

## Pertemuan6

### *Percabangan & Perulangan pada Ruby*

#### **Objektif:**

1. Mahasiswa mengetahui dan memahami percabangan dan perulangan pada Ruby
2. Mahasiswa mengetahui dan memahami bentuk umum dari percabangan dan perulangan pada Ruby
3. Mahasiswa dapat membuat program sederhana untuk percabangan dan perulangan dengan menggunakan bahasa pemrograman Ruby

## P6.1 Teori

### Percabangan

Dalam pemrograman, kita tentunya mengenal istilah seleksi dan perulangan. Pada umumnya dalam membuat program, selalu ada seleksi dimana diperlukan pengecekan suatu kondisi untuk mengarahkan program agar berjalan sesuai keinginan. Pada Ruby untuk melakukan suatu pengecekan kondisi, terdapat tiga macam statemen. Antara lain :

- Perintah *if*
- Perintah *if – else*
- Perintah *if – else – elsif*

### Perintah If

Statemen *if* digunakan untuk melakukan penyeleksian dimana jika kondisi bernilai benar maka program akan mengeksekusi statemen dibawahnya. Dalam ruby, setelah penulisan statement di akhiri dengan *end*.

Bentuk umum perintah *if* :

```
if kondisi
  statemen
end
```

Contoh Program :

```
irb(main):001:0> a = "Ruby"
=> "Ruby"
irb(main):002:0> if a == "Ruby"
irb(main):003:1> puts "Selamat Belajar Ruby"
irb(main):004:1> end
Selamat Belajar Ruby
=> nil
irb(main):005:0> I
```

### Perintah If – Else

Statemen *if – else* digunakan untuk melakukan penyeleksian kondisi dimana jika kondisi bernilai benar maka program akan mengeksekusi statemen 1. Namun, jika nilai kondisi bernilai salah maka statemen 2 yang akan dieksekusi.

Bentuk umum perintah *if – else* :

```
if kondisi
  statemen 1
else
  statemen 2
end
```

Contoh Program :

```
irb(main):005:0> kunci = "Ruby"
=> "Ruby"
irb(main):006:0> print "Masukkan Pass word : "; pass word = gets
Masukkan Pass word : Afikaaa
=> "Afikaaa\n"
irb(main):007:0> if pass word == kunci
irb(main):008:1> print "Pass word Benar"
irb(main):009:1> else
irb(main):010:1* print "Pass word salah"
irb(main):011:1> end
Pass word salah=> nil
irb(main):012:0>
```

### Perintah If – Else - Elself

Statemen *if – else - elsif* digunakan untuk melakukan penyeleksian kondisi dimana kondisi yang diberikan lebih dari 1 kondisi atau memiliki beberapa kondisi. Jika kondisi pertama bernilai benar maka lakukan seleksi kondisi ke-dua dan seterusnya.

Bentuk umum perintah *if – else – elsif* :

```
if kondisi1
  statemen
elsif kondisi2
  statemen
else
  statemen
end
```

Contoh Program :

```
irb(main):001:0> print "Masukkan bilangan : "; bil = gets
Masukkan bilangan : 135
=> "135\n"
irb(main):002:0> bil.chop!
=> "135"
irb(main):003:0> if bil.to_i <=999 && bil.to_i >=100
irb(main):004:1> print "Bilangan ratusan"
irb(main):005:1> elsif bil.to_i <=99 && bil.to_i >=10
irb(main):006:1> print "Bilangan puluhan"
irb(main):007:1> else
irb(main):008:1* print "Bilangan satuan"
irb(main):009:1> end
Bilangan ratusan=> nil
irb(main):010:0> I
```

### Perulangan

Perintah perulangan di gunakan untuk mengulang pengeksekusian statemen-statemen hingga berkali-kali sesuai dengan iterasi yang diinginkan. Dalam Ruby, perintah untuk perulangan (loop) adalah *times*, *while* dan *for*.

## Perintah *Times*

Perintah *times* digunakan untuk perulangan yang sangat sederhana. Hanya dengan bentuk umum seperti di bawah ini kita bisa langsung mendapatkan output berulang.

Bentuk umumnya adalah :

```
Jumlah_Perulangan.times do
    statemen
end
```

Contoh Program :

```
irb(main):010:0> 3.times {print "Hi "}
Hi Hi Hi => 3
```

atau

```
irb(main):011:0> 3.times do
irb(main):012:1* print "Hi "
irb(main):013:1> end
Hi Hi Hi => 3
irb(main):014:0>]
```

Pada perulangan *times*, secara otomatis kita melakukan pertambahan ke atas apabila ingin melakukan perulangan ke bawah dapat menggunakan sebuah *variable counter*.

Contoh program pertambahan ke bawah adalah sebagai berikut :

```
irb(main):014:0> count = 10
=> 10
irb(main):015:0> 10.times do
irb(main):016:1* count -=1
irb(main):017:1> puts count
irb(main):018:1> end
9
8
7
6
5
4
3
2
1
0
=> 10
irb(main):019:0>]
```

Yang disebut *variable counter* di program di atas adalah variable **count**.

## Perintah *While*

Perintah *while* pada Ruby merupakan perintah yang paling umum digunakan untuk proses perulangan. Konsep sederhana dari perintah *while* adalah *statement* yang terdapat di dalam *while* akan diulang pengeksekusiannya selama nilai kondisinya benar. Dan ia akan keluar atau tidak melakukan eksekusi blok statemen jika nilai kondisinya salah.

Bentuk umum statemen *while*,

```
while kondisi
  statemen
end
```

Contoh penggunaan *while* :

---

```
irb(main):001:0> i = 1
=> 1
irb(main):002:0> while i<= 5
irb(main):003:1> printf "%d ", i
irb(main):004:1> i = i + 1
irb(main):005:1> end
1 2 3 4 5 => nil
irb(main):006:0>
```

## Perintah *For*

Perintah *for* dalam Ruby sama seperti bahasa pemrograman lainnya. Pada perulangan *for* ini lebih spesifik dari pada perintah *while*. Kita dapat menentukan *range* (batasan) pada perulangan ini.

Bentuk umum statemen *for*,

```
For variable in range
  statemen
end
```

Contoh penggunaan *for* :

```

irb(main):006:0> for i in 1..5
irb(main):007:1> puts i.to_s
irb(main):008:1> end
1
2
3
4
5
=> 1..5
irb(main):009:0>

```

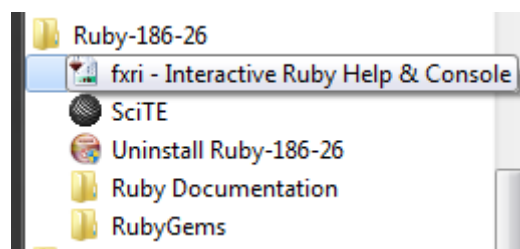
## P6.2 Contoh Kasus

### Contoh Kasus 1

Buat program untuk memasukkan nama, NPM, nilai UTS dan nilai UAS, yang memiliki perhitungan nilai rata-rata (dari nilai UTS dan UAS yang dijumlahkan kemudian dibagi dua) dan kondisi nilai akhir berupa huruf dari range nilai rata-rata yang dihasilkan (nilai A s/d E).

#### Langkah 1.

Buka fxri – Inteactive Ruby Help & Console yang ada di menu start => All Program => Ruby-186-26.



#### Langkah 2.

Ketik listing program berikut :

```

irb(main):001:0> print "Masukkan Nama Anda : "; nama = gets
Masukkan Nama Anda : Amanina Afika
=> "Amanina Afika\n"
irb(main):002:0> print "Masukkan NPM Anda : "; npm = gets
Masukkan NPM Anda : 50411123
=> "50411123\n"
irb(main):003:0> print "Masukkan Nilai UTS Anda : "; uts = gets.to_i
Masukkan Nilai UTS Anda : 76
=> 76
irb(main):004:0> print "Masukkan Nilai UAS Anda : "; uas = gets.to_i
Masukkan Nilai UAS Anda : 84
=> 84
irb(main):005:0> akhir = (uts+uas) / 2
=> 80
irb(main):006:0> if akhir.to_i >= 85
\irb(main):008:1* puts "Anda mendapat Grade A"
irb(main):009:1> elsif akhir.to_i >= 75
irb(main):010:1> puts "Anda mendapat Grade B"
irb(main):011:1> elsif akhir.to_i >= 65
irb(main):012:1> puts "Anda mendapat Grade C"
irb(main):013:1> elsif akhir.to_i >= 55
irb(main):014:1> puts "Anda mendapat Grade D"
irb(main):015:1> else
irb(main):016:1* puts "Anda mendapat Grade E"
irb(main):017:1> end
Anda mendapat Grade B
=> nil
irb(main):018:0>I

```

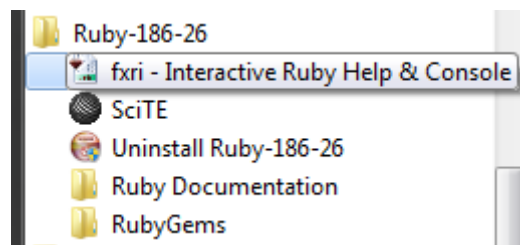
Setelah selesai, maka output dari program akan langsung muncul seperti gambar di atas.

### Contoh Kasus 2

Buat program dengan output segitiga sama kaki dengan tinggi segitiga = 5. Gunakan perintah while!

#### Langkah 1.

Buka fxri – Interactive Ruby Help & Console yang ada di menu start => All Program  
=> Ruby-186-26.



#### Langkah 2.

Ketik listing program berikut :

```

irb(main):019:0> x = 5
=> 5
irb(main):020:0> temp = x
=> 5
irb(main):021:0> i = 1
=> 1
irb(main):022:0> while i <= x
irb(main):023:1>   k = 1
irb(main):024:1>   while temp >= 1
irb(main):025:2>     print " "
irb(main):026:2>     temp -= 1
irb(main):027:2>   end
irb(main):028:1>   while k <= i
irb(main):029:2>     print "*"
irb(main):030:2>     k += 1
irb(main):031:2>   end
irb(main):032:1>   temp = x - i
irb(main):033:1>   i += 1
irb(main):034:1>   puts ""
irb(main):035:1> end
  *
  **
 ***
****
*****
=> nil
irb(main):036:0>

```

Setelah selesai, maka output dari program akan langsung muncul seperti gambar di atas.

## P6.3 Latihan

### Latihan 1

Buat program bilangan ganjil dan genap yang menghasilkan output sebagai berikut :

```

1 Ganjil
2 Genap
3 Ganjil
4 Genap
5 Ganjil
=> 1..5
irb(main):017:0>

```

Jawaban :

```

irb(main):010:0> for i in 1..5
irb(main):011:1>   if i%2 == 1
irb(main):012:2>     puts i.to_s + " Ganjil"
irb(main):013:2>   else
irb(main):014:2>     puts i.to_s + " Genap"
irb(main):015:2>   end
irb(main):016:1> end

```



## Latihan 2

Buat persegi panjang simbol '\$' dengan menggunakan perintah perulangan for dimana user dapat memasukkan jumlah panjang dan lebar dari persegi panjang tersebut!

Jawaban :

```
irb(main):060:0> print "Masukkan panjang Persegi Panjang : "; p = gets.to_i
Masukkan panjang Persegi Panjang : 7
=> 7
irb(main):061:0> print "Masukkan lebar Persegi panjang : "; l = gets.to_i
Masukkan lebar Persegi panjang : 3
=> 3
irb(main):062:0> for i in 1..l
irb(main):063:1>   for j in 1..p
irb(main):064:2>     print "$"
irb(main):065:2>   end
irb(main):066:1> puts ""
irb(main):067:1> end
$$$$$$$
$$$$$$$
$$$$$$$
=> 1..3
irb(main):068:0>
```

## Latihan 3

Buat piramida simbol '\*' dengan menggunakan perintah perulangan while dimana user dapat memasukkan tinggi piramida tersebut!

Jawaban :

```
irb(main):001:0> print "Masukkan tinggi Piramida : "; t = gets.to_i
Masukkan tinggi Piramida : 7
=> 7
irb(main):002:0> temp = t
=> 7
irb(main):003:0> i = 1
=> 1
irb(main):004:0> while i <= t
irb(main):005:1>   k = 1
irb(main):006:1>   while temp >= 1
irb(main):007:2>     print " "
irb(main):008:2>     temp -= 1
irb(main):009:2>   end
irb(main):010:1>   while k <= i
irb(main):011:2>     print "*"
irb(main):012:2>     k += 1
irb(main):013:2>   end
irb(main):014:1>   temp = t - i
irb(main):015:1>   i += 1
irb(main):016:1>   puts ""
irb(main):017:1> end
*
**
***
****
*****
*****
=> nil
irb(main):018:0>I
```

## **P6.4 DaftarPustaka**

- <http://codycoding.wordpress.com/2010/10/25/percabangan-dan-perulangan-pada-ruby/>, 17 Maret 2012
- <http://wartawarga.gunadarma.ac.id/2010/04/perulangan-dan-percabangan-pada-ruby/>, 18 Maret 2012
- <http://dudu.web.id/2010/10/belajar-ruby-perulangan-while/>, 19 Maret 2012