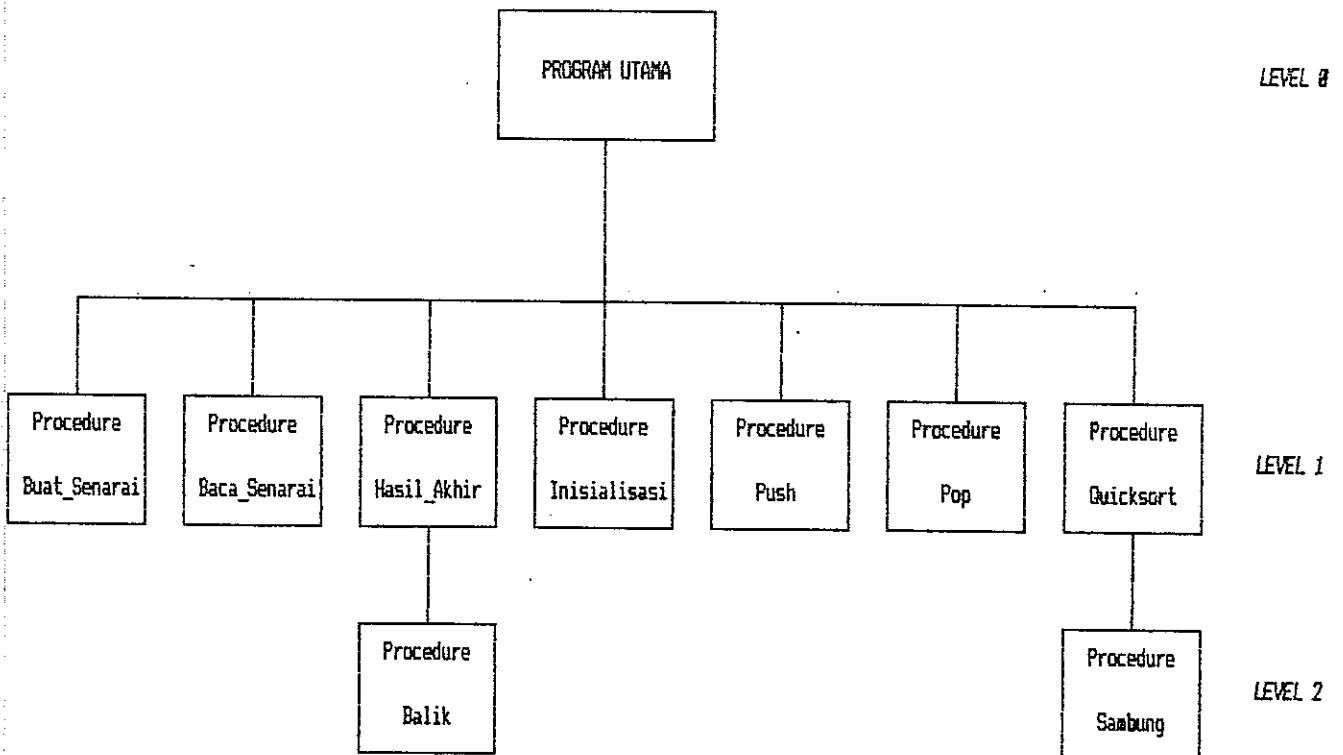


lampiran 1

STRUKTUR PROGRAM
MAKING_CHANGE_GREEDY



lampiran 2

```
{  
Judul Program      : Making_Change_Greedy  
Dibuat oleh       : Mohammad Nur Cholis  
NIM                : J 101 92 0705  
Kepentingan       : Tugas Akhir  
Tanggal Pembuatan : 29 April 1997  
Revisi Akhir      : 8 Agustus 1997  
}
```

Program MakingChangeGreedy;

uses crt;

type simpul = ^data;

data = record

info : integer;

jumlah:integer;

next : simpul;

end;

var

Kepala_1,Kepala_2,Kepala_3,Ekor_1,Kepala_2a,

Ekor_2,Ekor_3,STACK_1,STACK_2,Tail : simpul;

i,n,no,coin,change,check,optimal,

kosong,kumulatif,hasil,hasil_1,jumlah : integer;

masuk,selesai : boolean;

Procedure Buat_senarai(var Awal,Akhir :simpul;coin,sum:integer);

var Baru : simpul;

begin

new(Baru);

Baru^.Info := coin;

Baru^.Jumlah:=sum;

if Awal=nil then

Awal := Baru

else

Akhir^.Next := Baru;

```
Akhir := Baru;
Akhir^.Next:=nil;
end;

Procedure Baca_Senarai(var Awal,Akhir : simpul);
var Bantu : simpul;
begin
  Bantu := Awal;
  write ('W = { ');
  repeat
    write(Bantu^.Info:2);
    Bantu:=Bantu^.Next;
    if Bantu <> nil then write(' ');
  until Bantu = nil;
  writeln(' }');
end;

Procedure Balik(var Awal,Akhir :simpul);
var Bantu, Bantul : simpul;
begin
  Bantu:=Awal;
  Awal:=Akhir;

  {Proses membaca dari senarai terakhir}
  repeat
    Bantul:=Bantu;
    {Cari simpul sebelum simpul terakhir}
    While Bantul^.Next <> Akhir do
      Bantul := Bantul^.Next;
    Akhir^.Next := Bantul;
    Akhir:=Bantul;
  until Akhir = Bantu;
  Akhir^.Next:=nil;
end;
```

```

Procedure Hasil_Akhir(var Awal,Akhir : simpul;total:integer);
var Bantu : simpul;
begin
  {Check jumlah simpul dalam senarai lebih dari satu}
  if Awal^.Next <> nil then
    Balik(Awal,Akhir);
  Bantu := Awal;
  no:=1;
  writeln('          PENYELESAIAN MASALAH MAKING CHANGE');
  writeln('          MENGGUNAKAN ALGORITMA GREEDY');
  writeln; writeln;
  writeln('JUMLAH YANG HARUS DIBAYARKAN = ', change);
  Baca_senarai(Kepala_2a,Ekor_2);
  writeln;
  writeln(' |-----| ');
  writeln(' | NO | SATUAN MATA UANG | JUMLAH | ');
  writeln(' |-----| ');
  repeat
    write(' | ',no:2,' | ');
    write(Bantu^.Info:5,' | ');
    writeln(Bantu^.Jumlah:3,' | ');
    Bantu:=Bantu^.Next;
    inc(no);
    if bantu = nil then
      writeln(' |-----| ');
  until Bantu = nil;
  writeln(' |          TOTAL          | ',total:3,' | ');
  writeln(' |-----| ');
  writeln;
  writeln('JADI DIBUTUHKAN ',total:3,' SATUAN MATA');
end;

Procedure Inisialisasi(var atas:simpul);
begin
  new(Atas);
  Atas^.next:=nil;
end;

```

```
Procedure Push (var Stack : simpul ; Data :integer);
var Baru : simpul;
begin
    new(Baru);
    Baru^.Info:=Data;

    if Stack = nil then          (Tumpukan kosong)
        begin
            Stack:=Baru;
            Stack^.Next:=nil;
        end
    else
        (Tumpukan isi)
        Baru^.Next := Stack;
        Stack:=Baru
end;
```

```
Procedure Pop(var Stack: simpul; var data:integer);
var Bantu : simpul;
begin
    Bantu := Stack;
    Data := Stack^.Info;
    Stack := Stack^.Next;
    dispose(Bantu);
end;
```

```
Procedure Sambung(B:simpul; var S,S1 : simpul);
begin
    if S = nil then
        S := B
    else
        S1^.Next := B;
        S1 := B;
end;
```

```

Procedure Quicksort (var La1 : Simpul; La: Simpul);
var BwF,BwL,TengL,TengF,AtF,AtL : Simpul;
begin
  if La = nil then
    {Senarai berantai kosong}
    begin
      Tail := nil;
      La1 := La;
    end
  else
    {Memproses senarai berantai}
    begin
      BwF := nil;
      TengF := nil;
      AtF := nil;

      {ambil simpul pertama}
      SAMBUNG(La,TengF,TengL);
      La := La^.Next;
      {Kerjakan sampai simpul terakhir}
      while La <> nil do
        begin
          if La^.Info < TengF^.Info then
            SAMBUNG(La,BwF,BwL)
          else if La^.Info = TengF^.Info then
            SAMBUNG(La,TengF,TengL)
          else
            SAMBUNG(La,AtF,AtL);
          La := La^.Next;
        end;
      if BwF <> nil then
        {Menggabung senarai}
        begin
          BwL^.Next := nil;
          QUICKSORT(La1,BwF);
          Tail^.Next := TengF
        end
    end
  end

```

```
        else
            Lal := TengF;
            if AtF <> nil then
                AtL^.Next := nil;
            QUICKSORT(TengL^.Next,AtF);
            if Tail = nil then
                Tail := TengL
        end;
end;
```

{Program Utama}

begin

clrscr;

optimal:=0; *{* harga awal **

kumulatif:=0;

{ Mengosongkan STACK_1 dan STACK_2 **

inisialisasi(STACK_1);

inisialisasi(STACK_2);

{ Input Data **

writeln('PEMASUKAN DATA');

writeln;

write('BESARNYA JUMLAH YANG HARUS DIBAYARKAN KEPADA
 CUSTOMER = ');

readln(change);

writeln;

write('BANYAKNYA SATUAN MATA UANG YANG DIDEFINISIKAN
 = ');

readln(n);

writeln;

writeln;

writeln('SATUAN MATA UANG YANG DIDEFINISIKAN');

writeln;

```

(*Pendefinisian satuan mata uang yang berlaku*)
for i:=1 to n do
  begin
    write('SATUAN MATA UANG KE-',i,' = ');
    readln(coin);
    (*Membentuk senarai berantai*)
    Buat_senarai(Kepala_1,Ekor_1,coin,kosong);
  end;

(* Menampilkan senarai berantai yang terbentuk*)
clrscr;
writeln('SATUAN MATA UANG SEBELUM DIURUTKAN');
writeln;
Baca_senarai(Kepala_1,Ekor_1);

(* Mengurutkan data pada senarai dengan urutan naik*)
quicksort(Kepala_2,Kepala_1);

(* Menampilkan senarai berantai yang terbentuk dan sudah
  terurut*)
writeln; writeln; writeln;
writeln('SATUAN MATA UANG SETELAH DIURUTKAN');
writeln;
Baca_Senarai(Kepala_2,Ekor_2);
Kepala_2a^:=Kepala_2^;

(*Proses membuat/memasukkan simpul ke dalam Stack ke 1)
while Kepala_2<>nil do
  begin
    push(STACK_1,Kepala_2^.Info);
    Kepala_2:=Kepala_2^.Next;
  end;

```



```

{Algoritma Greedy}
selesai:=true;
while kumulatif <> change do
  begin
    pop(STACK_1,coin);
    while kumulatif + coin <= change do
      begin
        push(STACK_2,COIN);
        inc(optimal);
        kumulatif := kumulatif + coin;
      end;

      {Melakukan check apakah satuan mata uang terkecil
      yang berlaku tersedia}
      if STACK_1^.Next = Nil then
        begin
          if (change-kumulatif) mod coin <> 0 then
            begin
              selesai := false;  {tak ditemukan penyelesaian}
              kumulatif := change; {keluar dari looping}
            end;
          end
        end;
      end;

if selesai then
  begin
    {Hitung nominasi koin}
    while STACK_2^.Next <> nil do
      begin
        masuk:=true;
        jumlah:=1;
        pop(STACK_2,hasil);
        while masuk do
          begin
            pop(STACK_2,hasil_1);
            if hasil = hasil_1 then
              inc(jumlah)
          end;
        end;
      end;
    end;
  end;
end;

```

```
        else
            begin
                push(STACK_2,hasil_1);
                masuk:=false;
            end;
        end;
        Buat_Senarai(Kepala_3,Ekor_3,hasil,jumlah);
    end;
    readln;
    clrscr;
    Hasil_Akhir(Kepala_3,Ekor_3,optimal);
end

else (Tidak diperoleh penyelesaian)
    begin
        clrscr;
        writeln('TIDAK DIPEROLEH PENYELESAIAN');
        writeln;
        write('KARENA SATUAN MATA UANG = 1 TIDAK TERSEDIA');
    end;
    readln;
end.
```

PEMASUKAN DATA

BESARNYA JUMLAH YANG HARUS DIBAYARKAN KEPADA CUSTOMER = 289

BANYAKNYA SATUAN MATA UANG YANG DIDEFINISIKAN = 5

SATUAN MATA UANG YANG DIDEFINISIKAN

SATUAN MATA UANG KE-1 = 10
 SATUAN MATA UANG KE-2 = 1
 SATUAN MATA UANG KE-3 = 5
 SATUAN MATA UANG KE-4 = 100
 SATUAN MATA UANG KE-5 = 25

SATUAN MATA UANG SEBELUM DIURUTKAN

$W = \{ 10, 1, 5, 100, 25 \}$

SATUAN MATA UANG SETELAH DIURUTKAN

$W = \{ 1, 5, 10, 25, 100 \}$

PENYELESAIAN MASALAH MAKING CHANGE
 MENGGUNAKAN ALGORITMA GREEDY

JUMLAH YANG HARUS DIBAYARKAN = 289

$W = \{ 1, 5, 10, 25, 100 \}$

NO	SATUAN MATA UANG	JUMLAH
1	100	2
2	25	3
3	10	1
4	1	4
	TOTAL	10

JADI DIBUTUHKAN 10 SATUAN MATA UANG

PEMASUKAN DATA

BESARNYA JUMLAH YANG HARUS DIBAYARKAN KEPADA CUSTOMER = 15

BANYAKNYA SATUAN MATA UANG YANG DIDEFINISIKAN = 3

SATUAN MATA UANG YANG DIDEFINISIKAN

SATUAN MATA UANG KE-1 = 1

SATUAN MATA UANG KE-2 = 5

SATUAN MATA UANG KE-3 = 11

SATUAN MATA UANG SEBELUM DIURUTKAN

$W = \{ 1, 5, 11 \}$

SATUAN MATA UANG SETELAH DIURUTKAN

$W = \{ 1, 5, 11 \}$

PENYELESAIAN MASALAH MAKING CHANGE
MENGUNAKAN ALGORITMA GREEDY

JUMLAH YANG HARUS DIBAYARKAN = 15

$W = \{ 1, 5, 11 \}$

NO	SATUAN MATA UANG	JUMLAH
1	11	1
2	1	4
	TOTAL	5

JADI DIBUTUHKAN 5 SATUAN MATA UANG

PEMASUKAN DATA

BESARNYA JUMLAH YANG HARUS DIBAYARKAN KEPADA CUSTOMER = 352

BANYAKNYA SATUAN MATA UANG YANG DIDEFINISIKAN = 5

SATUAN MATA UANG YANG DIDEFINISIKAN

SATUAN MATA UANG KE-1 = 5

SATUAN MATA UANG KE-2 = 1

SATUAN MATA UANG KE-3 = 4

SATUAN MATA UANG KE-4 = 7

SATUAN MATA UANG KE-5 = 6

SATUAN MATA UANG SEBELUM DIURUTKAN

$W = \{ 5, 1, 4, 7, 6 \}$

SATUAN MATA UANG SETELAH DIURUTKAN

$W = \{ 1, 4, 5, 6, 7 \}$

PENYELESAIAN MASALAH MAKING CHANGE
MENGUNAKAN ALGORITMA GREEDY

JUMLAH YANG HARUS DIBAYARKAN = 352

$W = \{ 1, 4, 5, 6, 7 \}$

NO	SATUAN MATA UANG	JUMLAH
1	7	50
2	1	2
	TOTAL	52

JADI DIBUTUHKAN 52 SATUAN MATA