Pemrograman Komputer

oleh
Djoko Luknanto
Departemen Teknik Sipil dan Lingkungan
Fakultas Teknik Universitas Gadjah Mada

Evaluasi Polinomial

Bentuk umum

$$p_n(x) = \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

Contoh:

$$p_n(2) = \sum_{i=0}^{n} a_i x^i = a_0 + a_1 2 + a_2 2^2 + \ldots + a_n 2^n$$

$$p_2(x) = 6 + 5x + x^2$$

$$p_2(2) = 6 + 5(2) + (2)^2 = 6 + 10 + 4 = 20$$

Secara numeris evaluasi di atas tidak efisien!

Evaluasi Polinomial Efisien

Bentuk umum

$$p_n(x) = \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

diubah menjadi (cara Horner):

$$p_n(x) = a_0 + x(a_1 + x(a_2 + \ldots + x(a_{n-1} + a_n x)\ldots))$$

terdapat pola berulang $a_n x + a_{n-1}$

bentuk akhir menjadi:

$$p_n(x) = ((\ldots(a_n x + a_{n-1})x + a_{n-2})x + \ldots + a_1)x + a_0$$

Evaluasi Polinomial Efisien

Contoh $n = 4$

$$p_4(x) = (((a_4 x + a_3)x + a_2)x + a_1)x + a_0$$

Dalam pemrograman bentuk umum

$$p_n(x) = ((...(a_n x + a_{n-1})x + a_{n-2})x + ... + a_1)x + a_0$$

$$p_n = (((a_n x + a_{n-1})x + a_{n-2})x + \cdots)x + a_0$$

algoritmanya menjadi:

1. $P = A(n)$
2. For $i = n$-1 to 0
   $P = P*X+A(i)$
   Next i

Algoritmanya sangat sederhana dan langsung

Evaluasi Polinomial Cara Biasa

Indeks polinomial disesuaikan

$$p_n(x) = \sum_{i=0}^{n} a_i x^i = \sum_{j=1}^{m} a_j x^{j-1} = a_1 + a_2 x + a_3 x^2 + \ldots + a_m x^{m-1}$$

```
Function Polinomial(X As Single, A As Range) As Double

' Evaluasi polinomial dengan cara biasa

Dim j As Integer, Hasil As Double

  Hasil = A(1, 1)
  For j = 2 To A.Count
      Hasil = Hasil + A(1, j) * X ^ (j - 1)
  Next j
  Polinomial = Hasil

End Function
```

perkalian

perpangkatan

penambahan

Evaluasi Polinomial Cara Horner

Bentuk akhir cara Horner

$$p_m(x) = ((...(a_m x + a_{m-1})x + a_{m-2})x + ... + a_2)x + a_1$$

```
Function PoliHorner(X As Single, A As Range) As Double

' Evaluasi polinomial dengan cara

Dim i As Integer, Hasil As Double

  Hasil = A(1, A.Count)
  For i = A.Count - 1 To 1 Step -1
      Hasil = Hasil * X + A(1, i)
  Next i
  PoliHorner = Hasil

End Function
```

Indeks dalam Range artinya *A*(baris, kolom)

tanpa perpangkatan

penambahan

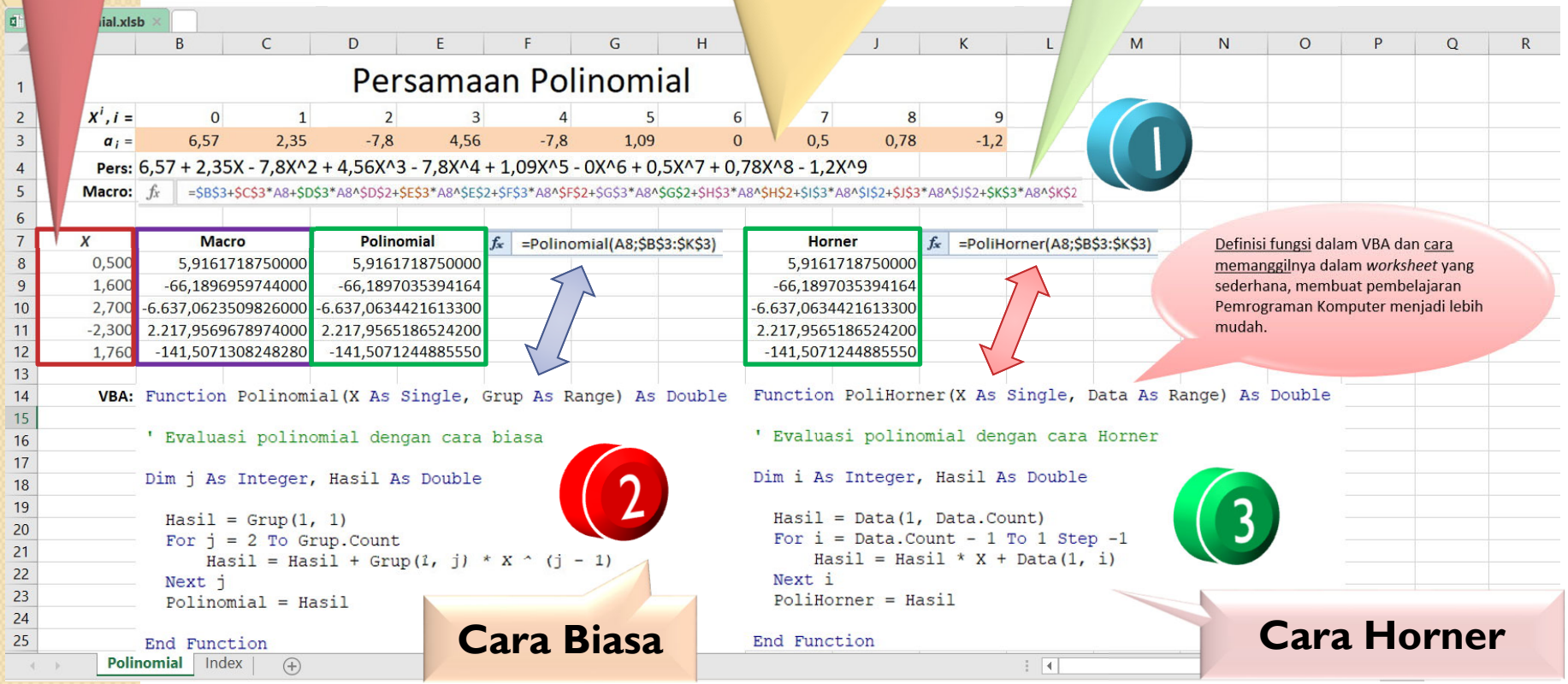
perkalian

VBA-Excel: Cara Evaluasi Polinomial

X sebagai input pertama

Koefisien polynomial Range ($B$3:$K$3) sebagai input kedua

Menggunakan Macro



Hasil hitungan Cara Biasa dan Cara Horner identik, namun dari sisi efisiensi Cara Horner lebih bagus!

Syntax: For...Next Loop

For *counter* = *start* **To** *end* [**Step** step]
[*statements*]
[**Exit For**]
[*statements*]
Next [*counter*]

Perhatikan konvensi penulisan dalam manual Bahasa Pemrograman VBA

Penjelasan dalam [...] boleh tidak ada!

The **For...Next** statement syntax has these parts:

Part	**Description**
counter	Required. Numeric variable used as a loop counter. The variable can't be a Boolean or an array element.
start	Required. Initial value of *counter*.
end	Required. Final value of *counter*.
step	Optional. Amount *counter* is changed each time through the loop. If not specified, *step* defaults to one.
statements	Optional. One or more statements between **For** and **Next** that are executed the specified number of times.

oleh
Djoko Luknanto
Departemen Teknik Sipil dan Lingkungan
Fakultas Teknik Universitas Gadjah Mada

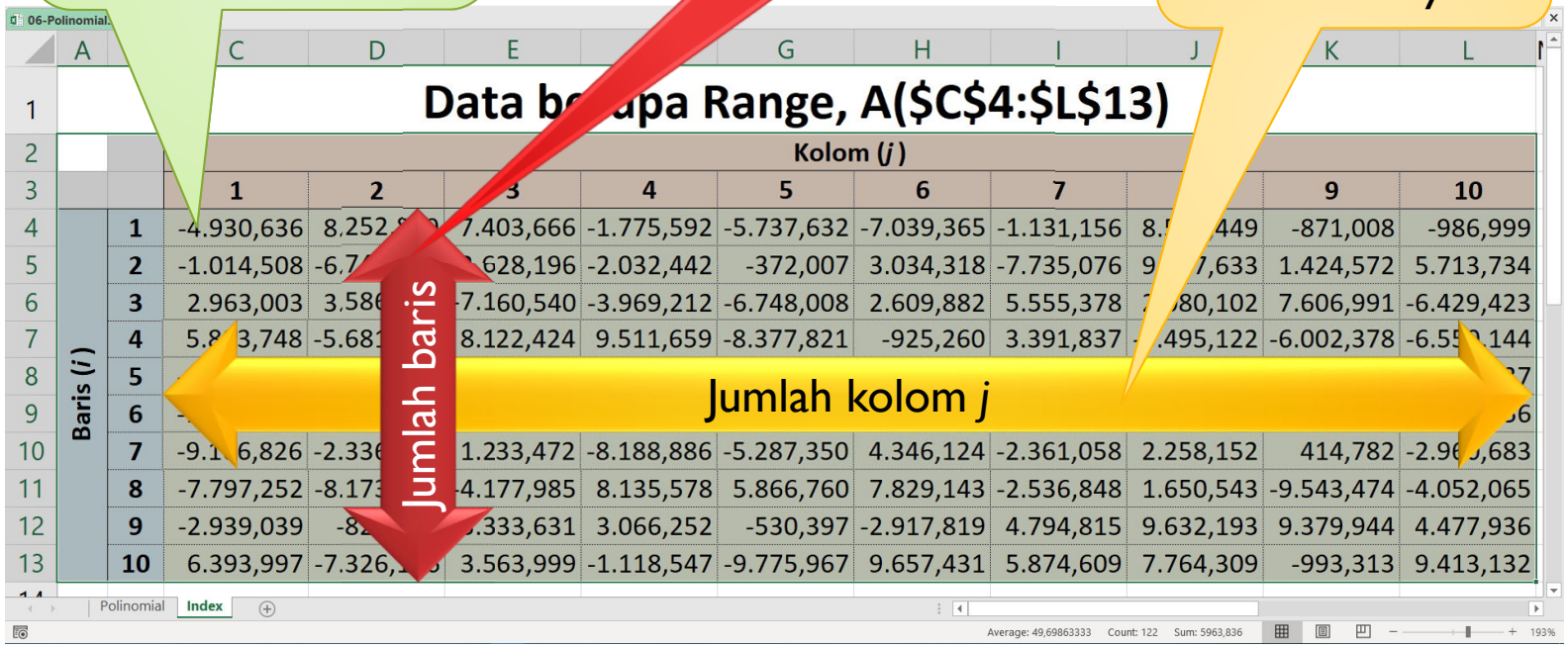
MENGGUNAKAN VBA-EXCEL INDEX UNTUK AKSES VARIABEL RANGE

Index pada variable Range *A*($C$4:$L$13)

Index *A*(**1,1**) selalu mengacu kepada sel pertama dari Range, dalam hal ini $C$4

Jumlah baris (*i*) harus dihitung sebelumnya

Jumlah kolom (*j*) harus dihitung sebelumnya



Dalam menggunakan index untuk Range, perlu diperhatikan: (1) Sel pertama yang diacu, (2) jumlah baris dan kolom. Di luar Kawasan Range masih dapat diacu, namun nilainya berisi "sampah" yang mungkin tidak mempunyai arti.

Index pada variable Range *Nilai*($C$4:$L$13)

Jumlah kolom (*j*) harus dihitung sebelumnya

Index *Nilai* (**1,1**) selalu mengacu kepada sel pertama dari Range, dalam hal ini $C$4

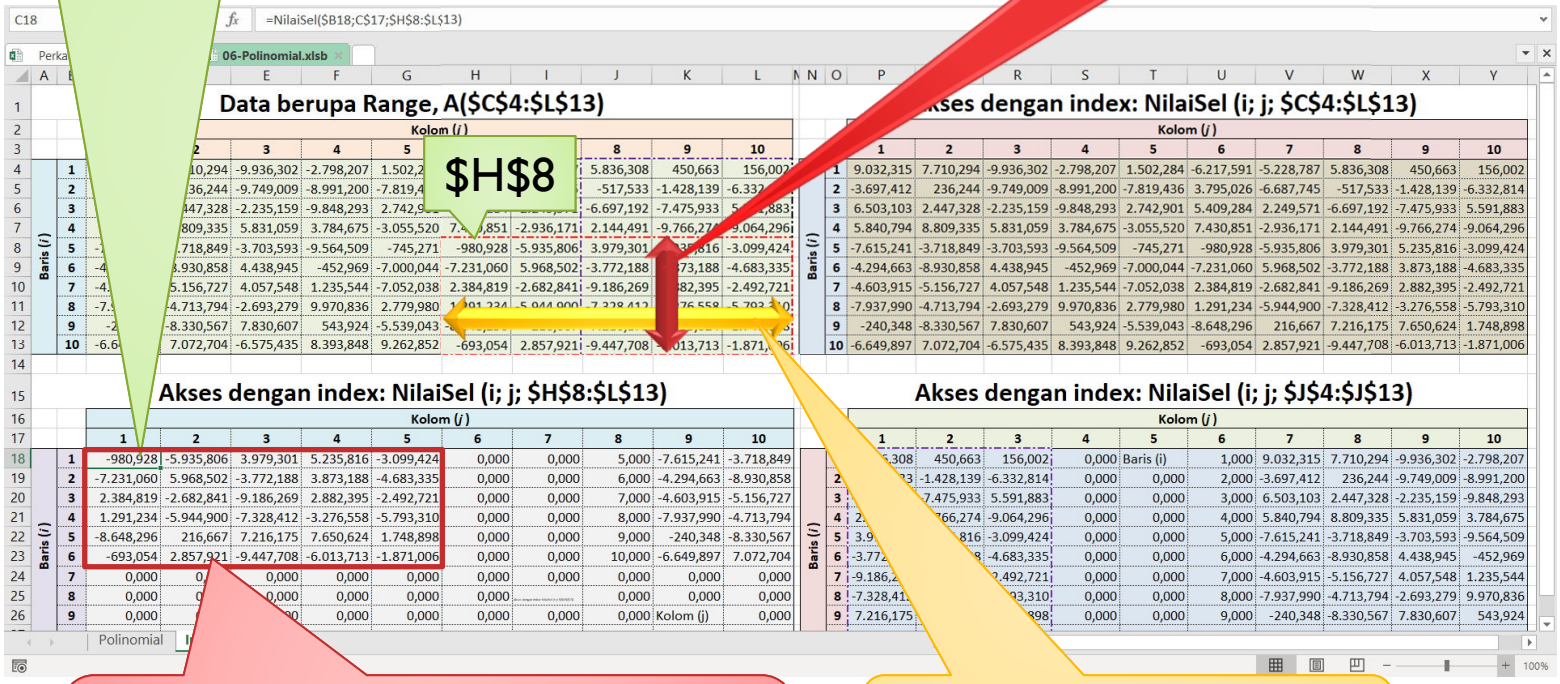
$C$4

Dalam menggunakan index untuk Range, perlu diperhatikan: (1) Sel pertama yang diacu, (2) jumlah baris dan kolom. Di luar Kawasan Range masih dapat diacu, namun nilainya berisi "sampah" yang mungkin tidak mempunyai arti.

Index pada variable Range *Nilai*($H$8:$L$13)

Index *Nilai* (**1,1**) selalu mengacu kepada sel pertama dari Range, dalam hal ini $H$8

Jumlah baris (*i*) harus dihitung sebelumnya



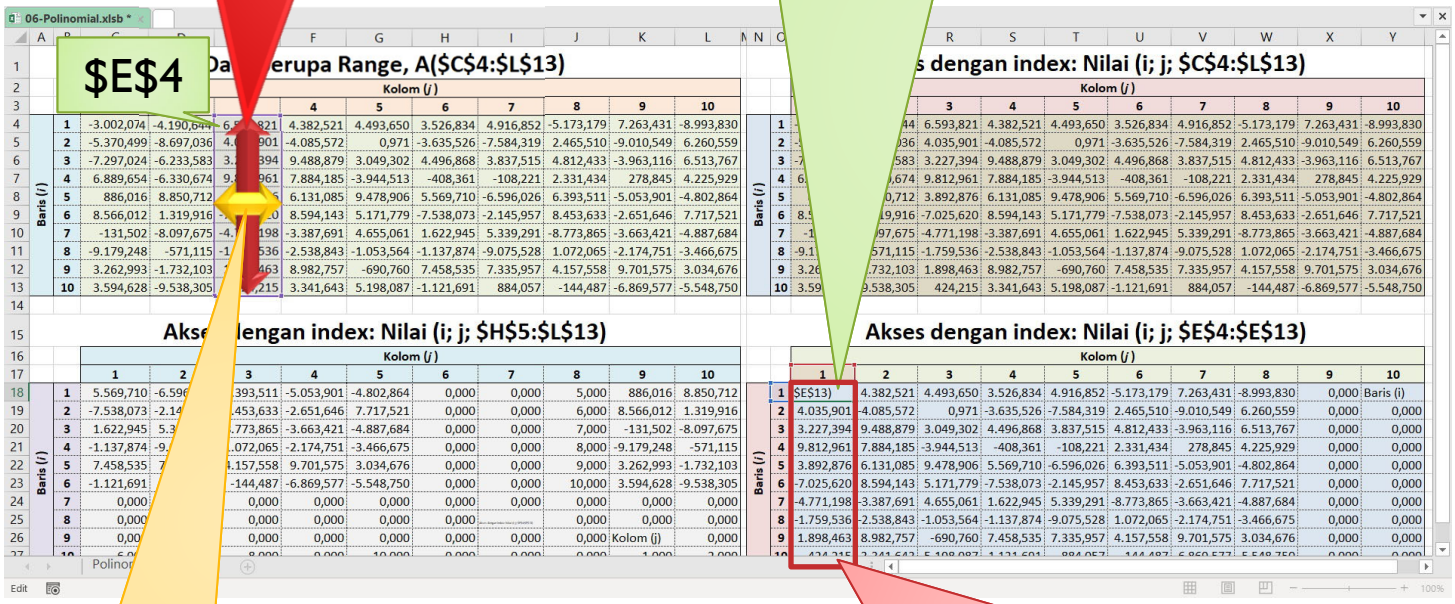
Hasil yang valid hanya dalam kawasan ini (sesuai ukuran data), yang lainnya sampah!

Jumlah kolom (*j*) harus dihitung sebelumnya

Index pada variable Range *Nilai*($E$4:$L$13)

Jumlah baris (*i*) harus dihitung sebelumnya

Index *Nilai* (**1,1**) selalu mengacu kepada sel pertama dari Range, dalam hal ini $E$4



oleh
Djoko Luknanto
Departemen Teknik Sipil dan Lingkungan
Fakultas Teknik Universitas Gadjah Mada

BEBERAPA JENIS VBA-EXCEL DO LOOP

Do...Loop statement (1/2)

Syntax

- **Do** [{ **While** | **Until** } *condition*]
  [*statements*]
  [**Exit Do**]
  [*statements*]
  Loop

While | **Until** di depan

Or, you can use this syntax:

- **Do**
  [*statements*]
  [**Exit Do**]
  [*statements*]
  Loop [{ **While** | **Until** } *condition*]

While | **Until** di belakang

Do...Loop statement (2/2)

While | Until di depan

```
Public Sub LoopExample()
    Dim Check As Boolean, Counter As Long, Total As Long
    Check = True: Counter = 0: Total = 0 ' Initialize variables.
    Do ' Outer loop.
        Do While Counter < 20 ' Inner Loop
            Counter = Counter + 1 ' Increment Counter.
            If Counter Mod 10 = 0 Then ' Check in with the user on every multiple of 10.
                Check = (MsgBox("Keep going?", vbYesNo) = vbYes) ' Stop when user click's on No
                If Not Check Then Exit Do ' Exit inner loop.
            End If
        Loop
        Total = Total + Counter ' Exit Do Lands here.
        Counter = 0
    Loop Until Check = False ' Exit outer loop immediately.
    MsgBox "Counted to: " & Total
End Sub
```

While | Until di belakang