

User Define Procedure (UDP)

wijanarto

Kompleksitas

- Menentukan kompleksitas pada user define procedure mirip dengan menentukan built-in procedure, yaitu saat aplikasi (procedure call)
- Sebelum menentukan nilai saat procedure call, kita harus menentukan kompleksitas di dalamnya terlebih dahulu
- Isi dari badan procedure biasanya terdiri dari struktur dasar algoritma, built-in atau user define procedure (mungkin recursive, di bahas dalam slide tersendiri).

contoh

```
Procedure Bla(var result:Integer,n:Integer)
Var i,x,y:Integer;
Begin
  for i:=1 to n do
    x:=3*n+1;
    if(x<50)then
      begin
        y:=2*x;
        result:=3*y+2;
      end
    else
      begin
        y:=3*x-1 mod 2;
        y:=25*y;
        result:=y+x*3;
      end;
    end;
  end
```

contoh

```
Procedure Bla(var result:Integer,n:Integer)
```

```
Var i,x,y:Integer;
```

```
Begin //loop
```

```
  for i:=1 to n do // (n-1+2)+(n-1+1) (p+1)
```

```
    x:=3*n+1; // (n+1)+3n=4n+1
```

```
  if(x<50) then //k=1
```

```
    begin
```

```
      y:=2*x; //s1=3
```

```
      result:=3*y+2;
```

```
    end
```

```
  else
```

```
    begin
```

```
      y:=3*x-1 mod 2; //s2=7
```

```
      y:=25*y;
```

```
      result:=y+x*3;
```

```
    end;
```

```
End
```

```
Analisa Bla : loop+k+max(s1,s) → (4n+1)+1+max(3,7) → 4n+9 → O(n)
```

Contoh : built-in + UDP

```
Procedure Bla1(var result:Integer,n:Integer)
```

```
Var i,x,y,z,p:Integer;
```

```
Begin
```

```
z:=exp(2,n)+sin(n div 2);
```

```
for i:=1 to n+1 do
```

```
  x:=3*n+1 div 3;
```

```
  if(x<50)then
```

```
    begin
```

```
      Bla(p,n);
```

```
      y:=2*x+z;
```

```
      result:=3*y+2;
```

```
    end
```

```
  else
```

```
    begin
```

```
      y:=3*x-1 mod 2;
```

```
      y:=25*y;
```

```
      result:=y+x*3;
```

```
    end;
```

```
End
```

Contoh : built-in + UDP

```
Procedure Bla1(var result:Integer,n:Integer)
```

```
Var i,x,y,z,p:Integer;
```

```
Begin
```

```
  z:=exp(2,n)+sin(n div 2); //S1
```

```
  for i:=1 to n+1 do
```

```
    x:=3*n+1 div 3;
```

```
  if(x<50) then //k=1
```

```
    begin
```

```
      Bla(p,n); //S2
```

```
      y:=2*x+z;
```

```
      result:=3*y+2;
```

```
    end
```

```
  else
```

```
    begin
```

```
      y:=3*x-1 mod 2; //S3
```

```
      y:=25*y;
```

```
      result:=y+x*3;
```

```
    end;
```

```
End
```

Analisa Bla1

- S1

$$6 + ((n+1) - 1 + 2) + ((n+1) - 1 + 1)(4)$$

$$6 + (n+2) + (n+1).4$$

$$6 + (2n+3).4$$

$$6 + 8n + 24$$

$$8n + 30$$

- k = 1

- S2

$$(4n+9)+4$$

$$4n+13$$

- S3=7

Analisa Bla1

$$S_1 + s_2 + s_3$$

$$8n + 30 + 1 + \max(s_2, s_3)$$

$$8n + 30 + 1 + \max(4n + 13, 7)$$

$$8n + 30 + 1 + 4n + 13$$

$$12n + 44$$

- Jadi Bla1 = $12n + 44 \in O(n)$

Rangkuman

- Penentuan kompleksitas UDP tergantung pada isi badan procedure
- Isi badan procedure terdiri dari
 - Struktur dasar algoritma
 - Built-in function call, UDP call
- Dalam kuliah ini lebih banyak akan dilakukan **latihan** dengan menggunakan media **whiteboard**
- **Quiz** mungkin dapat di berikan pada pertemuan ini
- **PR** mungkin dapat di berikan pada pertemuan ini

Latihan 1

Procedure Pusing(var hasil:Integer, n:Integer, D:array[1..n]of integer)

Begin

 for i:=1 to n do

 x:= 3*n+2

 if ((x mod 2) = 0) then

 D[i]:= x *3 +1

 else

 x:= x div 2

 D[i]:= x *3 +1

 if (x mod 2 =0) then

 for i:=1 to n-1 do

 D[i]:=D[i+1]+2;

 else

 for i:=1 to n do

 D[i]:= sqrt(D[i]*2)

 for k:=1 to n-1 do

 x:= D[k+1]*2

 hasil:= x*3-1

End

Tentukan waktu tempuh dan big oh dari procedure diatas

Latihan 2

```
Procedure Bingung(var hasil:Integer, n:Integer, D:array[1..n]of integer)
Begin
if (n mod 2 =0) then
  for i:=1 to n-1 do
    D[i]:=D[i+1]+2;
  Pusing(hasil,n,D);
else
  for i:=1 to n do
    D[i]:= power(D[i],n)
  for k:=1 to n-1 do
    x:= D[k+1]*2
for i:=1 to n do
  if ((x mod 2) = 0) then
    D[i]:= x *3 +1
    x:= 3*n+2
  else
    x:= x div 2
    D[i]:= x *3 +1
hasil:= x*3-1
End
```

Dengan procedure Pusing sebelumnya, tentukan waktu tempuh dan big oh dari procedure diatas