKOK PROGRAM

4.2.1. Fungsi InisialGraph()

Fungsi ini digunakan untuk membuka sistem grafik. Karena program yang dibuat menggunakan sistem grafik maka fungsi InisialGraph() merupakan fungsi yang paling vital untuk keseluruhan program.

Listing fungsi InisialGraph()

```
void InisialGraph()
{
   int ModeGraph,DriverGraph = DETECT,KodeSalah;

   initgraph(&DriverGraph, &ModeGraph," ");
   KodeSalah = graphresult();
   if (KodeSalah != grOk)
   {
            cprintf("\n\rKode Grafik tdk dpt diaktifkan ...!",
            grapherrormsg(KodeSalah));
            cprintf("\n\rKesalahan yang terjadi: %s",grapherrormsg(KodeSalah));
            getch();
            exit(1);
   }
}
```

Dengan memberikan DETECT kepada pengendali grafik, maka *initgraph()* akan melakukan pemilihan terhadap pengendali grafik dan menentukan mode grafik dengan resolusi tertinggi pada pengendali grafik yang diaktifkan. *Graphresult()* akan mengembalikan suatu kode nilai jika terjadi suatu kesalahan pada operasi grafik yang dijalankan, sedangkan *grapherrormsg()* dipakai untuk mendapatkan pesan kesalahan dalam bahasa Inggris berdasarkan kode kesalahan yang dikirimkan.

4.2.2. Fungsi MenuUtama()

Fungsi ini digunakan untuk membuat menu pilihan dan tampilan informasi-informasi program.

Listing fungsi MenuUtama()

```
void MenuUtama()
{
  union inkey
            {
     char ch[2];
     int i;
            }c;
  int awaly = 0, akhiry = 11,Files;
  int gantiawy=akhiry,gantiaky=0;

  Files=MuatLagu();
  FrameLagu();
  CetakLagu(awaly,akhiry);
  for(;;)
  {
   Files=MuatLagu();
   FrameLagu();
   Daftar_Perintah();
   c.i = bioskey(0);
   if(c.ch[0])
   {
    switch(c.ch[0])
    {
    case 27 :
      closegraph();
      exit(1);
      break;
    }
   }
   else
   {
    switch(c.ch[1])
      {
          case 59 :          /* tekan F1 */
            InfoUtama();
            break;
          case 60 :          /* tekan F2 */
            BuatLagu();
            break;
          case 61 :          /* tekan F3 */
              EditLagu();
              break;
            case 62 :                  /* tekan F4 */
            MainLagu();
            break;
          case 72 :if(awaly<=0)
              {
                awaly=0;
                akhiry=11;
                gantiawy=akhiry;
              }
```

```
                else
                {
                  awaly=awaly-11;
                  akhiry=akhiry-(gantiaky-gantiawy);
                  gantiawy=akhiry;
                }
                break;
          case 80 :if(((Files-1)-akhiry)<12)
                {
                  awaly=gantiawy;
                  akhiry=Files-1;
                  gantiaky=akhiry;
                gantiawy=awaly;
                }
                else
                {
                  awaly=awaly+11;
                  akhiry=awaly+11;
                  gantiaky=akhiry;
                  gantiawy=awaly;
                  break;
                }
        }
      setviewport(1,342,385,479,1);
      setcolor(3);
      setfillstyle(0,0);
      bar(0,0,384,136);
      rectangle(1,1,384,136);
     CetakLagu(awaly,akhiry);
   }
  }
 }
```

Pada saat fungsi *MenuUtama()* dijalankan, akan ditampilkan daftar informasi dari semua lagu yang sudah dibuat melalui pemanggilan fungsi-fungsi *MuatLagu(), FrameLagu(), dan CetakLagu().* Fungsi *MuatLagu()* adalah fungsi yang menyediakan tempat untuk penyimpanan semua data musikal yang dibuat. Fungsi *FrameLagu()* digunakan untuk membuat bingkai jendela informasi lagu. Sedangkan fungsi *CetakLagu()* digunakan untuk menampilkan seluruh data yang dibuat, dengan masing-masing halaman (layar) memuat informasi dari 12 lagu.

Selain menampilkan informasi lagu, fungsi *MenuUtama()* juga menampilkan menu program yang bisa dipilih beserta keterangan tombol yang harus digunakan. Jika pemakai ingin melihat informasi program, maka harus menekan tombol *F1* dimana

fungsi *MenuUtama()* akan memanggil fungsi *InfoUtama()*. Jika ingin membuat lagu baru, pemakai harus menekan tombol *F2*, dimana fungsi *MenuUtama()* akan memanggil fungsi *BuatLagu()*. Untuk mengubah data yang sudah ada, pemakai harus menekan *F3* dimana fungsi *MenuUtama()* akan memanggil fungsi *EditLagu()*. Begitupun untuk memainkan lagu, pemakai harus menekan tombol *F4* dimana fungsi *MenuUtama()* akan memanggil fungsi *MainLagu()*. Sedangkan tombol *Esc* digunakan untuk keluar dari program.

4.2.3. Fungsi InfoUtama()

Fungsi *InfoUtama()* digunakan untuk menampilkan informasi program mengenai menu pilihan yang ada dan cara penggunaan program.

Listing fungsi InfoUtama()

```
void InfoUtama()
{
  union inkey  {
                 char ch[2];
                 int i;
               }c;
  char *Key[]={"PgUp","PgDwn","PgUp/PgDwn"};
  char *Run;
  KopInfo(300,5,7,Key[1]);
  setcolor(14); settextstyle(0, 0, 0); Info1();
  do
   {
      c.i=bioskey(0);
      if(c.ch[1])
         {
            switch(c.ch[1])
            {
               case 73 : KopInfo(300,5,7,Key[2]);          /* PgUp */
                         setcolor(14);  Info1();
                         break;
               case 81  :  KopInfo(310,5,7,Key[0]);        /* PgDn */
                           setcolor(14); Info2();
                           break;
            }
         }
   }
   while(c.ch[0]!=27);                                     /* Esc */
}
```

Fungsi *InfoUtama()* pertama-tama akan menampilkan informasi program pada halaman pertama. Untuk ke halaman berikutnya, tombol yang digunakan adalah *pagedown* sedangkan tombol untuk menuju halaman sebelumnya adalah *pageup*.

4.2.4. Fungsi BuatLagu()

Fungsi ini digunakan untuk membuat suatu file lagu baru yang berisi judul lagu, tempo, dan data dari masing-masing not lagu, serta nama file dengan ekstensi DAT.

Listing fungsi BuatLagu()

```
void BuatLagu()
{
 static char String1[40];
 static char String2[20];
 static char String3[20];
 char String4[1000][10];
 int i=0;
 char sukses='0';
 char pilihan;

 FrameMenu(8,1,7,3,1,11,6,"MEMBUAT DATA LAGU BARU",100,5);
 fflush(stdin);
 settextstyle(2,0,5); setcolor(14);
 outtextxy(15,30,"Masukkan Judul Lagu : ");
 if(get_str(String1,190,30,3)!=NULL)
  {
    strcpy(info_lagu.judul,String1);
  }
 setcolor(6); outtextxy(15,50,"Masukkan Tempo Lagu : ");
 if(get_str(String2,190,50,3)!=NULL)
 {
   info_lagu.tempo=atoi(String2);
 }
 while (sukses=='0')
 {
          FrameMenu(8,1,7,3,1,11,6,"MEMBUAT DATA LAGU BARU",100,5);
          settextstyle(2,0,5); setcolor(6);
          outtextxy(15,70,"Masukkan Nama File : ");
          if(get_str(String3,190,70,3)!=NULL)
          {
             strcpy(NamaFile,String3);
          }
          if ((f_lagu=fopen(NamaFile,"rb"))!=NULL)
          {
             Massage("File ada dalam diskl Ditumpuk? Y/N");
             pilihan=getche();
             if(pilihan=='Y' || pilihan=='y')
```

```
                {
                    f_lagu=fopen(NamaFile,"w");
                    sukses='1';
                }
                else
                 sukses='0';
            }
            else
            {
                f_lagu=fopen(NamaFile,"w");
                sukses='1';
            }
    }
    fwrite(&info_lagu,sizeof(info_lagu),1,f_lagu);
    do
      {
            FrameMenu(8,1,7,3,1,11,6,"MEMBUAT DATA LAGU BARU",100,5);
            settextstyle(2,0,5);
            setcolor(1);
            outtextxy(15,70,"Masukkan Data, 0,0 Untuk Selesai");
            outtextxy(15,90,"Format (Nada,Durasi)->(XX,XX) : ");
            if(get_str(String4[i],250,90,3)!=NULL)
            {
                    BacaNada(String4[i]);
            i++;
            }
            fflush(stdin);
            fwrite(&info_nada,sizeof(info_nada),1,f_lagu);
            }
            while(info_nada.pitch!=0 || info_nada.dur!=0);
            fclose(f_lagu);
}
```

Di sini disiapkan 4 buah larik berisi string, *String1* digunakan untuk menampung judul lagu, *String2* untuk tempo, *String3* untuk nama file (dengan ekstensi DAT), dan *String4* untuk menampung seluruh data lagu (dalam format *pitch* dan *dur*). Untuk menampung seluruh masukan data, di awal program disiapkan juga 2 buah struktur yaitu info_lagu dan info_nada. Judul dan tempo lagu kemudian dimasukkan ke dalam info_lagu, sedangkan nama file dan data lagu dimasukkan ke dalam info_nada

4.2.5. Fungsi EditLagu()

Fungsi ini digunakan untuk mengubah data dan informasi lagu, baik judul, tempo, nomor nada, ataupun durasi.

Listing fungsi EditLagu()

```
void EditLagu()
#define ESC 27
#define JML_REC 1
{
 unsigned int editan;
 long int posisi;
 int NadaKe,kolom,indeks,jumlah;
 int x=10,y=5,k=0,l=0,m=0;
 char Text[100];
 static char String1[100][30],String2[100][10],String3[100][20];

 BacaLagu("rb+");
 jumlah=record;
 kolom=1;
 ClearPiano();
 BorderPiano();
 settextstyle(2,0,4);
 setcolor(14);
 for(indeks=0;indeks<jumlah-1;indeks++)
 {
  sprintf(Text,"%2d,%.2f",mem_nada[0][indeks].pitch,mem_nada[0][indeks].dur);
  outtextxy(x+20,y,Text);
  if(kolom==13)
      {
     y+=12;
       kolom=1;
       x=10;
      }


    else
      {
       kolom++;
       x+=47;
      }
 }
 do
 {
       InfoLagu(info_lagu.tempo);
       FrameMenu(8,1,7,1,1,11,14,"E D I T  L A G U",150,5);
       setcolor(11);
       settextstyle(2,0,5);
       outtextxy(15,40,"Edit [1] Judul, [2] Tempo,[3] Nada, ESC Menu");
       editan=getch();
       switch (editan)
        {
```

```
                case '1':
                        fflush(stdin);
                        posisi=0*sizeof(info_lagu);
                        fseek(f_lagu,posisi,SEEK_SET);
                        fread(&info_lagu,sizeof(info_lagu),JML_REC,f_lagu);
                        sprintf(Text,"Judul Lagu     : %s",info_lagu.judul);
                        outtextxy(15,70,Text);
                        outtextxy(15,90,"Judul Yang Baru : ");
                        if(get_str(String1[i],160,90,1)!=NULL)
                        strcpy(info_lagu.judul,String1[i]);
                        fseek(f_lagu,posisi,SEEK_SET);

                        fwrite(&info_lagu,sizeof(info_lagu),JML_REC,f_lagu);
                        i++;
                        break;
                case '2':
                        posisi=0*sizeof(info_lagu);
                        fseek(f_lagu,posisi,SEEK_SET);
                        fread(&info_lagu,sizeof(info_lagu),JML_REC,f_lagu);
                        sprintf(Text,"Tempo Lagu     : %d",info_lagu.tempo);
                        outtextxy(15,70,Text);
                        outtextxy(15,90,"Tempo Yang Baru : ");
                        if(get_str(String2[k],160,90,1)!=NULL)
                        info_lagu.tempo=atoi(String2[k]);
                        fseek(f_lagu,posisi,SEEK_SET);

                        fwrite(&info_lagu,sizeof(info_lagu),JML_REC,f_lagu);
                        k++;
                        break;

                case '3':
                        outtextxy(15,70,"Nada Keberapa Yang Mau Di Rubah : ");
                        scanf("%d",&NadaKe);
                        posisi=sizeof(info_lagu)+(NadaKe-1)*sizeof(info_nada);
                        fseek(f_lagu,posisi,SEEK_SET);
                        fread(&info_nada,sizeof(info_nada),JML_REC,f_lagu);
                        sprintf(Text,"Nadanya adalah : %d %.2f ",info_nada.pitch,
                        info_nada.dur);
                        outtextxy(15,90,Text);
                        outtextxy(15,110,"Masukkan Nada Baru : ");
                        if(get_str(String3[m],190,110,1)!=NULL)
                        BacaNada(String3[m]);
                        fseek(f_lagu,posisi,SEEK_SET);

                        fwrite(&info_nada,sizeof(info_nada),JML_REC,f_lagu);
                        m++;
                        break;
        }
    }
    while(editan!=ESC);

    fclose(f_lagu);
    GambarPiano();
}
```

Pertama-tama, fungsi *EditLagu()* akan memanggil fungsi *BacaLagu(char *kode)* dengan kode "rb+" yang berarti membaca dan mengubah bagian data nada pada lagu yang akan dipilih. Keseluruhan data nada kemudian ditampilkan dalam baris dan kolom dengan urutan pembacaan dari baris1 kolom1, baris1 kolom2, ... , baris1 kolom13, baris2 kolom1, ... , baris2 kolom13, dan seterusnya sampai keseluruhan data terbaca. Setelah *user* menentukan nomor lagu yang akan diubah, akan ditampilkan menu dengan pilihan [1] untuk mengganti judul, [2] untuk mengubah tempo, [3] untuk mengubah data nada (pitch atau dur) dan tonbol *Esc* untuk kembali ke *MenuUtama()*.

4.2.6. Fungsi BacaLagu(char *kode)

Fungsi ini digunakan untuk membuka dan membaca data lagu yang dipilih *user*.

Listing fungsi BacaLagu()

```
int BacaLagu(char *kode)
#define JML_REC 1
{
 char pilihan;
 int NoLagu=0;
 static char String[5];

   fflush(stdin);
   Massage1("Pilih Lagu No : ");
   if(get_str(String,200,20,4)!=NULL)
   NoLagu=atoi(String);
   strcpy(NamaFile,List[NoLagu-1]);
   f_lagu=fopen(List[NoLagu-1],kode);
   record=0;
   fread(&info_lagu,sizeof(info_lagu),1,f_lagu);
   while (fread(&info_nada,sizeof(info_nada),JML_REC,f_lagu) == JML_REC)
         {
          mem_nada[0][record].pitch=info_nada.pitch;
          mem_nada[0][record].dur=info_nada.dur;
          record=record+1;
         }
}
```

Setelah *user* memilih nomor lagu, file yang dimaksudkan (berdasarkan nomor lagu) akan dibuka dan kemudian data dari setiap record dibaca (*pitch* dan *dur*-nya).

4.2.7. Fungsi MainLagu()

Fungsi *MainLagu()* digunakan untuk memainkan lagu sekaligus mengubah tempo lagu dan nada dasarnya. Tempo lagu ditambah jika lagu dirasa terlalu cepat atau sebaliknya dikurangi jika terlalu lambat. Nada dasar ditambah jika nadanya dirasa terlalu rendah dan sebaliknya, nada dasar dikurangi jika dirasa terlalu tinggi.

Listing fungsi MainLagu()

```
void MainLagu()
{
        char pilihan;
        static char String[10],String1[10];

        BacaLagu("rb");
        Massage1("Rubah Tempo ? Y/N ");
        pilihan=getch();
        if(pilihan=='Y' || pilihan=='y')
        {
                Massage1("Masukkan Tempo : ");
                if(get_str(String,90,20,4)!=NULL)
                tempo=atoi(String);
        }
        else
                tempo=info_lagu.tempo;

        Massage1("Rubah Nada Dasar ? Y/N ");
        pilihan=getch();
        if(pilihan=='Y' || pilihan=='y')
        {
                Massage1("Masukkan Perubahan : ");
                if(get_str(String1,90,20,4)!=NULL)
                nada_dasar=atoi(String1);
        }
        else
                nada_dasar=0;
        InfoLagu(tempo);
        setfillstyle(1,9);
        bar(10,100,239,130);
        setcolor(14);
        outtextxy(15,110,"To abort, press any key...!");
        ProsesLagu(tempo);
        nosound();
}
```

Setiap akan memainkan lagu, *user* dihadapkan dengan kotak dialog apakah ingin mengubah tempo (atau nada dasar) atau tidak. Setelah itu, barulah fungsi *MainLagu()* akan memanggil fungsi *ProsesLagu()*.

4.2.8. Fungsi ProsesLagu()

Fungsi ini berguna untuk memainkan lagu sekaligus memvisualkan gerakan tombol piano dengan memanggil fungsi-fungsi pada program TUTS.C

Listing fungsi ProsesLagu()

```
void ProsesLagu (int tempo)
{
        float t;
        int i, n, v=0,nada;
        int kiri,atas,p,q,x,y,r,s;
        ClearKiri();
        gambar();
        setactivepage(0);
        setvisualpage(0);
        setfillstyle(1,8);
        setviewport(1,325,385,479,1);
        clearviewport();
        setcolor(11);
        setlinestyle(0,0,2);
        rectangle(1,1,384,153);
        x=0;
        y=0;
        p=100;
        q=80;
        r=300;
        s=20;
        for (n = 0; (n<record-1)&&(!kbhit()); n++)
        {
          setviewport(1,325,385,479,1);
          putimage(x,y, simbol, XOR_PUT);
          putimage(p,q, simbol, XOR_PUT);
          putimage(r,s, simbol, XOR_PUT);
          t = mem_nada[v][n].dur;
                nada = mem_nada[v][n].pitch;
                nada=nada+nada_dasar;

          if ( nada ==0)                    /* nomor nada = 0)
            {
                kiri = 20; atas = 50;
                tduar(kiri, atas);
                sound(27.5*pow(2.,nada/12.));
                delay(tempo*t);
                lduar(kiri, atas);
            }
```

```
.....                                   /* dan seterusnya sampai nomor nada = 86 /*
  else                                  /* nomor nada ke-87 */
    {
        kiri = 397; atas = 230;
        ttiga(kiri, atas);
        sound(27.5*pow(2.,nada/12.));
        delay(tempo*t);
        ltiga(kiri, atas);
    };
  setviewport(1,325,385,479,1);
  putimage(x,y, simbol, XOR_PUT);
  putimage(p,q, simbol, XOR_PUT);
  putimage(r,s, simbol, XOR_PUT);
  p=random(100);
  q=random(20);
  x=random(310);
  y=random(80);
  r=random(210);
  s=random(50);
 }
}
```

Dalam fungsi *ProsesLagu()* dilakukan looping untuk membaca semua data nada dan membangkitkan suara dengan perintah *sound(frek)* dimana *frek* adalah frekuensi yang dihasilkan dengan rumus pada teknik transkripsi langsung. sesuai tinggi rendahnya nomor nada. Selain itu akan divisualisasikan gerakan piano dengan memanggil fungsi-fungsi *tsatu(), lsatu(), tduar(), lduar(), tdual(), ldual(), ttiga(), dan ltiga().*

Fungsi *tsatu()* dan *lsatu()* adalah fungsi-fungsi untuk menekan dan melepas tuts piano putih yang tertutup 2 tuts hitam, yaitu pada tuts-tuts dengan nomor nada 5, 10, 12, 17, 22, 24, 29, 34, 36, 41, 46, 48, 53, 58, 60, 65, 70, 72, 77, 82, dan 84. Fungsi *tduar()* dan *lduar()* untuk menekan dan melepas kembali tuts-tuts piano putih yang tertutup 1 tuts hitam di sebelah kanan, yaitu tuts-tuts dengan nomor nada 0, 3, 8, 15, 20, 27, 32, 39, 44, 51,56, 63, 68, 75, dan 80. Fungsi-fungsi *tdual()* dan *ldual()* digunakan pada tuts-tuts piano putih yang tertutup 1 tuts hitam di sebelah kiri, yaitu tuts-tuts dengan nomor nada 2, 7, 14, 19, 26, 31, 38, 43, 50, 55, 62, 67, 74, 79, dan

86. Fungsi-fungsi *ttiga()* dan *ltiga()* untuk tuts-tuts hitam. Sedangkan lainnya adalah untuk tuts putih yang tidak tertutup tuts hitam (nomor nada 87).

4.2.9. Fungsi GambarPiano()

Fungsi ini adalah fungsi untuk menggambar piano pada kondisi diam (tidak memainkan lagu) .

Listing fungsi GambarPiano()

```
void GambarPiano()
{
 int i;

 Kop();
 BorderPiano();
 setfillstyle(6,6);
 bar(1, 1, 637, 279);

 setfillstyle(1, 7); bar(20, 10, 590, 88);
 setcolor(15); line(21, 10, 589, 10);
 for( i=20; i<590; i+=30 )
 {
   setcolor(8);
   setlinestyle(0, 0, 1);
        line(i+29, 10, i+29, 88);
   line(i+1, 88, i+28, 88);
   setcolor(15);
        line(i+2, 10, i+2, 88);
 }
setfillstyle(1, 1);
bar(37, 10, 63, 55); bar(97, 10, 123, 55); bar(127, 10, 153, 55);
bar(187, 10, 213, 55); bar(217, 10, 243, 55); bar(247, 10, 273, 55);
bar(307, 10, 333, 55); bar(337, 10, 363, 55); bar(397, 10, 423, 55);
bar(427, 10, 453, 55); bar(457, 10, 483, 55); bar(517, 10, 543, 55);
bar(547, 10, 573, 55);
setcolor(8);
rectangle(37, 10, 63, 55); rectangle(97, 10, 123, 55);
rectangle(127, 10, 153, 55);rectangle(187, 10, 213, 55);
rectangle(217, 10, 243, 55); rectangle(247, 10, 273, 55);
rectangle(307, 10, 333, 55); rectangle(337, 10, 363, 55);
rectangle(397, 10, 423, 55);rectangle(427, 10, 453, 55);
rectangle(457, 10, 483, 55); rectangle(517, 10, 543, 55);
rectangle(547, 10, 573, 55);
setcolor(15);
line(37, 10, 63, 10); line (37, 10, 37, 55);line(97, 10, 123, 10);
line (97, 10, 97, 55);line(127, 10, 153, 10); line (127, 10, 127, 55);
line(187, 10, 213, 10); line (187, 10, 187, 55);line(217, 10, 243, 10);
line (217, 10, 217, 55);line(247, 10, 273, 10); line (247, 10, 247, 55);
line(307, 10, 333, 10); line (307, 10, 307, 55);line(337, 10, 363, 10);
line (337, 10, 337, 55);line(397, 10, 423, 10); line (397, 10, 397, 55);
line(427, 10, 453, 10); line (427, 10, 427, 55);line(457, 10, 483, 10);
```

```
line (457, 10, 457, 55);line (517, 10, 543, 10); line (517, 10, 517, 55);
line (547, 10, 573, 10); line (547, 10, 547, 55);

setfillstyle (1, 7);bar(20,100,560,178);setcolor(15);line (21,100,559,100);
for( i=20; i<560; i+=30 )
 {
        setcolor(8);setlinestyle (0, 0, 1);
        line (i+29, 100, i+29, 178);line (i+1, 178, i+28, 178);
        setcolor(15);line (i+2, 100, i+2, 178);
 }
setfillstyle (1, 1);
bar(37, 100, 63, 145); bar(67, 100, 93, 145); bar(97, 100, 123, 145);
bar(157, 100, 183, 145); bar(187, 100, 213, 145); bar(247, 100, 273, 145);
bar(277, 100, 303, 145); bar(307, 100, 333, 145); bar(367, 100, 393, 145);
bar(397, 100, 423, 145); bar(457, 100, 483, 145); bar(487, 100, 513, 145);
bar(517, 100, 543, 145);
setcolor(8);
rectangle (37, 100, 63, 145); rectangle (67, 100, 123, 145);
rectangle (97, 100, 123, 145);rectangle (157, 100, 183, 145);
rectangle (187, 100, 213, 145);rectangle (247, 100, 273, 145);
rectangle (277, 100, 303, 145);rectangle (307, 100, 333, 145);
rectangle (367, 100, 393, 145);rectangle (397, 100, 423, 145);
rectangle (457, 100, 483, 145); rectangle (487, 100, 513, 145);
rectangle (517, 100, 543, 145);
setcolor(15);
line (37, 100, 63, 100); line (37, 100, 37, 145);
line (67, 100, 93, 100); line (67, 100, 67, 145);
line (97, 100, 123, 100); line (97, 100, 97, 145);
line (157, 100, 183, 100); line (157, 100, 157, 145);
line (187, 100, 213, 100); line (187, 100, 187, 145);
line (247, 100, 273, 100); line (247, 100, 247, 145);
line (277, 100, 303, 100); line (277, 100, 277, 145);
line (307, 100, 333, 100); line (307, 100, 307, 145);
line (367, 100, 393, 100); line (367, 100, 367, 145);
line (397, 100, 423, 100); line (397, 100, 397, 145);
line (457, 100, 483, 100); line (457, 100, 457, 145);
line (487, 100, 513, 100); line (487, 100, 487, 145);
line (517, 100, 543, 100); line (517, 100, 517, 145);

setfillstyle (1, 7); bar(20, 190, 470, 268);
setcolor(15); line (21, 190, 469, 190);
for( i=20; i<470; i+=30 )
{
  setcolor(8);
  setlinestyle (0, 0, 1);
        line (i+29, 190, i+29, 268);
        line (i+2, 268, i+28, 268);
  setcolor(15);
        line (i+1, 190, i+1, 268);
}
setfillstyle (1, 1);

bar(37, 190, 63, 235); bar(67, 190, 93, 235);bar(127, 190, 153, 235);
bar(157, 190, 183, 235); bar(187, 190, 213, 235);bar(247, 190, 273, 235);
```

```
bar(277, 190, 303, 235); bar(337, 190, 363, 235);bar(367, 190, 393, 235);
bar(397, 190, 423, 235);
setcolor(15);
line(37, 190, 63, 190); line (37, 190, 37, 235);
line(67, 190, 93, 190); line (67, 190, 67, 235);
line(127, 190, 153, 190); line (127, 190, 127, 235);
line(157, 190, 183, 190); line (157, 190, 157, 235);
line(187, 190, 213, 190); line (187, 190, 187, 235);
line(247, 190, 273, 190); line (247, 190, 247, 235);
line(277, 190, 303, 190); line (277, 190, 277, 235);
line(337, 190, 363, 190); line (337, 190, 337, 235);
line(367, 190, 393, 190); line (367, 190, 367, 235);
line(397, 190, 423, 190); line (397, 190, 397, 235);
}
```

Piano dengan jumlah tuts 88 digambarkan dalam 3 bagian karena keterbatasan layar. Bagian pertama (paling atas) mewakili bagian kiri piano mencakup tuts-tuts dengan nomor 0 sampai 31, bagian kedua mewakili tuts-tuts nomor 32 sampai 62, sedangkan bagian ketiga (paling bawah) mewakili bagian kanan piano yaitu tuts-tuts dengan nomor 63 sampai 87.

4.2.10. Fungsi-fungsi untuk menekan atau melepaskan tuts piano

Fungsi-fungsi berikut menggunakan sistem *viewporting*. Penekanan tuts piano digambarkan dengan memberikan warna yang berbeda pada tuts yang sedang dimainkan, dan pelepasannya digambarkan dengan mengembalikan warna tuts yang ditekan ke warna sebelumnya.

a. Fungsi tsatu() dan lsatu()

Fungsi *tsatu()* digunakan untuk menekan tuts piano putih yang tertutup 2 tuts hitam. Untuk melepaskannya kembali digunakan fungsi *lsatu()*.

Listing fungsi tsatu() dan lsatu()

```
void tsatu(int kiri, int atas)
 {
  setviewport(kiri, atas, kiri + 30, atas + 80, 1);
  setfillstyle(9, 15); bar(1, 1, 29, 79);setcolor(8);
  setlinestyle(0, 0, 1);line(2, 1, 2, 78); line(2, 1, 29, 1);
  setcolor(15); line(29, 2, 29, 78); line(2, 79, 29, 79);
  setviewport(kiri - 13, atas, kiri + 13, atas + 45, 1);
  setfillstyle(1, 1); bar(1, 1, 26, 45);
```

```
  setviewport(kiri + 17, atas, kiri + 43, atas + 45, 1);
  setfillstyle(1, 1); bar(1, 1, 26, 45);
 }

void lsatu(int kiri, int atas)
 {
  setviewport(kiri, atas, kiri + 30, atas + 80, 1);
  setfillstyle(1, 7); bar(1, 1, 29, 79);
  setcolor(15); setlinestyle(0, 0, 1);
  line(2, 1, 2, 78); line(2, 1, 29, 1);
  setcolor(8); line(29, 2, 29, 78); line(2, 79, 29, 79);
  setviewport(kiri - 13, atas, kiri + 13, atas + 45, 1);
  setfillstyle(1, 1); bar(1, 1, 26, 45);
  setviewport(kiri + 17, atas, kiri + 43, atas + 45, 1);
  setfillstyle(1, 1); bar(1, 1, 26, 45);
 }
```

b. Fungsi tdual() dan ldual()

Fungsi-fungsi ini digunakan untuk menekan tuts piano putih yang tertutup 1 tuts hitam di sebelah kiri (*tdual()*) dan kemudian melepaskannya kembali (*ldual()*).

Listing fungsi-fungsi tdual() dan ldual()

```
void tdual(int kiri, int atas)
 {
  setviewport(kiri, atas, kiri + 30, atas + 80, 1);
  setfillstyle(9, 15); bar(1, 1, 29, 79);
  setcolor(8); setlinestyle(0, 0, 1);
  line(2, 1, 2, 78); line(2, 1, 29, 1);
  setcolor(15); line(29, 2, 29, 78); line(2, 79, 29, 79);
  setviewport(kiri - 13, atas, kiri + 13, atas + 45, 1);
  setfillstyle(1, 1); bar(1, 1, 26, 45);
 }

void ldual(int kiri, int atas)
 {
  setviewport(kiri, atas, kiri + 30, atas + 80, 1);
  setfillstyle(1, 7); bar(1, 1, 29, 79);
  setcolor(15); setlinestyle(0, 0, 1);  line(2, 1, 2, 78); line(2, 1, 29, 1);
  setcolor(8); line(29, 2, 29, 78); line(2, 79, 29, 79);
  setviewport(kiri - 13, atas, kiri + 13, atas + 45, 1);
  setfillstyle(1, 1); bar(1, 1, 26, 45);
 }
```

c. Fungsi-fungsi tduar() dan lduar()

Fungsi *tduar()* digunakan untuk menekan tuts piano putih yang tertutup tuts hitam di sebelah kanannya. Untuk melepaskannya digunakan fungsi *lduar()*.

Listing fungsi tduar() dan lduar()

```
void tduar(int kiri, int atas)
 {
  setviewport(kiri, atas, kiri + 30, atas + 80, 1);
  setfillstyle(9, 15); bar(1, 1, 29, 79);
  setcolor(8); setlinestyle(0, 0, 1);
  line(2, 1, 2, 78); line(2, 1, 29, 1);
  setcolor(15); line(29, 2, 29, 78); line(2, 79, 29, 79);
  setviewport(kiri + 17, atas, kiri + 43, atas + 45, 1);
  setfillstyle(1, 1); bar(1, 1, 26, 45);
 }


void lduar(int kiri, int atas)
 {
  setviewport(kiri, atas, kiri + 30, atas + 80, 1);
  setfillstyle(1, 7); bar(1, 1, 29, 79);
  setcolor(15); setlinestyle(0, 0, 1);
  line(2, 1, 2, 78); line(2, 1, 29, 1);
  setcolor(8); line(29, 2, 29, 78); line(2, 79, 29, 79);
  setviewport(kiri + 17, atas, kiri + 43, atas + 45, 1);
  setfillstyle(1, 1); bar(1, 1, 26, 45);
 }
```

d. Fungsi-fungsi ttiga() dan ltiga()

Fungsi *ttiga()* dan *ltiga()* digunakan untuk menekan dan melepaskan tuts-tuts berwarna hitam.

Listing fungsi ttiga() dan ltiga()

```
void ttiga(int kiri, int atas)
 {
  setviewport(kiri, atas, kiri + 26, atas + 45, 1);
  setfillstyle(1, 9); bar(1, 1, 25, 44);
  setcolor(8); setlinestyle(0, 0, 1);
  line(0, 0, 0, 45); line(0, 0, 26, 0);
  setcolor(15); line(26, 0, 26, 45); line(0, 45, 26, 45);
 }

void ltiga(int kiri, int atas)
 {
  setviewport(kiri, atas, kiri + 26, atas + 45, 1);
  setfillstyle(1, 1); bar(1, 1, 25, 44);
  setcolor(15); setlinestyle(0, 0, 1);
  line(0, 0, 0, 45); line(0, 0, 26, 0);
  setcolor(8); line(26, 0, 26, 45); line(0, 45, 26, 45);
 }
```

4.3. CARA MENJALANKAN PROGRAM

Untuk menjalankan program aplikasi teknik penggambaran musik dalam proses composing ini, kita tinggal masuk ke direktori TurboC kemudian mengetikkan *UTAMA*, maka akan langsung muncul salam pembuka program diteruskan dengan Menu Utama. Disini kita bisa memilih salah satu menu, yaitu melihat informasi program (dengan menekan tombol F1), membuat lagu baru (dengan menekan tombol F2), Mengubah data (dengan terlebih dulu menekan tombol F3), atau memainkan lagu (dengan menekan tombol F4). Jika ingin keluar, maka kita tekan tombol Esc.

Untuk pilihan pertama (F1), maka akan ditampilkan halaman pertama dari informasi program yang ditulis dalam 2 halaman. Tekan tombol PgDn untuk melihat ke halaman 2, dan tekan PgUp untuk kembali ke halaman 1.

Untuk pilihan kedua (F2), akan muncul kotak dialog untuk mengisikan judul lagu, tempo, nama file, dan data-data nada dari lagu yang dibuat. Judul diisi dengan mengetikkan judul lagu maksimal 24 karakter (huruf). Tempo diisi dengan suatu bilangan, sebagai perbandingan lihat dulu daftar lagu yang ada. Nama file diisi dengan nama maksimal 8 karakter (tanpa spasi) dan diakhiri dengan ekstensi DAT. Sedangkan data nada diisi dengan sebarang bilangan bulat 1 sampai 87 untuk nomor nada, dan sebarang bilangan riil yang sesuai dengan harga ketukan masing-masing not.

Pilihan ketiga (F3) diambil untuk mengubah data lagu baik judul, tempo, maupun data nada dari masing-masing not. Disini akan ditampilkan menu pilihan [1] untuk mengubah judul, [2] untuk mengubah tempo, [3] untuk mengubah data nada atau durasi, dan Esc untuk kembali ke Menu Utama. Untuk mengubah data pemakai tinggal mengikuti semua dialog yang disediakan.

Pilihan keempat (F4) untuk memainkan lagu yang sudah ada. Disini, pemakai tinggal memilih lagu yang akan dimainkan dan mengubah tempo atau nada dasar lagu (jika diinginkan).