Lampiran 1

```
{* ------------------------------------------------------------ *
 *        Judul Program      : Hashing_Separate_Chaining         *
 *        Dibuat oleh        : Toni Khalimi                      *
 *        N I M              : J 101 92 0727                     *
 *        Kepentingan        : Tugas Akhir                       *
 *        Tanggal pembuatan  : 25 Maret 1998                     *
 *        Revisi akhir       : 1 April 1998                      *
 * ------------------------------------------------------------ *}

program Hashing_Separate_Chaining;

uses crt;

const Uk_Tabel = 11; {* Ukuran Tabel Hash *}
      Lima    = 5;
      Enam    = 6; {* Panjang maksimum karakter kunci *}
      Delapan = 8;
      LimaBls = 15;
      DuaPuluh = 20;
      Bel = ^G;

type Str5  = string[Lima];
     Str6  = string[Enam];
     Str8  = string[Delapan];
     Str15 = string[LimaBls];
     Str20 = string[DuaPuluh];

     {* Tipe data masukan *}
     Simpul = ^Data;
     Data   = record
                Info    : Str6;
                Nama    : Str15;
                Alamat  : Str20;
                Kota    : Str8;
                KodePos : Str5;
                Berikut : Simpul
              end;

     {* Tipe kepala tabel hash *}
     Tabel_Hash = record
                    Berikut  : Simpul
                  end;

     {* Array untuk kepala tabel hash *}
     Li = array[0..Uk_Tabel-1] of Tabel_Hash;

var Kepala             : Li;
    Pilih, Pilihan,
    Ada, Lanjut        : char;
    Nm                 : Str15;
    Alm                : Str20;
    Kota               : Str8;
    KP                 : Str5;
```

```
    Posx, Posy, i      : byte;
    Selesai, Cek_Tabel : boolean;


{* -------------------- Fungsi Hash Division -------------------- *}
function HashDivision(Kunci : Str6) : word;

var k : integer;
    p : word;

begin
  {* Periksa apakah ada spasi *}
  p := length(Kunci);
  while (Kunci[p] = ' ') and (p > 0) do
     p := p - 1;

  k := 0;
  for i := 1 to length(Kunci) do
      k := k + ord(Kunci[i]);

  HashDivision := k mod Uk_Tabel
end;


{* ------------------------------------------------------------ *}
function Ada_Tabel(Kep : Li) : boolean;

var Acak : word;
    Ada  : boolean;

begin  Ada := false;
  Acak := 0;
  repeat
    if Kep[Acak].berikut <> nil then
       Ada := true;
    Acak := Acak + 1;
  until (Acak = Uk_Tabel-1) or Ada;
  Ada_Tabel := Ada
end;


{* ------------------------------------------------------------ *}
procedure Buat_Garis;

begin
  writeln('------------------------------------------------------')
end;


{* ------------------------------------------------------------ *}
procedure Inisialisasi_Tabel_Hash(var Kep : Li);

var i : word;

begin  for i := 0 to Uk_Tabel-1 do
  begin
```

```
        new(Kep[i].berikut);
        Kep[i].Berikut  := nil
      end;
end;


{* ------------------------------------------------------------ *}
procedure Baca_Teks_Tambah_Simpul(var Kep : Li);

var Akhir, Baru : Simpul;
    F           : text;
    Kunci       : Str6;
    Harga_Hash  : word;

begin
   assign(F,'KUNCI.DAT');
   reset(F); {* Buka file *}
   while not eof(F) do
     begin
        {* Baca file teks F *}
        readln(F,Kunci,Nm,Alm,Kota,KP);
        write('Nomor Pengenal : ');
        writeln(Kunci);writeln;
        write(' Nama    : ');
        writeln(Nm);
        write(' Alamat  : ');
        writeln(Alm);
        write(' Kota    : ');
        writeln(Kota);
        write(' Kode Pos : ');
        writeln(KP);
        writeln;

        {* -------------------------
            Menghitung Alamat Hasnya
           ------------------------- *}
        Harga_Hash := HashDivision(Kunci);
        writeln('Alamat Tabel Hashnya : ',Harga_Hash:2);

        readln; {* Berhenti sejenak *}

        {* Buat dan tambah simpul selama pembacaan file teks *}
        new(Baru);
        Baru^.Info := Kunci ;
        Baru^.Nama := Nm;
        Baru^.Alamat := Alm;
        Baru^.Kota := Kota;
        Baru^.KodePos := KP;

        if Kep[Harga_Hash].Berikut = nil then
           Kep[Harga_Hash].Berikut := Baru
        else
           begin
             akhir := Kep[Harga_Hash].Berikut;
             {* Menemukan akhir list *}
```

```
            while (Akhir^.Berikut <> nil ) do
            {* Geser Akhir ke akhir list *}
              Akhir := Akhir^.Berikut;
            {* Sambung simpul *}
            Akhir^.Berikut := Baru;
          end;
        Akhir := Baru;
        Akhir^.berikut := nil
      end; {* while eof *}
   close(F); {* Tutup file teks F *}
end; {* Baca_Teks_Tambah_Simpul *}


{* ------------------------------------------------------------ *}
procedure Baca_Data(var Kunci : Str6; var Nm : Str15;
                    var Alm   : Str20; var Kota  : Str8;
                    var KP    : Str5; var Harga_Hash : word);

begin
  write('Nomor Pengenal : ');
  readln(Kunci);
  write(' Nama   : ');
  readln(Nm);
  write(' Alamat : ');
  readln(Alm);
  write(' Kota   : ');
  readln(Kota);
  write(' Kode Pos : ');
  readln(KP);
  writeln;
  {* -------------------------
      Menghitung Alamat Hashnya
     ------------------------- *}
    Harga_Hash := HashDivision(Kunci);
  writeln('Alamat Tabel Hashnya : ',Harga_Hash:2);
end;


{* ------------------------------------------------------------ *}
{*       Prosedur ini menambah simpul baru dibelakang list      *}
{* ------------------------------------------------------------ *}
procedure Tambah_Simpul (var Kep   : Li;
                             Kunci : Str6;
                             Harga_Hash : word);
const Bel = ^G;

var Baru, Akhir : Simpul;

begin
  new(Baru);
  Baru^.Info := Kunci ;
  Baru^.Nama := Nm;
  Baru^.Alamat := Alm;
  Baru^.Kota := Kota;
  Baru^.KodePos := KP;
  Baru^.Berikut := nil;
```

```
   {* Kepala Tabel Hash masih kosong *}
   if Kep[Harga_Hash].Berikut = nil then
      Kep[Harga_Hash].Berikut := Baru
   else
      begin
        Akhir := Kep[Harga_Hash].Berikut;
        {* Menemukan akhir list *}


        while (Kunci <> Akhir^.Info) and (Akhir^.Berikut <> nil ) do
              Akhir := Akhir^.Berikut;
        {* Cek apakah data sudah ada *}
        if Kunci = Akhir^.Info then  {* Data sudah ada *}
           begin
             writeln;
             writeln(Bel,'No. Pengenal sudah ada dalam tabel.');
           end
        else {* Sambung simpul baru *}
          Akhir^.Berikut := Baru;

      end;
      Akhir := Baru;
      Akhir^.Berikut := nil;
end;


{* ------------------------------------------------------------ *}
procedure Baca_Tambah_Simpul(var Kep : Li);

var Ada                  : char;
    Kunci                : Str6;
    Posx,Posy            : byte;
    Harga_Hash           : word;

begin
  repeat
    Baca_Data(Kunci,Nm,Alm,Kota,KP,Harga_Hash);
    Tambah_Simpul(Kep,Kunci,Harga_Hash);
    writeln;


    Posx := wherex; Posy := wherey;
    repeat
      gotoxy(Posx,Posy);
      write('Ada data lagi <Y>a / <T>idak ? ');
      readln(Ada);
    until Ada in ['Y','y','T','t'];
    writeln;
  until (Ada = 'T') or (Ada = 't')
end;


{* ------------------------------------------------------------ *}
function Ada_Data(Kep        : Li;
                  Kunci      : Str6;
                  Harga_Hash : word) : boolean;
```

```
var Ketemu : boolean;
    Bantu  : Simpul;
begin
  Ketemu := false;
  Bantu := Kep[Harga_Hash].Berikut;
  repeat
    if Bantu^.Info = Kunci  then
       Ketemu := true
    else
       Bantu := Bantu^.Berikut;
  until Ketemu or (Bantu = nil );
  Ada_Data := Ketemu;
end;


{* ------------------------------------------------------------ *}
procedure Hapus_Data(var Kep   : Li);

const Bel = ^G;

var Hapus, Bantu, Awal      : Simpul;
    Data_Ketemu             : boolean;
    Posx, Posy, Harga_Hash  : word;
    Lagi, Del               : char;
    Kunci                   : Str6;

begin
  repeat
    write(' Nomor Pengenal : ');
    readln(Kunci);
    Harga_Hash := HashDivision(Kunci);
    Data_Ketemu := Ada_Data(Kep,Kunci,Harga_Hash);
    if Data_Ketemu = true then {* Data yang dihapus ada *}
       begin
         Awal := Kep[Harga_Hash].Berikut;


         if (Awal^.Berikut = nil ) and (Kunci = Awal^.Info) then
         {* Kepala mempunyai 1 simpul *}
            begin
              Hapus := Awal;
              Kep[Harga_Hash].Berikut := nil ;
              dispose(Hapus);
            end
         {* Kepala mempunyai lebih dari 1 simpul *}
         else if (Awal^.Berikut <> nil ) and (Kunci = Awal^.Info) then
              {* Hapus simpul awal *}
              begin
                Hapus := Awal;
                Awal := Hapus^.Berikut;
                Kep[Harga_Hash].Berikut := Awal;
                dispose(Hapus)
              end
         else
```

```
          {* Hapus simpul tengah atau akhir *}
             begin
               Bantu := Awal;
               Hapus := Bantu^.Berikut;
               {* Mencari simpul yang akan dihapus *}
               while (Kunci <> Hapus^.Info) and
                     (Bantu^.Berikut <> nil) do
                 begin
                   Bantu := Bantu^.Berikut;
                   Hapus := Bantu^.Berikut
                 end;
               if Kunci = Hapus^.Info then
               {* Simpul yang dihapus ketemu *}
                  begin
                    if Hapus^.Berikut <> nil then
                    {* Hapus simpul tengah *}
                       Bantu^.Berikut := Hapus^.Berikut
                    else
                       Bantu^.Berikut := nil;
                    dispose(Hapus)
                  end
            end
       end
    else
        writeln(Bel,' Data tidak ada dalam Tabel Hash');
    writeln;
    Posx := wherex; Posy := wherey;
    repeat
      gotoxy(Posx,Posy);
      write('Hapus Lagi <Y>a / <T>idak ? ');
      readln(Lagi);
    until Lagi in ['Y','y','T','t'];
    writeln;
  until (Lagi = 'T') or (Lagi = 't');
end; {* Hapus_Data *}


{* ------------------------------------------------------------ *}
procedure Cetak_Isi_Simpul(Kep        : Li;
                           Kunci      : Str6;
                           Harga_Hash : word);

var Bantu : Simpul;

begin
  Bantu := Kep[Harga_Hash].Berikut;
  repeat
    if Bantu^.Info = Kunci then {* Data ketemu *}
       begin
         write('-----------------------------------------------');
         writeln('--------------------------------');
         write('Kepala[i]   Kunci        Nama                  Alamat');
         writeln('             Kota  Kode Pos');
         write('=============================================');
         writeln('================================');
         write(Harga_Hash:4,' ':4,Bantu^.Info:10,Bantu^.Nama:18);
```

```
            writeln(Bantu^.Alamat:23,Bantu^.Kota:11,Bantu^.KodePos:8);
            write('-----------------------------------------------');
            writeln('--------------------------------');
            Bantu := Bantu^.Berikut;
          end
       else
          Bantu := Bantu^.Berikut;
    until Bantu = nil;
  end;


  {* ------------------------------------------------------------------ *}
  procedure Edit_Data(Kep : Li);

  const Bel = ^G;

  var Selesai, Data_Ketemu   : boolean;
      Posx, Posy, harga_hash : word;
      Lagi                   : char;
      Kunci                  : Str6;

  begin
    Selesai := false;
    repeat
      Data_Ketemu := false;
      write('Nomor Pengenal : ');
      readln(Kunci);
      {* --------------------------
          Menghitung Alamat Hashnya
         -------------------------- *}
      Harga_Hash := HashDivision(Kunci);

      Data_Ketemu := Ada_Data(Kep,Kunci,Harga_Hash);

      if Data_Ketemu = true then
         Cetak_Isi_Simpul(Kep,Kunci,Harga_Hash)
      else
         begin
           writeln;
           writeln(Bel,'Data tidak ada di Tabel Hash');
         end;
      writeln;
      Posx := wherex; Posy := wherey;
      repeat
        gotoxy(Posx,Posy);
        write('Ingin lihat lagi <Y>a / <T>idak ? ');
        readln(Lagi);
      until Lagi in ['Y','y','T','t'];
      writeln;
      if (Lagi = 'T') or (Lagi = 't') then
           Selesai := true;
    until Selesai;
  end; {* Edit_Data *}
```

```
{* ------------------------------------------------------------------ *}
procedure Buat_Judul_Tabel;
begin
  writeln('             TABEL HASH SEPARATE CHAINING');
  writeln('             ============================');
  writeln
end;


{* ------------------------------------------------------------------ *}
procedure Cetak_Tabel_Hash(Kep : Li);

var Bantu : Simpul;
    i     : word;

begin
  clrscr;
  Buat_Judul_Tabel;
  Buat_Garis;
  writeln(' i | Kepala[i] |              Kunci-kunci');
  Buat_Garis;
  writeln;
  for i := 1 to Uk_Tabel do
  begin
    write(i:2,' | ');write(i:6,' ':3,' |');
    if Kep[i].Berikut <> nil then
       begin
         Bantu := Kep[i].Berikut;
         write(Bantu^.Info:8);
         while Bantu^.Berikut <> nil do
           begin
             Bantu := Bantu^.Berikut;
             write(Bantu^.Info:8);
           end;
       end;
    writeln;
  end;
  Buat_Garis;
  readln
end; {* Cetak_Tabel_Hash *}


{* ----------------------  Program Utama  ---------------------- *}
begin
  clrscr;
  writeln(' Program Hashing dengan Separate Chaining');
  writeln(' ========================================');
  {* Tampilkan judul menu *}
  repeat
    writeln;
    writeln('<A>. Buat Tabel Hash Baru ');
    writeln('<B>. Lihat Tabel Hash ');
    writeln('<C>. Tambah Data ');
    writeln('<D>. Hapus Data ');
    writeln('<E>. Edit Data ');
    writeln('<F>. Selesai ');
```

```
      writeln;
      Posx := wherex; Posy := wherey;
      repeat
        gotoxy(Posx,Posy);
        write('Pilihan Anda : ');
        readln(Pilih);
      until Pilih in ['A'..'F','a'..'f'];
      writeln;
      Pilihan := upcase(Pilih);
      Selesai := false;
      case Pilihan of
        'A' : begin
                Cek_Tabel := Ada_Tabel(Kepala);
                if Cek_Tabel = true then
                   begin
                     writeln(Bel,'         Perhatian !');
                     writeln('     Tabel sudah berisi data.');
                     writeln('Jika dilanjutkan,
                             data lama akan terhapus.');
                     writeln;
                     Posx := wherex; Posy := wherey;
                     repeat
                       gotoxy(Posx,Posy);
                       write('<L>anjutkan / <T>idak ? ');
                       readln(Lanjut);
                     until Lanjut in ['L','l','T','t'];
                     if (Lanjut = 'L') or (Lanjut = 'l') then
                        begin {* Buat tabel hash baru *}
                          Inisialisasi_Tabel_Hash(Kepala);
                          Baca_Tambah_Simpul(Kepala)
                        end
                   end
                else  {* Tabel belum pernah dibuat *}
                   begin
                     Inisialisasi_Tabel_Hash(Kepala);
                     Baca_Teks_Tambah_Simpul(Kepala)
                   end
              end;
        'B' : begin
                Cetak_Tabel_Hash(Kepala)
              end;
        'C' : begin
                Baca_Tambah_Simpul(Kepala)
              end;
        'D' : begin
                Hapus_Data(Kepala)
              end;
        'E' : begin
                Edit_Data(Kepala)
              end;
        'F' : Selesai := true
      end {* case Pilihan *}
    until Selesai;
    write('Program dihentikan oleh User ...');
    readln
  end. {* Program Utama *}
```

Lampiran 2

```
Program Hashing dengan Separate Chaining
========================================

  <A>. Buat Tabel Hash Baru
  <B>. Lihat Tabel Hash
  <C>. Tambah Data
  <D>. Hapus Data
  <E>. Edit Data
  <F>. Selesai

Pilihan Anda : A

Nomor Pengenal : afk52
 Nama    : Toni Khalimi
 Alamat : Jl. Elang D.V./56
 Kota    : Cirebon
 Kode Pos : 45142
Alamat Tabel Hashnya : 2

Nomor Pengenal : cdb49
 Nama    : Purnomo
 Alamat : Jl. Banyu Putih /10
 Kota    : Bandung
 Kode Pos : 50273
Alamat Tabel Hashnya : 10

Nomor Pengenal : rty09
 Nama    : Agus Sumarjo
 Alamat : Jl. Tumpang M. I/03
 Kota    : Malang
 Kode Pos : 67036
Alamat Tabel Hashnya : 5

Nomor Pengenal : olk34
 Nama    : Ishadi Idris
 Alamat : Jl. Kali Gawe no. 4
 Kota    : Semarang
 Kode Pos : 50773
Alamat Tabel Hashnya : 0

Nomor Pengenal : gjw36
 Nama    : Dana Iswara
 Alamat : Jl. Bimbang IV / 23
 Kota    : Solo
 Kode Pos : 49051
Alamat Tabel Hashnya : 4
```

```
Nomor Pengenal : sdr78
 Nama    : Rudi Tamara
 Alamat  : Jl Bukit Sari I/24
 Kota    : Semarang
 Kode Pos : 50274
Alamat Tabel Hashnya : 0

Nomor Pengenal : asm 4
 Nama    : Lusi Yulianti
 Alamat  : Jl. Mangkang II/22
 Kota    : Bali
 Kode Pos : 46779
Alamat Tabel Hashnya : 9

Nomor Pengenal : yt672
 Nama    : Santi S.S.
 Alamat  : Jl. Tembalang V/39
 Kota    : Jakarta
 Kode Pos : 86030
Alamat Tabel Hashnya : 0

Nomor Pengenal : zh091
 Nama    : Al-Huda K.S.
 Alamat  : Jl. Mangga Bali 23
 Kota    : Surabaya
 Kode Pos : 45386
Alamat Tabel Hashnya : 2

Nomor Pengenal : uc311
 Nama    : U. F. Heru
 Alamat  : Jl. Satria A2/52
 Kota    : Bekasi
 Kode Pos : 88945
Alamat Tabel Hashnya : 2


Program Hashing dengan Separate Chaining
========================================

  <A>. Buat Tabel Hash Baru
  <B>. Lihat Tabel Hash
  <C>. Tambah Data
  <D>. Hapus Data
  <E>. Edit Data
  <F>. Selesai

Pilihan Anda : B
```

TABEL HASH SEPARATE CHAINING

i	Kepala[i]	Kunci-kunci
0	0	olk34 sdr78 yt672
1	1	
2	2	afk52 uc311
3	3	
4	4	gjw36
5	5	rty09
6	6	zh091
7	7	
8	8	
9	9	asm 4
10	10	cdb49

Program Hashing dengan Separate Chaining

<A>. Buat Tabel Hash Baru
<B>. Lihat Tabel Hash
<C>. Tambah Data
<D>. Hapus Data
<E>. Edit Data
<F>. Selesai

Pilihan Anda : C

Nomor Pengenal : an115
Nama : Joko P.S.
Alamat : Jl. Gajah K1/102
Kota : Medan
Kode Pos : 56617
Alamat Tabel Hashnya : 2

Ada data lagi <Y>a / <T>idak ? T

Program Hashing dengan Separate Chaining

<A>. Buat Tabel Hash Baru
<B>. Lihat Tabel Hash
<C>. Tambah Data
<D>. Hapus Data
<E>. Edit Data
<F>. Selesai

Pilihan Anda : B

TABEL HASH SEPARATE CHAINING

i	Kepala[i]	Kunci-kunci		
0	0	o1k34	sdr78	yt672
1	1			
2	2	afk52	uc311	
3	3			
4	4	gjw36		
5	5	rty09		
6	6	zh091	an115	
7	7			
8	8			
9	9	asm 4		
10	10	cdb49		

```
Program Hashing dengan Separate Chaining
========================================

  <A>. Buat Tabel Hash Baru
  <B>. Lihat Tabel Hash
  <C>. Tambah Data
  <D>. Hapus Data
  <E>. Edit Data
  <F>. Selesai

Pilihan Anda : D


 Nomor Pengenal : an115

Hapus Lagi <Y>a / <T>idak ? T

Program Hashing dengan Separate Chaining
========================================

  <A>. Buat Tabel Hash Baru
  <B>. Lihat Tabel Hash
  <C>. Tambah Data
  <D>. Hapus Data
  <E>. Edit Data
  <F>. Selesai

Pilihan Anda : B
```

TABEL HASH SEPARATE CHAINING

i	Kepala[i]	Kunci-kunci
0	0	olk34 sdr78 yt672
1	1	
2	2	afk52 uc311
3	3	
4	4	gjw36
5	5	rty09
6	6	zh091
7	7	
8	8	
9	9	asm 4
10	10	cdb49

```
Program Hashing dengan Separate Chaining
========================================

  <A>. Buat Tabel Hash Baru
  <B>. Lihat Tabel Hash
  <C>. Tambah Data
  <D>. Hapus Data
  <E>. Edit Data
  <F>. Selesai

Pilihan Anda : E

Nomor Pengenal : sdr78
```

Kepala[i]	Kunci	Nama	Alamat	Kota	Kode Pos
0	sdr78	Rudi Tamara	Jl. Bukit Sari I/24	Semarang	50274

```
Ingin lihat lagi <Y>a / <T>idak ? T

Program Hashing dengan Separate Chaining
========================================

  <A>. Buat Tabel Hash Baru
  <B>. Lihat Tabel Hash
  <C>. Tambah Data
  <D>. Hapus Data
  <E>. Edit Data
  <F>. Selesai

Pilihan Anda : F

Program dihentikan oleh User ...
```