

IF1191

Pemrograman Dalam JavaScript

Dwi Aji Mardiyanto

Departemen Teknik Informatika
2004



Materi Pertemuan 2

- ☞ Memahami aspek pemrograman
- ☞ Variabel, Literal, Ekspresi, Operator
- ☞ Statement/Pernyataan
- ☞ Pencabangan dan looping
- ☞ Fungsi



Pemrograman

- ☞ Pemrograman = Membuat resep ??
- ☞ Pemrograman = belajar bahasa asing ??



11/17/2005

IF1191/P2/

3

```
PLEASE DO ,1 <- #13
DO ,1 SUB #1 <- #238
DO ,1 SUB #2 <- #112
DO ,1 SUB #3 <- #112
DO ,1 SUB #4 <- #0
DO ,1 SUB #5 <- #64
DO ,1 SUB #6 <- #238
DO ,1 SUB #7 <- #26
DO ,1 SUB #8 <- #248
DO ,1 SUB #9 <- #168
DO ,1 SUB #10 <- #24
DO ,1 SUB #11 <- #16
DO ,1 SUB #12 <- #158
DO ,1 SUB #13 <- #52
PLEASE READ OUT ,1
PLEASE GIVE UP
```

Intercal

```
>v"Hello world!"0<
,:
^_25*,@
```

Befunge

```
(DEFUN HELLO-WORLD ()
  (PRINT (LIST 'HELLO 'WORLD)))
```

LISP/Scheme

```
#include <iostream>

int main()
{
  std::cout << "Hello, world!";
}
```

C++

```
print "Hello, World!";
```

Perl

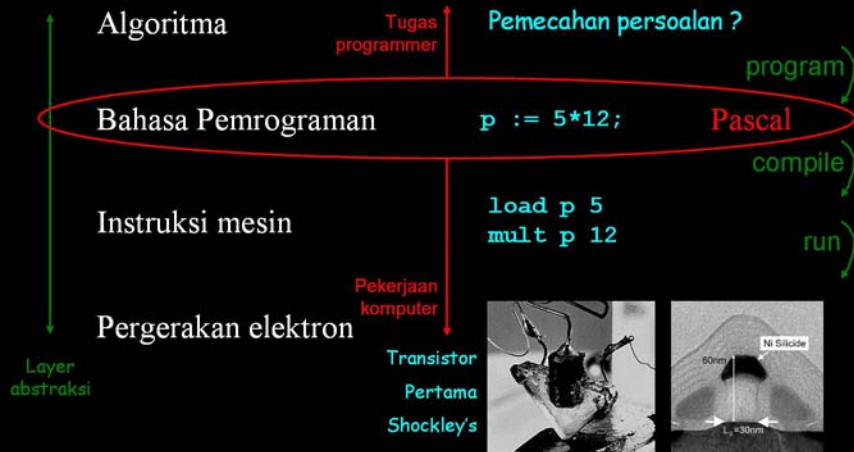
11/17/2005

IF1191/P2/

4

Membuat Program

Bagaimana mesin berpikir ?



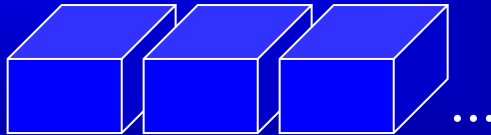
11/17/2005

IF1191/P2/

5

Data Program ?

Tempat Penyimpanan



Area Kerja



11/17/2005

IF1191/P2/

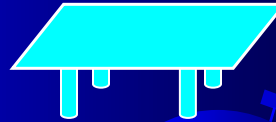
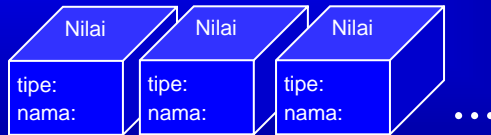
6



Variabel ~ “Kotak penyimpanan”

Variabel == Tas (or Kotak Penyimpanan)

Setiap variabel mempunyai 2 atribut: $\begin{cases} \text{tipe} \\ \text{nama} \end{cases}$



Setiap variabel mungkin saja tidak memiliki nilai

Nilai adalah isi kotak tempat penyimpanan

11/17/2005

IF1191/P2/

7

Deklarasi vs Definisi



```
String s;  
String my;
```

} Mendeklarasikan var.

```
s = "hark... a shark!";  
my = "t";
```

} Mendefinisikan var.

```
s = "ha";  
s = my;
```

} Me-redefinisi var.

11/17/2005

IF1191/P2/

8

Literal (1/2)

☞ Literal digunakan utk merepresentasikan nilai tetap dalam JavaScript yang bukan variabel.

☞ Integer

- **Decimal:** basis 10, tanpa diawali 0, contoh: 4, 10
- **Octal:** basis 8, diawali 0, dari 0-7, contoh: 04, 010
- **Hexadecimal:** basis 16, diawali 0x atau 0X, dari 0-9 dan a-f atau A-F, contoh: 0xff, 0X501, 0XBC

☞ Float (Bil Real)

- **Decimal integer**
- **Decimal point** (“.”)
- **Fraction** (angka desimal lain)
- **Exponent** (“e” atau “E”)
- **Type suffix** (“+” atau “-”)
- Minimal 1 digit + [.,e|E]
- Contoh: 3.145, -3.1E12, .1e12, 2E-12

☞ Boolean

- True
- False

11/17/2005

IF1191/P2/

9

Literal (2/2)

Kar	Arti
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage Return
\t	Tab
\\	Backslash
\xFF	Next char bagian string

☞ String

- Nol karakter atau lebih yang diapit oleh tanda petik dua (“”) atau petik satu (’)
- Tanda petik pengapit harus sama
- Contoh: “blah”, ‘blah’, “1234”, “pindah \n baris”

11/17/2005

IF1191/P2/

10

Ekspresi & Statement

String message = "literal string";

```
if( X == 94 )  
{  
  message = "Nilai < 100";  
}
```

Statement

Ekspresi

11/17/2005

IF1191/P2/

11

Assignment Operator

$x = y$, berarti meng-*assign* nilai y ke x

Shorthand	Arti
$x += y$	$x = x + y$
$x -= y$	$x = x - y$
$x *= y$	$x = x * y$
$x /= y$	$x = x / y$
$x \% = y$	$x = x \% y$
$x << = y$	$x = x << y$

Shorthand	Arti
$x >> = y$	$x = x >> y$
$x >>> = y$	$x = x >>> y$
$x \& = y$	$x = x \& y$
$x \wedge = y$	$x = x \wedge y$
$x = y$	$x = x y$

11/17/2005

IF1191/P2/

12

Comparison Operator

Operator	Description	Example
Equal (==)	Returns true if the operands are equal.	$x == y$ returns true if x equals y .
Not equal (!=)	Returns true if the operands are not equal.	$x != y$ returns true if x is not equal to y .
Greater than (>)	Returns true if left operand is greater than right operand.	$x > y$ returns true if x is greater than y .
Greater than or equal (>=)	Returns true if left operand is greater than or equal to right operand.	$x >= y$ returns true if x is greater than or equal to y .
Less than (<)	Returns true if left operand is less than right operand.	$x < y$ returns true if x is less than y .
Less than or equal (<=)	Returns true if left operand is less than or equal to right operand.	$x <= y$ returns true if x is less than or equal to y .

11/17/2005

IF1191/P2/

13

Operator Aritmetika

- ☞ Operator standard: +, -, *, /
- ☞ **Modulus (%)**: var1 \% var2 ,
 - contoh: $12 \% 5$ menghasilkan 2.
- ☞ **Increment (++)**: var++ atau ++var ,
 - contoh jika $x=3$, maka $y=x++$ menyebabkan $y=3$ dan $x=4$, sedang $y=++x$ menyebabkan $x=4$, dan $y=4$
- ☞ **Decrement (--)**: var-- atau -var ,
 - contoh jika $x=3$, maka $y=x--$ menyebabkan $y=3$ dan $x=2$, sedang $y=--x$ menyebabkan $x=2$ dan $y=2$
- ☞ **Unary negation (-)**,
 - contoh $x=-x$, jika $x=3$ maka x menjadi -3

11/17/2005

IF1191/P2/

14

Operator Logika

Tahun apa yang sesuai dengan pernyataan ?

```
if ( tahun%4 == 0 && tahun > 1776 )
```

↑ “dan”

```
if ( tahun == 0 || tahun%4 == 2 )
```

↑ “atau”

```
if ( tahun%20 != 0 )
```

“bukan” “tidak sama dengan”

```
if ( !(tahun%20 == 0) )
```

11/17/2005

IF1191/P2/

15



Conditional Statement (1/4)

☞ **Conditional statement** adalah statemen untuk menampilkan hasil yang berbeda pada kondisi yang berbeda

☞ **Skema conditional statement:**

```
if (condition) {  
    statements1  
[ }  
else {  
    statements2 ]  
}
```

11/17/2005

IF1191/P2/

16



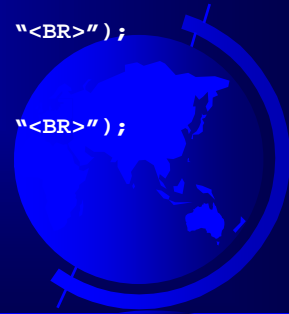
Conditional Statement (2/4)

☞ Satu kasus

```
i = 1;
if (i>0) {
    document.write(i + "positif" + "<BR>");
}
```

☞ Dua kasus

```
i = 1;
if (i>0) {
    document.write(i + "positif" + "<BR>");
}
else {
    document.write(i + "negatif" + "<BR>");
}
```



Conditional Statement (3/4)

☞ Banyak kasus

```
i = 1;
if (i>0) {
    document.write(i + " positif " + "<BR>");
}
else {
    if (i==0) {
        document.write(i + " nol " + "<BR>");
    }
    else {
        document.write(i + " negatif " + "<BR>");
    }
}
```



Conditional Statement (4/4)

- ☞ **Satu kasus:** digunakan untuk penanganan satu kondisi saja, sisanya diabaikan
- ☞ **Dua kasus:** digunakan untuk penanganan dua kondisi.
- ☞ **Banyak kasus:** digunakan untuk penanganan beberapa kondisi. Kita dapat membuat conditional statement bersarang dengan cara membuat conditional statement pada statemen else
- ☞ Statement 1 dan 2 dapat berupa statement apapun, termasuk conditional statement
- ☞ Statement dibatasi dengan **kurung kurawal {}**

11/17/2005

IF1191/P2/

19

Loop Statement

- ☞ **Loop** adalah himpunan perintah yang dieksekusi secara berulang sampai kondisi tertentu dipenuhi.
- ☞ **Loop statement** digunakan untuk menampilkan hasil-hasil secara berulang sampai suatu kondisi dipenuhi. Contoh: counter
- ☞ Loop statement di JavaScript:
 - for
 - while
 - do-while
 - break
 - continue

11/17/2005

IF1191/P2/

20

for statement

- ☞ **for statement** terdiri atas 3 bagian yang dipisahkan oleh ;
 - **initial-expression**: berisi nilai awal, bersifat opsional, dan bisa berisi satu assignment atau lebih
 - **condition**: berisi kondisi yang harus dipenuhi, dan harus ada. **condition** menentukan berlangsungnya loop, yang akan berhenti jika **condition** bernilai **false**
 - **increment-expression**: berisi ekspresi yang akan mengubah nilai, bersifat opsional, dan bisa berisi satu assignment atau lebih
- ☞ Skema **for statement** adalah sbb.

```
for ([initial-expression]; [condition];  
    [increment-expression]) {  
    statements  
}
```

Contoh for statement

```
for (i=1;i<=10;i++) {  
    document.write("For " + i + "<BR>");  
}  
  
for (;i<=10;) {  
    document.write("For " + i + "<BR>");  
}
```

Pada contoh kedua, tidak terdapat nilai awal dan ekspresi inkremen yang akan mengubah nilai kondisi. Nilai kondisi akan selalu true dan loop selalu berulang. Kondisi ini disebut sebagai *infinite loop*.

while statement

☞ **while** statement mengulang loop selama kondisi tertentu bernilai **true**. Jika kondisi menjadi **false**, statement dalam loop akan berhenti dieksekusi dan melanjutkan ke statement di bawah loop.

☞ Skema **while** statement adalah sbb

```
while (condition) {  
    statements  
}
```

☞ Penggunaan skema **while** akan mengeksekusi statements jika pengujian kondisi bernilai true. Jika false maka loop tidak akan terjadi dan akan menjalankan statement selanjutnya yang berada di luar loop while.

11/17/2005

IF1191/P2/

23

Contoh while statement

```
☞ i = 1;  
   while (i<=10) {  
       document.write("While " + i +  
       "<BR>");  
       i++;  
   }  
☞ while (true) {  
   alert("Hello, world")  
}
```

11/17/2005

IF1191/P2/

24

do-while statement

☞ **do-while** statement merupakan pengulangan yang akan dijalankan selama kondisi true.

☞ Skema **do-while** statement

```
do {  
    statements  
} while (repeat condition test)
```



Contoh do-while statement

```
i=10  
do {  
    document.write("do-while " + i + "<BR>");  
    i--  
} while (i>0)
```



break statement

- ☞ **break** statement menghentikan loop **while** atau **for** yang sedang berjalan, dan memindahkan kendali program ke statement di bawah loop yang dihentikan.
- ☞ Skema **break** statement adalah sbb.
break



11/17/2005

IF1191/P2/

27

Contoh penggunaan break

- ☞ **break** statement di bawah ini akan menghentikan **while** loop jika $i = 3$

```
var i = 0
while (i < 6) {
    document.write("break " + i + "<BR>");
    if (i == 3)
        break;
    i++;
}
```



11/17/2005

IF1191/P2/

28

continue statement

- ☞ **continue** statement menghentikan eksekusi blok statement dalam loop **while** atau **for** dan melanjutkan eksekusi loop untuk iterasi selanjutnya.
- ☞ Skema **continue** statement adalah sbb
`continue`
- ☞ Tidak seperti **break** statement, **continue** tidak menghentikan keseluruhan loop, tetapi
 - pada **while** loop, mengembalikan ke condition
 - pada **for** loop, mengembalikan ke increment-expression

11/17/2005

IF1191/P2/

29

Contoh continue

```
i = 0
n = 0
while (i < 5) {
  i++
  document.write("continue
" +i+ "<BR>");
  if (i == 3)
    continue
  n += i
  document.write("n= "+n+
"<BR>");
}
```

- ☞ continue 1
n= 1
- ☞ continue 2
n= 3
- ☞ continue 3
n= 7
- ☞ continue 4
n= 12

11/17/2005

IF1191/P2/

30

Baris Komentar

- ☞ **comments** adalah notasi yang dibuat oleh penulis untuk menjelaskan apa yang dikerjakan oleh script. **comments** diabaikan oleh interpreter.
- ☞ JavaScript mendukung Java-style comments:
 - Comments untuk baris tunggal (diawali //)
 - Comments untuk beberapa baris (diawali /* dan diakhiri dengan */)



11/17/2005

IF1191/P2/

31

Fungsi

```
// Program Utama  
Z= FungsiA();  
FungsiB(a,b,c);
```

```
function FungsiA(){  
....  
x=....  
return x;  
}
```

```
function FungsiB(int a,int b, int c)  
{  
.....  
}
```

11/17/2005

IF1191/P2/

32

Contoh Fungsi

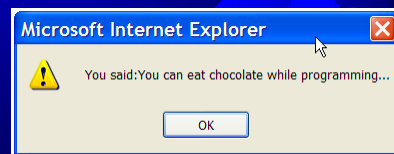
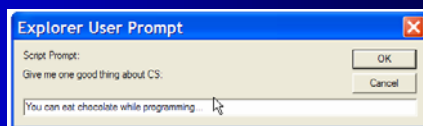
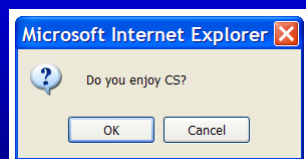
```
function check()
{
  var agree = false;
  agree = confirm('Do you enjoy CS?');
  if (agree)
    notes=prompt("Give me one good thing about CS:");
  if (! agree)
    notes=prompt("Why don't you like CS?");
  alert("You said:"+notes);
}
```

11/17/2005

IF1191/P2/

33

Hasil



11/17/2005

IF1191/P2/

34