**Algoritma dan Script Program**

**Konversi Bilangan ke Hexadesimal**

**Disusun oleh:**

**Bintang Farhandika Editya (10118371)**

**Muhammad Rizqi Hidayatullah (10118381)**

**Kelas : IF - 9**

**Algoritma**  Program Konversi Ke Hexadecimal

**{ I.S:: Nilai bilangan dengan basis biner ,octal atau decimal }**

**{ F.S:: Nilai bilangan dengan basis hexadecimal }**

**Type { jenis variable buatan }**

biner\_hexa = **record**  **{ tipe data untuk membuat array pasangan biner hexa }**

hexa:**string**

biner:**string**

**endrecord**

**Kamus**

array\_biner\_hexa: **array [1..16] of biner\_hexa** **{array untuk tabel pasangan biner dan hexadecimal}**

index\_pilih : **integer** **{ index untuk pilihan menu }**

cariint : **integer** **{ menampung angka yang akan di konversi dari decimal ke hexadecimal }**

caristr: **string** **{ menampung angka yang akan di konversi dari biner atau octal ke hexadecimal }**

**Procedure buat\_array\_biner\_hexa { berguna untuk membuat array pasangan biner dan hexadecimal untuk mempermudah konversi }**

array\_biner\_hexa[1].hexa 🡨 '0'

array\_biner\_hexa[1].biner 🡨 '0000'

array\_biner\_hexa[2].hexa 🡨 '1'

array\_biner\_hexa[2].biner 🡨 '0001'

array\_biner\_hexa[3].hexa 🡨 '2'

array\_biner\_hexa[3].biner 🡨 '0010'

array\_biner\_hexa[4].hexa 🡨 '3'

array\_biner\_hexa[4].biner 🡨 '0011'

array\_biner\_hexa[5].hexa 🡨 '4'

array\_biner\_hexa[5].biner 🡨 '0100'

array\_biner\_hexa[6].hexa 🡨 '5'

array\_biner\_hexa[6].biner 🡨 '0101'

array\_biner\_hexa[7].hexa 🡨 '6'

array\_biner\_hexa[7].biner 🡨 '0110'

array\_biner\_hexa[8].hexa 🡨 '7'

array\_biner\_hexa[8].biner 🡨 '0111'

array\_biner\_hexa[9].hexa 🡨 '8'

array\_biner\_hexa[9].biner 🡨 '1000'

array\_biner\_hexa[10].hexa 🡨 9'

array\_biner\_hexa[10].biner 🡨 '1001'

array\_biner\_hexa[11].hexa 🡨 'A'

array\_biner\_hexa[11].biner 🡨 '1010'

array\_biner\_hexa[12].hexa 🡨 'B'

array\_biner\_hexa[12].biner 🡨 '1011'

array\_biner\_hexa[13].hexa 🡨 'C'

array\_biner\_hexa[13].biner 🡨 '1100'

array\_biner\_hexa[14].hexa 🡨 'D'

array\_biner\_hexa[14].biner 🡨 '1101'

array\_biner\_hexa[15].hexa 🡨 'E'

array\_biner\_hexa[15].biner 🡨 '1110'

array\_biner\_hexa[16].hexa 🡨 'F'

array\_biner\_hexa[16].biner 🡨 '1111'

**endprocedure**

**Function biner\_to\_hexa (val\_biner:string):string {fungsi yang meminta input bilangan biner dan menghasilkan output hasil konversinya ke hexadecimal}**

**{I.S::DITERIMA ANGKA BERBASIS 2 ATAU BINER}**

**{F.S::HASIL KONVERSI DARI ANGKA YANG DITERIMA MENJADI BENTUK BASIS 16 NYA ATAU HEXADECIMAL }**

**kamus**

i,j:**integer** **{sebagai index dalam perulangan “for to do”}**

temp\_hasil:**string** **{variabel *temporary* untuk menyimpan hasil konversi sementara}**

**Algoritma**

temp\_hasil 🡨 ' ' **{ kosongkan temp\_hasil }**

**if** (**length**(val\_biner) **mod** 4) > 0 **then** **{ jika banyak karakter dalam string val\_biner tidak habis dibagi 4 maka lakukan }**

**for** i 🡨 1 **to** (4-(**length**(val\_biner) **mod** 4)) **do**

**{tambahkan “0” secara *string* didepan val\_biner sampai banyak karakter dari *string* val\_biner habis dibagi 4, ini tidak akan mengubah nilai biner dari val\_biner}**

val\_biner 🡨 '0' + val\_biner

**{ berguna agar bilangan biner bisa dikelompokkan empat empat, untuk mempermudah konversi }**

**for** i 🡨 1 **to** (**length**(val\_biner) **div** 4) **do** **{ ulangai dari 1 sampai banyak karakter val\_biner dibagi 4 }**

**for** j 🡨 1 **to** 16 **do** **{ ulangi dari 1 sampai 16 }**

**if** **copy**(val\_biner,((i-1)\*4)+1,4) = array\_biner\_hexa[j].biner **then**

**{ val\_biner di cek setiap empat karakter, lalu dicari pasangan hexadecimalnya dengan bantuan array “array\_biner\_hexa” }**

temp\_hasil:= temp\_hasil + array\_biner\_hexa[j].hexa;

**{ pasangan hexadecimalnya disisipkan pada temp\_hasil }**

**endfor**

**{ pada akhir perulangan, temp\_hasil akan berisi hasil konversi biner ke hexadecimal dengan metode mencocokan setiap 4 karakter , dan dengan bantuan array “array\_biner\_hexa” }**

**endfor**

**if** ((temp\_hasil <> '0') **or** (temp\_hasil <> '')) **then**

**while** **copy**(temp\_hasil,1,1) = '0' **do**

**delete**(temp\_hasil,1,1) **{ menghapus substring ke-1 selagi substring ke-1 adalah 0, menghindari penulisan “0” yang tidak lazim seperti: “00AA”, “0­15E”, 000EF4C, 0000 }**

biner\_to\_hexa 🡨 temp\_hasil **{fungsi ini me-*return* hasil konversi biner ke hexadecimal }**

**endfunction**

**Function octal\_to\_hexa (val\_octal:string):string { fungsi yang meminta input bilangan octal dan menghasilkan output hasil konversinya ke hexadecimal }**

**{I.S::DITERIMA ANGKA BERBASIS 8 ATAU OCTAL}**

**{F.S::HASIL KONVERSI DARI ANGKA YANG DITERIMA MENJADI BENTUK BASIS 16 NYA ATAU HEXADECIMAL }**

**Kamus**

i:**integer** **{ sebagai index dalam perulangan for to do }**

temp\_int:**integer** **{ variabel *temporary* untuk membantu perhitungan }**

temp\_hasil\_biner :**string** **{ variabel *temporary* untuk menyimpan hasil konversi sementara }**

temp\_str:**string** **{ variabel *temporary* untuk membantu perhitungan }**

**Algoritma**

temp\_hasil\_biner 🡨 ' ' **{ kosongkan temp\_hasil\_biner }**

**for** i 🡨 1 **to** (length(val\_octal)) **do** **{ ulangi dari 1 sampai banyak karakter pada val\_octal }**

temp\_str 🡨 **copy**(val\_octal,i,1) **{ temp\_str = substring val\_octal ke-I }**

**val**(temp\_str,temp\_int,codeerror) **{ mengkonversi dari temp\_str bertipe string ke temp\_int bertipe integer agar bisa menjadi index pada case of }**

**case** temp\_int **of**

0:temp\_hasil\_biner 🡨 temp\_hasil\_biner + '000'

1:temp\_hasil\_biner 🡨 temp\_hasil\_biner + '001'

2:temp\_hasil\_biner 🡨 temp\_hasil\_biner + '010'

3:temp\_hasil\_biner 🡨 temp\_hasil\_biner + '011'

4:temp\_hasil\_biner 🡨 temp\_hasil\_biner + '100'

5:temp\_hasil\_biner 🡨 temp\_hasil\_biner + '101'

6:temp\_hasil\_biner 🡨 temp\_hasil\_biner + '110'

7:temp\_hasil\_biner 🡨 temp\_hasil\_biner + '111'

**{ octal dicek satu satu, dicari pasangan binernya pada case of diatas }**

**{ dicari pasangan binernya karena metode konversinya adalah dengan mengkonversi dari octal kebiner dulu baru nanti dikonversikan dari biner ke hexadecimal }**

**endcase**

**endfor** **{ hasil akhir dari perulangan yaitu temp\_hasil\_biner yang bernilai hasil konversi octal di ke biner }**

octal\_to\_hexa:= biner\_to\_hexa(temp\_hasil\_biner) **{ dari biner dikonversikan ke hexadecimal dengan fungsi biner\_to\_hexa yang me-*return* hasil konversi biner ke hexadecimal }**

**{ fungsi octal\_to\_hexa me-*return* hasil konversi octal ke bentuk hexadecimalnya }**

**endfunction**

**Function decimal\_to\_hexa (val\_decimal:integer):string { fungsi yang meminta input bilangan decimal dan menghasilkan output hasil konversinya ke hexadecimal }**

**{I.S::DITERIMA ANGKA BERBASIS 10 ATAU DECIMAL}**

**{F.S::HASIL KONVERSI DARI ANGKA YANG DITERIMA MENJADI BENTUK BASIS 16 NYA ATAU HEXADECIMAL }**

**Kamus**

temp\_hasil:**string** **{ variabel *temporary* untuk menyimpan hasil konversi sementara }**

temp\_str:**string** **{ variabel *temporary* untuk membantu perhitungan }**

temp\_mod\_int:**integer** **{ variabel *temporary* untuk membantu perhitungan }**

**Algoritma**

temp\_hasil 🡨 ' ' **{ kosongkan temp\_hasil }**

**while** val\_decimal >= 1 **do** **{ selagi val\_decimal lebih besar sama dengan 1 ulangi }**

temp\_str 🡨 ' ' **{ kosongkan temp\_str }**

temp\_mod\_int 🡨(val\_decimal mod 16)**{temp\_mod\_int = sisa pembagian val\_decimal dengan 16 }**

**if** temp\_mod\_int = 10 **then** temp\_str 🡨 'A'

**else** **if** temp\_mod\_int = 11 **then** temp\_str 🡨 'B'

**else** **if** temp\_mod\_int = 12 **then** temp\_str 🡨 'C'

**else** **if** temp\_mod\_int = 13 then temp\_str 🡨 'D'

**else** **if** temp\_mod\_int = 14 then temp\_str 🡨 'E'

**else** **if** temp\_mod\_int = 15 then temp\_str 🡨 'F'

**else** **str**(temp\_mod\_int,temp\_str) **{inisialisasi nilai temp\_int ke temp\_str}**

**{ *if else nested* berguna mencocokkan sisa pembagian dengan bilangan hexadecimalnya }**

temp\_hasil 🡨 temp\_str + temp\_hasil **{ temp\_hasil** = **temp\_str ditambah temp\_hasil sebelumnya, ditambahkan secara *string* }**

val\_decimal 🡨 val\_decimal div 16 **{val\_decimal selanjutnya = val\_decimal dibagi 16, dan val\_decimal ini akan digunakan pada perulangan selanjutnya jika val\_decimal lebih besar samadengan 1 }**

**endwhile** **{ dikonversikan dengan metode pembagian dengan 16 dan sisa pembagian }**

decimal\_to\_hexa 🡨 temp\_hasil **{fungsi ini me-*return* hasil konversi decimal ke hexadecimal }**

**endfunction**

**Program Utama**

**Algoritma**

buat\_array\_biner\_hexa **{ jalankan procedure buat\_array\_biner\_hexa membuat array pasangan biner dan hexadecimal untuk mempermudah perhitungan }**

**while** (index\_pilih <> 0) do **begin** **{ selagi index\_pilih tidak sama dengan 0 makan ulangi perulangan. Agar menu bisa diulang kembali kecuali dinputkan 0 pada “index\_pilih” }**

**output**('1.biner to hexadecimal') **{ mencetak menu pilihan dilayar }**

**output**('2.octal to hexadecimal') **{ mencetak menu pilihan dilayar }**

**output**('3.decimal to hexadecimal') **{ mencetak menu pilihan dilayar }**

**output**('0.keluar') **{ mencetak menu pilihan dilayar }**

**output**('pilihan: ')

**input**(index\_pilih) **{ menginputkan nilai index\_pilih }**

**case** index\_pilih **of**

1: **output**('angka yang akan di konversi : ')

**input**(caristr) **{ inputkan angka yang akan di konversi ke hexadecimal }**

**output**(biner\_to\_hexa(caristr)) **{ mencetak kelayar hasil fungsi biner\_to\_hexa dengan input awal caristr }**

2: **output**('angka yang akan di konversi : ')

**input**(caristr) **{ inputkan angka yang akan di konversi ke hexadecimal }**

**output**(octal\_to\_hexa(caristr)) **{ mencetak kelayar hasil fungsi octal\_to\_hexa dengan input awal caristr }**

3: **output**('angka yang akan di konversi : ')

**input**(cariint) **{ inputkan angka yang akan di konversi ke hexadecimal**

**output**(decimal\_to\_hexa(cariint)) **{** **mencetak kelayar hasil fungsi decimal\_to\_hexa dengan input awal cariint }**

**endcase**

**endwhile**

**{ perulangan while akan terulang selagi user tidak menginputkan 0 pada index\_pilih, untuk mepermudah penggunaan program }**

**endprogram.**

**SCRIPT PROGRAM (PASCAL)**

program konversi\_ke\_hexadecimal;

uses crt;

//type--------------------------------------------------------------------

type

biner\_hexa = record // tipe data untuk array biner hexa

hexa:string;

biner:string;

end;

//variabel----------------------------------------------------------------

var

array\_biner\_hexa: array [1..16] of biner\_hexa; //array pasangan biner hexa

index\_pilih,codeerror:integer;

cariint:longint;

caristr:string;

//membuat array pasangan biner hexa 0 sampai F----------------------------

procedure buat\_array\_biner\_hexa;

begin

array\_biner\_hexa[1].hexa:='0'; array\_biner\_hexa[1].biner:='0000';

array\_biner\_hexa[2].hexa:='1'; array\_biner\_hexa[2].biner:='0001';

array\_biner\_hexa[3].hexa:='2'; array\_biner\_hexa[3].biner:='0010';

array\_biner\_hexa[4].hexa:='3'; array\_biner\_hexa[4].biner:='0011';

array\_biner\_hexa[5].hexa:='4'; array\_biner\_hexa[5].biner:='0100';

array\_biner\_hexa[6].hexa:='5'; array\_biner\_hexa[6].biner:='0101';

array\_biner\_hexa[7].hexa:='6'; array\_biner\_hexa[7].biner:='0110';

array\_biner\_hexa[8].hexa:='7'; array\_biner\_hexa[8].biner:='0111';

array\_biner\_hexa[9].hexa:='8'; array\_biner\_hexa[9].biner:='1000';

array\_biner\_hexa[10].hexa:='9'; array\_biner\_hexa[10].biner:='1001';

array\_biner\_hexa[11].hexa:='A'; array\_biner\_hexa[11].biner:='1010';

array\_biner\_hexa[12].hexa:='B'; array\_biner\_hexa[12].biner:='1011';

array\_biner\_hexa[13].hexa:='C'; array\_biner\_hexa[13].biner:='1100';

array\_biner\_hexa[14].hexa:='D'; array\_biner\_hexa[14].biner:='1101';

array\_biner\_hexa[15].hexa:='E'; array\_biner\_hexa[15].biner:='1110';

array\_biner\_hexa[16].hexa:='F'; array\_biner\_hexa[16].biner:='1111';

end;

//------------------------------------------------------------------------

function biner\_to\_hexa (val\_biner:string):string;

{I.S::DITERIMA ANGKA BERBASIS 2 ATAU BINER}

{F.S::HASIL KONVERSI DARI ANGKA YANG DITERIMA MENJADI BENTUK BASIS 16 NYA ATAU HEXADECIMAL }

var

i,j:integer;

temp\_hasil:string;

begin

temp\_hasil:='';

if (length(val\_biner) mod 4) > 0 then

for i:= 1 to (4-(length(val\_biner) mod 4)) do

val\_biner:= '0' + val\_biner;

// membuat panjang value\_biner bisa dibagi 4

for i:= 1 to (length(val\_biner) div 4) do begin

for j:= 1 to 16 do begin

if copy(val\_biner,((i-1)\*4)+1,4) = array\_biner\_hexa[j].biner then

temp\_hasil:= temp\_hasil + array\_biner\_hexa[j].hexa;

end;

end;

if ((temp\_hasil <> '0') or (temp\_hasil <> '')) then

while copy(temp\_hasil,1,1) = '0' do

delete(temp\_hasil,1,1);

// menghaspus substring ke-1 selagi substring ke-1 0, menghindari penulisan seperti: 00AA, 015E, 000EF4C, 0000

biner\_to\_hexa:= temp\_hasil; //return dari hasil konversi

end;

//------------------------------------------------------------------------

function octal\_to\_hexa (val\_octal:string):string;

{I.S::DITERIMA ANGKA BERBASIS 8 ATAU OCTAL}

{F.S::HASIL KONVERSI DARI ANGKA YANG DITERIMA MENJADI BENTUK BASIS 16 NYA ATAU HEXADECIMAL }

var

i,temp\_int:integer;

temp\_hasil\_biner,temp\_str:string;

begin

temp\_hasil\_biner:='';

for i:= 1 to (length(val\_octal))do begin

temp\_str:=copy(val\_octal,i,1);

val(temp\_str,temp\_int,codeerror);

//mengkonversi dari temp\_str bertipe string ke temp\_int bertipe integer agar bisa menjadi index pada case of

case temp\_int of

0:temp\_hasil\_biner:= temp\_hasil\_biner + '000';

1:temp\_hasil\_biner:= temp\_hasil\_biner + '001';

2:temp\_hasil\_biner:= temp\_hasil\_biner + '010';

3:temp\_hasil\_biner:= temp\_hasil\_biner + '011';

4:temp\_hasil\_biner:= temp\_hasil\_biner + '100';

5:temp\_hasil\_biner:= temp\_hasil\_biner + '101';

6:temp\_hasil\_biner:= temp\_hasil\_biner + '110';

7:temp\_hasil\_biner:= temp\_hasil\_biner + '111';

end;

end; // dari octal dikonversikan dulu ke biner,

octal\_to\_hexa:= biner\_to\_hexa(temp\_hasil\_biner);

// dari biner dikonversikan ke hexa dengan fungsi biner\_to\_hexa()

end;

//------------------------------------------------------------------------

function decimal\_to\_hexa (val\_decimal:longint):string;

{I.S::DITERIMA ANGKA BERBASIS 10 ATAU DECIMAL}

{F.S::HASIL KONVERSI DARI ANGKA YANG DITERIMA MENJADI BENTUK BASIS 16 NYA ATAU HEXADECIMAL }

var

temp\_hasil,temp\_str:string;

temp\_mod\_int:integer;

begin

temp\_hasil:='';

while val\_decimal >= 1 do begin

temp\_str:='';

temp\_mod\_int:=(val\_decimal mod 16);

if temp\_mod\_int = 10 then temp\_str := 'A'

else if temp\_mod\_int = 11 then temp\_str := 'B'

else if temp\_mod\_int = 12 then temp\_str := 'C'

else if temp\_mod\_int = 13 then temp\_str := 'D'

else if temp\_mod\_int = 14 then temp\_str := 'E'

else if temp\_mod\_int = 15 then temp\_str := 'F'

else str(temp\_mod\_int,temp\_str);

temp\_hasil := temp\_str + temp\_hasil;

val\_decimal := val\_decimal div 16;

end;

//dikonversikan dengan metode pembagian dengan 16 dan sisa pembagiannya

decimal\_to\_hexa := temp\_hasil;

end;

//program utama-----------------------------------------------------------

begin

buat\_array\_biner\_hexa;

index\_pilih := 9; // menghindari error pada while

while (index\_pilih <> 0) do begin // agar menu bisa diulang kembali kecuali dinputkan 0 pada "index\_pilih"

clrscr;

writeln('1.biner to hexadecimal');

writeln('2.octal to hexadecimal');

writeln('3.decimal to hexadecimal');

writeln('0.keluar');

write('pilihan: ');readln(index\_pilih); //inputkan pilihan

clrscr;

case index\_pilih of

1: begin

write('angka yang akan dikonversi: '); readln(caristr); // inputkan angka yang akan di konversi ke hexadecimal

writeln('hasil konversi: ',biner\_to\_hexa(caristr));

// biner\_to\_hexa memakai input caristr yang bertipe string

end;

2: begin

write('angka yang akan dikonversi: '); readln(caristr); // inputkan angka yang akan di konversi ke hexadecimal

writeln('hasil konversi: ',octal\_to\_hexa(caristr)); // octal\_to\_hexa memakai input caristr yang bertipe string

end;

3:begin

write('angka yang akan dikonversi: '); readln(cariint); // inputkan angka yang akan di konversi ke hexadecimal

writeln('hasil konversi: ',decimal\_to\_hexa(cariint));

// decimal\_to\_hexa memakai input cariint yang bertipe integer

end;

0: writeln('Tekan Enter Jika Ingin Keluar');

end;

readln();

end;

end.