TUGAS MATA KULIAH TEKNIK KOMPILASI

“TEKNIK OPTIMASI DAN TABEL INFORMASI”

DOSEN : HARUN MUKHTAR,M.KOM

Nama : Walter Sitio

NIM : 080403019

Jurusan : Teknik Infomatika

UNIVERSITAS MUHAMMAHDIYAH RIAU

Teknik Optimasi

Teknik Optimasi terdiri dari 3 jenis, yaitu :

- Dependensi Optimasi
- Optimasi Lokal
- Optimasi Global

- ***Dependensi Optimasi***

Dependensi Optimasi bertujuan untuk menghasilkan kode program yang berukuran lebih kecil dan lebih cepat

  - Machine Dependent Optimizer
  - Machine Independent Optimizer (Optimasi lokal dan Optimasi global)

- **Optimasi Lokal**

Optimasi Lokal : adalah optimasi yang dilakukan hanya pada suatu blok dari source code, dengan cara:

- ***Folding***

menganti konstata atau ekpresi yang bisa dievaluasi pada saat *compile time* dengan nilai komputasinya. Misalnya:

A := 2 + 3 + B bisa diganti dengan A:= 5 + B

5 dapat mengantikan ekspresi 2 + 3

- *Redundant-Subexpression Elimination*

hasilnya digunakan lagi dari pada dilakukan computasi ulang, contoh:

A:= B + C

X := Y + B + C

- Optimasi dalam sebuah Iterasi
  - *Loop Unrrolling*:Menganti suatu *loop* dengan menulis statement yang ada dalam loop ditulis beberapa kali
  - Karena sebuah iterasi pada implemnetasi ke level rendah, memerlukan :
    - Inisialisasi nilai awal, pada loop dilakukan sekali pada saat permulaan eksekusi loop

- Penge-test-an, apakah variabel loop telah mencapai kondisi terminasi
- Adjustment yaitu: penambahan atau pengurangan nilai pada variabel loop dengan jumlah tertentu
- Operasi yang terjadi pada tubuh perulangan (loop body)

- Contoh :

```
FOR I := 1 to 2 DO
   A[I] := 0;
```

dapat dioptimasikan menjadi

```
A[1] := 0;
A[2] := 0;
```

- Frequency Reduction: Pemindahan statement ke tempat yang lebih jarang dieksekusi, contoh

```
FOR I:= 1 to 10 DO          X := 5
BEGIN                       FOR I:= 1 to 10 DO
  X := 5                    BEGIN
    A := A + 1                    A := A + 1
    END:                    END:
```

- Strength Reduction
  - Penggantian suatu operasi dengan operasi lain yang lebih cepat dieksekusi
  - misalnya: pada komputer operasi perkalian memerlukan waktu eksekusi lebih banyak dari pada operasi penjumlahan
  - contoh lain

    A:= A + 1
  - dapat digantikan dengan

    INC(A)

Optimasi global biasanya dilakukan dengan suatu graph terarah yang menunjukkan jalur yang mungkin selama eksekusi program

ada dua kegunaan yaitu bagi programmer dan compiler itu sendiri

- Bagi Programmer

- *Unreachable/dead code*: Kode yang tidak pernah dieksekusi
- misalnya :

```
X :=  5;
IF X = 0 THEN
     A := A + 1
Instruksi
    A := A + 1 tidak pernah dikerjakan
```

- *Unused parameter* : parameter yang tidak pernah digunakan dalam procedure
  - Misalnya :

```
procedure penjumlahan(a,b,c ; Integer);
var x : integer;
begin
  x := a + b;
end
```

Parameter c tidak pernah digunakan sehingga tidak perlu diikut sertakan

- *Unsused Variabel :* variabel yang yang tidak pernah dipergunakan

```
Program pendek;
var a, b: integer
begin
    a := 5;
end;
```

B tidak pernah digunakan

- *Variabel :* variabel yang dipakai tanpa nilai awal. Contoh

```
Program Awal;
var a, b: integer
begin
    a := 5
    a := a + b;
```

end;

- variabel b digunakan tetapi tidak memiliki harga awal

- *Bagi Compiler*
  - Meningkatkan efisiensi eksekusi program
  - Menghilangkan useless code/kode yang tidak terpakai

```
var A, B, C, D, E, I, J, X, Y : integer;
begin
        B := 5;
A := 10 - B / 4 * 3 + 2;
C :=  A + B;
 Y := 10;
 D := A + B – E;
for  I := 1 to 85 do
begin
        X := X + 1;
        B := B – X;
        Y := 7;
end
While Y< 5 do
        E := E – B;
end
```

Tabel Informasi

Dua fungsi penting Tabel Informasi

- Untuk membantu pemeriksaan kebenaran semantik dari program sumber
- Untuk membantu dan mempermudah dalam pembuatan intermediate code dan proses pembuatan kode-kode (pembangkitan kode)

Secara umum, sebuah tabel simbol bisa memiliki elemen-elemen tabel sebagai berikut, meskipun tidak semuanya dipergunakan oleh semua compiler

- No.urut identifier: menentukan nomor urut pada tabel simbol

- Nama identifier
- Tipe identifier
- Object time address
- Dimensi dari identifier yang bersangkutan
- Nomor baris variabel yang dideklarasikan
- Nomor baris variabel yang direferensikan
- Field link

Tabel Informasi : Implementasi

Ada beberapa jenis Tabel Informasi

- *Tabel identifier*; berfungsi menampung semua identifier yang terdapat dalam program
- *Tabel Array:* berfungsi menampung informasi tambahan untuk sebuah array
- *Tabel blok*: mencatat varibel-variabel yang ada pada blok yang sama
- *Tabel Real*: Menyimpan elemen tabel bernilai real
- *Tabel string*: menyimpan informasi string
- *Tabel display*: mencatat blok yang aktif

Tabel Identifier memiliki;

- No Urut identifier dalam tabel
- Nama Identifier
- Jenis dari identifier; seperti Prosedur, fungsi, tipe variabel dan konstanta
- Tipe dari identifier yang bersangkutan; seperti Integer (bilangan bulat), Char, boolean , array, record, file
- level dari identifier (depth of block); hal ini menyangkut letak identifier dalam program, konsepnya sama dengan pembentukan *tree*, misalnya main program level 0

Untuk identifier, pencatatan dapat berupa seperti;

- Alamat *relatif/address* dari identifier untuk implementasi

- Informasi referensi dari identifier terntentu ke alamat tabel identifier yang lainnya
- *link;* menghubung antar identifier
- Normal: digunakan pada pemanggilan parameter, untuk membedakan parameter by value dan by reference
- Contoh (dalam pascal)

```
Program A;
Var B : Integer;
    Procedure  X (Z: char)
    var C : Integer
    begin
      ....dst
```

Tabel identifier akan mencatat semua identifier;

1	A
2	B
3	X
4	Z
5	C

```
Table: Array [0..tabmax] of  record
    nama : String;
    link    : integer;
    Obj    : object;
Tipe : Types;
    ref     : Integer;
   normal: Boolean;
   Level   : 0.. Maxlevel;
   address        : Integer;
End
```

Dimana

objek =(konstant, variabel, prosedur, fungsi)

Types = (notipe, int, reals, booleans, chars, arrays, record

Tabel Array

dipergunakan untuk menyimpan informasi suatu identifier yang bertipe array, tabel ini memilik field:

- No. Urut suatu array dalam tabel
- Tipe dari indeks array yang bersangkutan
- Tipe element array
- Referensi dari elemen array
- Index batas atas dan bawah array
- Jumlah elemen array
- Ukuran total array (total = atas - bawah + 1) x elemen size
- Elemen size

Tabel Blok

Dipergunakan untuk menyimpan informasi blok-blok yang ada pada tabel utama. Berisikan field

- no urut blok
- batas awal blok
- batas akhir blok
- ukuran parameter/parameter size
- ukuran variabel/ variabel size
- last variabel
- last parameter

Contoh

```
Program A
Var B : Integer;
   Procedure X (Z:char);
    Var C : Integer;
```

begin

....

Untuk	Blok A	Blok B
last variable	= 2	4
Variable size	= 2 (dianggap int 2 byte)	2
Last parameter	= 0 (tanpa parameter)	3
parameter size	= 0	1 (char butuh 1 byte)

Tabel Real

Dipergunakan untuk menyimpan nilai dari suatu identifier yang bertipe real (pecahan). Elemen-elemen dari tabel ini adalah sebagai berikut;

- NO urut elemen
- Nilai real suatu variabel real yang mengacu ke indeks tabel ini

Pemikirannya disini setiap tipe yang memiliki oleh suatu bahasa akan memiliki tabelnya sendiri

Tabel String

Dipergunakan untuk menyimpan informasi string yang terdapat pada program sumber. Elemen-elemen yang terdapat dalam tabel ini adalah:

- no Urut elemen
- Karakter-karakter yang merupakan konstanta

Tabel Display

menyimpan informasi-informasi mengenai blok-blok yang lagi aktif. Elemen-elemen yang terdapat dalam tabel ini adalah:

- No Urut tabel
- Blok yang aktif

Pengisian tabel display dilakukan dengan konsep stack