

Pertemuan7

Array, Hash, Range dan Function pada Ruby

Objektif:

1. Mahasiswa mengetahui dan memahami array dan fungsi pada Ruby
2. Mahasiswa mengetahui dan memahami bentuk umum dari array dalam Ruby
3. Mahasiswa membuat program sederhana untuk array dan fungsi menggunakan bahasa pemrograman Ruby

P7.1 Teori

Array

Tipe data Array di Ruby hampir sama dengan tipe data array pada bahasa pemrograman lain umumnya. Tipe data Array di Ruby sama dengan tipe data list di Python. Untuk membuat suatu objek Array baru, Kita dapat menggunakan *literal* Array [] ataupun konstruktor Array.new. Isi array dapat berupa objek apapun baik angka, string, maupun objek lain termasuk objek array itu sendiri.

Array pada Ruby adalah sekelompok data yang disimpan ke dalam variabel dengan nama yang sama, dengan memberi indeks pada variabel untuk membedakan antara yang satu dengan yang lain. Array pada Ruby ditulis dalam tanda kurung siku ([]) dan antara elemen didalamnya ditulis dengan pemisah berupa koma.

Bentuk Umum:

nama_variabel = [elemen array]

contoh:

```
A = [1, 2, "adet", 3]
```

Elemen pada array dapat diakses dengan menuliskan nama_variabel[indeks]. Indeks pada Array dimulai dari nol.

Pada contoh diatas elemen array A memiliki indeks seperti berikut:

```
1 = [0]
```

```
2 = [1]
```

```
"adet" = [2]
```

```
3 = [3]
```

Jadi untuk mengakses elemen 1 yang harus kita lakukan adalah menuliskan A[0].

Untuk mengakses elemen objek Array ini, kita menggunakan indeks di mana indeks awal dimulai dari 0 sampai panjang array – 1. Kita juga dapat menggunakan indeks negatif di mana indeks -1 berarti elemen terakhir, -2 berarti elemen kedua terakhir, dan seterusnya.

Apabila kita mencoba mengakses elemen Array pada indeks yang di luar jangkauan, pernyataan tersebut tidak akan menyebabkan *error* seperti IndexError : list index out of range di Python, Ruby akan mengembalikan objek nil. Untuk mengubah isi array ataupun

menambahkan data pada objek Array dengan indeks tertentu, kita bisa menggunakan *statement* biasa seperti `array[indeks] = nilai baru`. Apabila kita ingin membuat objek Array yang berisi string, akan repot sekali untuk menambahkan tanda kutip pada setiap elemen Array. Hal tersebut dapat diatasi dengan menggunakan prefiks `%w`. Untuk jelasnya, lihat contoh berikut:

```
>> a = []
=> []
>> a = [1, 2, "m", [3, 4, 5]]
=> [1, 2, "m", [3, 4, 5]]
>> a[0]
=> 1
>> a[3]
=> [3, 4, 5]
>> a[3][1]
=> 4
>> a[-1]
=> [3, 4, 5]
>> a[6]
=> nil
>> a[1] = 88
=> 88
>> a
=> [1, 88, "m", [3, 4, 5]]
>> c = %w(Anto, Eric, Isabel)
=> ["Anto,", "Eric,", "Isabel"]
```

Berikut adalah beberapa metode pada kelas Array : **(contoh ini dijalankan pada irb)**

`Array#push`

untuk menambah data pada array. Data terbaru akan ditempatkan paling akhir.

```
>> data = %W[ape bee cat dog]
=> ["ape", "bee", "cat", "dog"]
>> data = data.push("eel", "fox")
=> ["ape", "bee", "cat", "dog", "eel", "fox"]
```

`Array#pop`

mengembalikan data pada indeks terakhir array dan menghapus data pada indeks terakhir.

```
>> data = %W[ape bee cat dog]
=> ["ape", "bee", "cat", "dog"]
>> animal = data.pop()
=> "dog"
>> data
=> ["ape", "bee", "cat"]
```

Array#unshift

Menambahkan sebuah elemen ke awal array.

```
>> data = %W[ape bee cat dog]
=> ["ape", "bee", "cat", "dog"]
>> data.unshift("eel", "fox")
=> ["eel", "fox", "ape", "bee", "cat", "dog"]
```

Array#shift

Menghapus elemen pertama dalam array.

```
>> data = %W[ape bee cat dog]
=> ["ape", "bee", "cat", "dog"]
>> animal = data.shift()
=> "ape"
>> data
=> ["bee", "cat", "dog"]
```

Array#index

Menemukan posisi item dalam array. Memberikan nilai nihil jika item tersebut tidak ada dalam array.

```
>> data = %W[ape bee cat dog]
=> ["ape", "bee", "cat", "dog"]
>> data.index("bee")
=> 1
>> data.index("dog")
=> 3
>> data.index("elk")
=> nil
```

Array#include?

Memberitahukan apakah ada atau tidak ada item tertentu pada array.

```
>> data = %W[ape bee cat dog]
=> ["ape", "bee", "cat", "dog"]
>> data.include?("ape")
=> true
>> data.include?("cat")
=> true
>> data.include?("hen")
=> false
```

Array#delete_at

Menghapus elemen pada posisi tertentu dalam array.

```
>> data = %W[ape bee cat dog eel fox]
=> ["ape", "bee", "cat", "dog", "eel", "fox"]
>> animal = data.delete_at(3)
=> "dog"
>> animal
=> "dog"
>> data
=> ["ape", "bee", "cat", "eel", "fox"]
```

Array#slice

Memberikan Anda bagian dari array. Tentukan indeks awal dan jumlah item yang diinginkan.

```
>> data = %W[ape bee cat dog eel fox goat hen ibex]
=> ["ape", "bee", "cat", "dog", "eel", "fox", "goat", "hen", "ibex"]
>> data.slice(3, 2)
=> ["dog", "eel"]
```

Array#shuffle

Mengacak array ke dalam urutan acak. Hal ini berguna untuk permainan kartu.

```
>> data = %W[ape bee cat dog eel]
=> ["ape", "bee", "cat", "dog", "eel"]
>> data.shuffle()
=> ["eel", "ape", "cat", "bee", "dog"]
```

Catatan: mengacak tidak mengubah array asli. Jika Anda ingin menggunakan array yang diacak. Anda harus menetapkan ke sebuah variable.

```
data = %W[ape bee cat dog eel]
mixed = data.shuffle()
puts mixed
```

Array#sort

Mengurutkan array ke urutan numerik atau abjad.

```
>> info = [3, 1, 27, 4, 18, 9]
=> [3, 1, 27, 4, 18, 9]
>> info.sort()
=> [1, 3, 4, 9, 18, 27]
```

Array#reverse

membalik isi data array.

```
>> data = %W[fox eel dog cat bee ape]
=> ["fox", "eel", "dog", "cat", "bee", "ape"]
>> data.reverse()
=> ["ape", "bee", "cat", "dog", "eel", "fox"]
```

Array#join

Ini adalah kebalikan dari String # split, melainkan bergabung dengan item dalam array dengan pembatas yang diberikan.

```
>> data = %W[ape bee cat dog]
=> ["ape", "bee", "cat", "dog"]
>> data.join("/")
=> "ape/bee/cat/dog"
>> data.join("; ")
=> "ape; bee; cat; dog"
```

Hash

Hash, kadang bisa disebut jg sebagai *associative array*. Hampir sama dengan array, hash juga merupakan kumpulan dari object reference. Bedanya, index array menggunakan integer. Sedangkan, index hash menggunakan object, bisa berupa string, symbol, bahkan regex, dll. Dengan demikian, hash memiliki dua bagian, yaitu *key*(index yg berupa object) dan *value*. **Penggunaan object sebagai key inilah yg menjadi kelebihan hash dibanding dengan array.**

Tipe data hash di Ruby sama dengan tipe data dictionary di Python. Untuk membuat objek hash yang baru, Kita dapat menggunakan *literal* hash { } ataupun konstruktor Hash.new. Lain halnya dengan array, untuk mengakses elemen objek Hash kita tidak menggunakan indeks seperti array. Kita dapat mendefinisikan kunci(*key*) yang unik untuk setiap elemen. Sama halnya seperti array, apabila kita mengakses elemen dengan *key* yang tidak tercantum di hash, maka akan mengembalikan nilai nil.

```
>> alamat = {  
  ?> "Amat" => "Kemanggisan Raya  
50",  
  ?> "Budi" => "Sudirman 80",  
  ?> "Eric" => "Sandang 9A",  
  ?> "Melissa" => "Pandu 77"  
  ?> }  
=> {"Eric"=>"Sandang 9A",  
  "Budi"=>"Sudirman 80",  
  "Amat"=>"Kemanggisan Raya 50",  
  "Melissa"=>"Pandu 77"}  
>> alamat["Eric"]  
=> "Sandang 9A"  
>> alamat["Dani"]  
=> nil
```

Range

Objek range di Ruby merupakan suatu barisan di mana terdapat nilai awal dan nilai akhir dari barisan tersebut. Objek Range ini sama dengan perintah range() di Python. Untuk membuat suatu objek Range, kita dapat menggunakan operator range '..' dan '...'. Format '..'

akan menciptakan objek Range yang ikut menyertakan elemen terakhirnya, sedangkan format ‘...’ akan menghilangkan elemen terakhirnya. Range di Ruby tidak dianggap sebagai objek Array melainkan dianggap sebagai suatu objek Range sendiri. Untuk menjadikannya objek Array, Kita dapat menggunakan metode `to_a` . Untuk lebih jelasnya, perhatikanlah contoh berikut.

```
>> a = (1..5)
=> 1..5
>> a.class
=> Range
>> (1..5).to_a
=> [1, 2, 3, 4, 5]
>> (1...5).to_a
=> [1, 2, 3, 4]
```

Fungsi

Fungsi pada Ruby adalah sejumlah pernyataan yang dikemas dengan sebuah nama. Selanjutnya nama ini dapat dipanggil beberapa kali dalam program dan tentu saja cara ini dapat mengurangi duplikasi kode. Alasan lain dipecah menjadi sejumlah bagian yang dapat dikelola dengan lebih mudah oleh pemrogram daripada kalau hanya berupa satu bagian kode yang besar.

Bentuk Umum:

```
def nama_fungsi(arg1, arg2, ...)
  <pernyataan-pernyataan>
end
```

Untuk mendefinisikan suatu fungsi, kita menggunakan kata kunci `def` diakhiri dengan `end`. Kita akan membuat fungsi faktorial.

```
#faktorial.rb
#!/usr/bin/env ruby
def faktorial(n)
```

```

hasil = 1
ctr = 0
n.times do
  ctr += 1
  hasil *= ctr
end #end untuk do
return hasil
end #end untuk def
puts "Faktorial 5 = " +
faktorial(5).to_s
puts "Faktorial 10 = " +
faktorial(10).to_s
$ ruby faktorial.rb
Faktorial 5 = 120
Faktorial 10 = 3628800

```

P7.2 Contoh Kasus

Contoh Kasus 1

Pada contoh kasus yang pertama yaitu membuat program untuk mencari nilai rata-rata dari nilai-nilai yang terdapat pada data yang berbentuk array. Output yang akan ditampilkan adalah sebagai berikut :



```

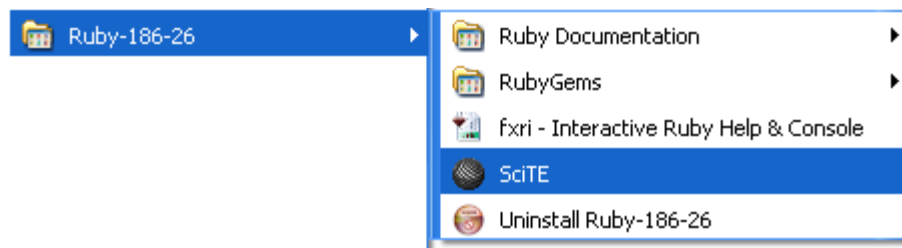
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\User>cd\
C:\>cd ruby
C:\ruby>cd bin
C:\ruby\bin>ruby kasus1.rb
Nilai rata-rata dari data tersebut adalah 19.0
C:\ruby\bin>

```


Langkah-langkah pengerjaan adalah sebagai berikut :

1. Klik tombol start Program Ruby 186-26 lalu pilih SciTe dengan tampilan sebagai berikut :

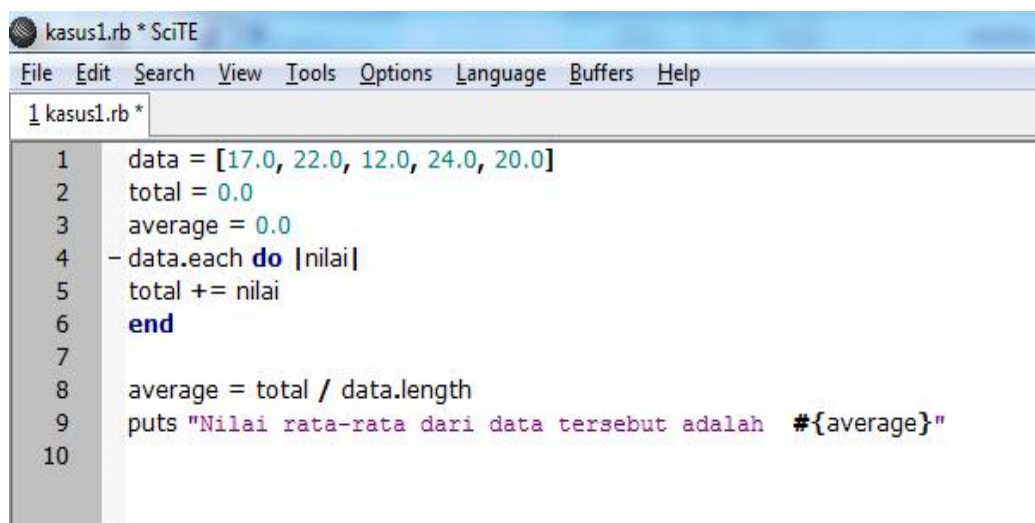


2. Klik Menu File -> New lalu ketikkan listing program sebagai berikut.

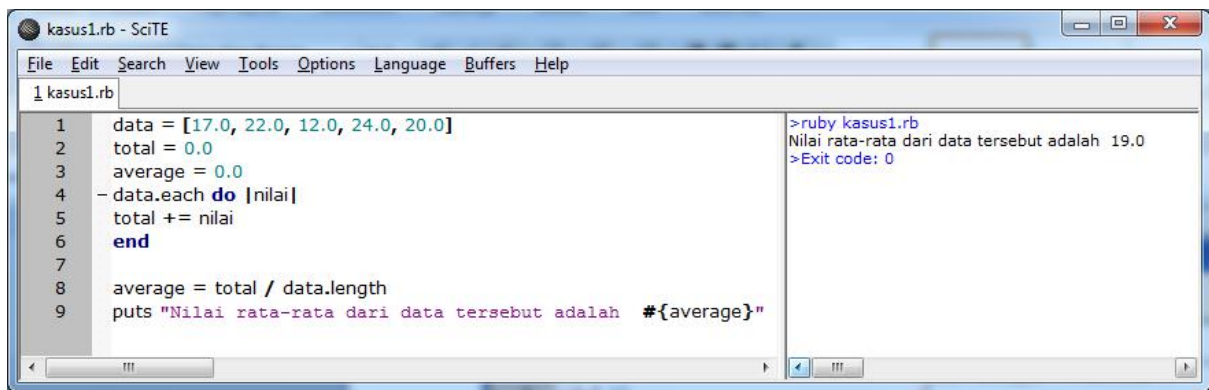
```
data = [17.0, 22.0, 12.0, 24.0, 20.0]
total = 0.0
average = 0.0
data.each do |nilai|
  total += nilai
end

average = total / data.length
puts "Nilai rata-rata dari data tersebut adalah #{average}"
```

3. Setelah selesai mengetikkan code, langkah selanjutnya menyimpan file tersebut dengan cara klik menu File -> Save As. Masukkan nama file dengan nama kasus1.rb



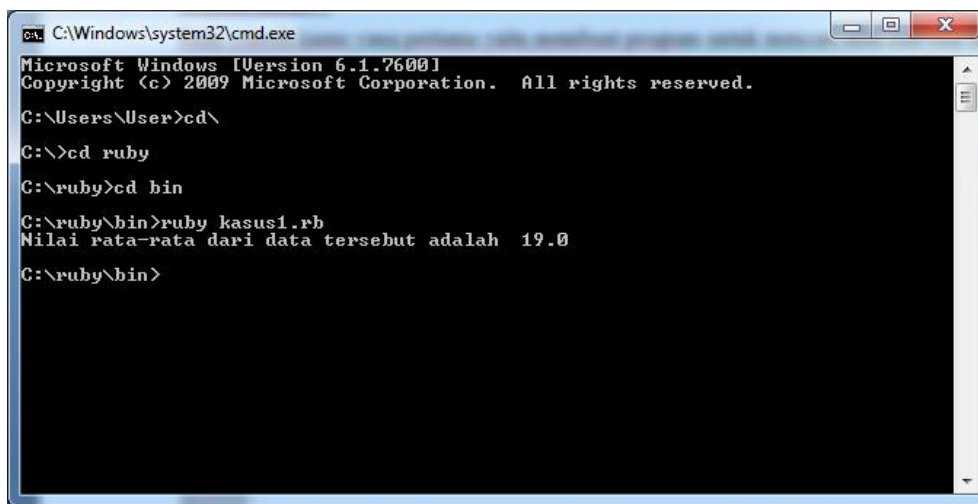
4. Setelah itu menjalankan program dengan cara menekan tombol F5.



```
1 data = [17.0, 22.0, 12.0, 24.0, 20.0]
2 total = 0.0
3 average = 0.0
4 data.each do |nilai|
5   total += nilai
6 end
7
8 average = total / data.length
9 puts "Nilai rata-rata dari data tersebut adalah #{average}"
```

```
>ruby kasus1.rb
Nilai rata-rata dari data tersebut adalah 19.0
>Exit code: 0
```

5. Selain itu kita juga dapat menjalankan program pada command prompt dengan cara mengetikkan `C:\Ruby\bin> ruby kasus1.rb`



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\User>cd \
C:\>cd ruby
C:\ruby>cd bin
C:\ruby\bin>ruby kasus1.rb
Nilai rata-rata dari data tersebut adalah 19.0
C:\ruby\bin>
```

6. Apabila tidak ada error maka program yang kita *compile* berhasil.

Contoh Kasus 2

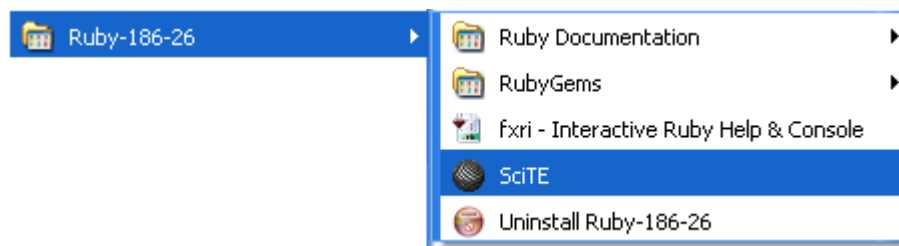
Pada contoh kasus kedua adalah program dengan menggunakan dua array, satu berisi nama-nama orang dan array lainnya berisi usia mereka. Ini disebut array paralel, informasi dalam sebuah elemen dari satu array berhubungan dengan informasi dalam elemen sesuai dari array lain. Program akan meminta *user* menginput nama yang terdapat pada array dan memberikan kembali usia yang sesuai. Hal ini bergantung pada metode Array `#index`, yang mencari entri dan mengembalikan nomor posisi, atau nihil jika nama tersebut tidak ditemukan. Output yang akan ditampilkan adalah sebagai berikut :

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\User>cd\
C:\>cd ruby\bin
C:\ruby\bin>ruby kasus2.rb
Masukkan Nama: Selena
Selena berusia 20 tahun.
C:\ruby\bin>ruby kasus2.rb
Masukkan Nama: IsnI
Saya tidak tahu IsnI itu siapa.
C:\ruby\bin>_
```

Langkah-langkah pengerjaan adalah sebagai berikut :

1. Klik tombol start Program Ruby 186-26 lalu pilih SciTe dengan tampilan sebagai berikut :

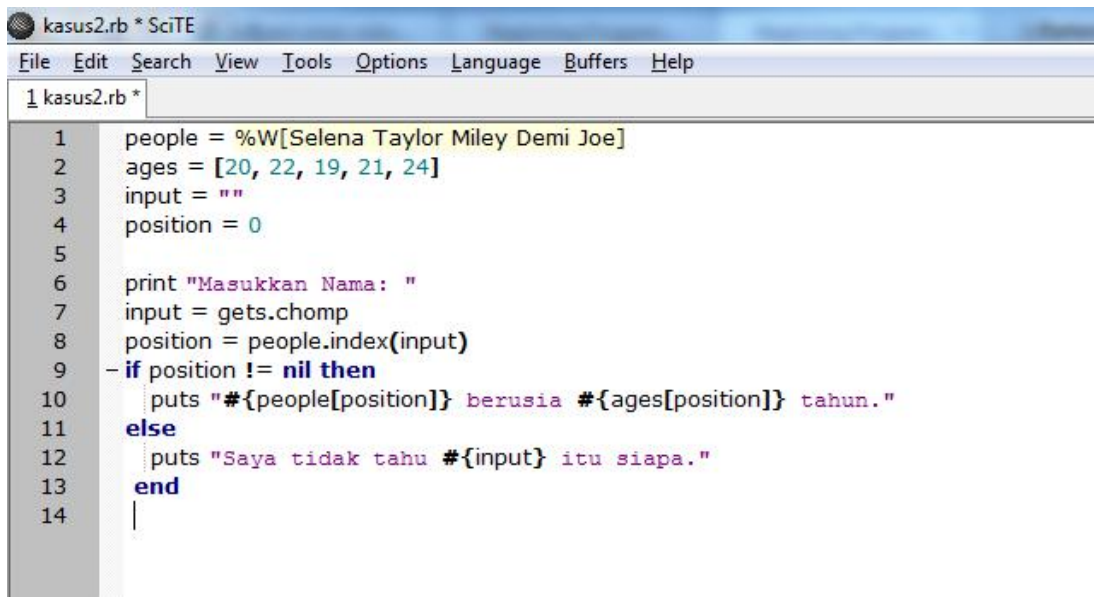


2. Klik Menu File -> New lalu ketikkan listing program sebagai berikut.

```
people = %W[Selena Taylor Miley Demi Joe]
ages = [20, 22, 19, 21, 24]
input = ""
position = 0


print "Masukkan Nama: "
input = gets.chomp
position = people.index(input)
if position != nil then
  puts "#{people[position]} berusia #{ages[position]} tahun."
else
  puts "Saya tidak tahu #{input} itu siapa."
end
```

3. Setelah selesai mengetikkan code, langkah selanjutnya menyimpan file tersebut dengan cara klik menu File -> Save As. Masukkan nama file dengan nama kasus2.rb



```
1 people = %W[Selena Taylor Miley Demi Joe]
2 ages = [20, 22, 19, 21, 24]
3 input = ""
4 position = 0
5
6 print "Masukkan Nama: "
7 input = gets.chomp
8 position = people.index(input)
9 if position != nil then
10   puts "#{people[position]} berusia #{ages[position]} tahun."
11 else
12   puts "Saya tidak tahu #{input} itu siapa."
13 end
14 |
```

4. Selanjutnya adalah menjalankan program pada command prompt dengan cara mengetikkan `C:\Ruby\bin> ruby kasus2.rb`



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\User>cd \ruby
C:\ruby>cd bin
C:\ruby\bin>ruby kasus2.rb
Masukkan Nama: Taylor
Taylor berusia 22 tahun.

C:\ruby\bin>ruby kasus2.rb
Masukkan Nama: Isn'i
Saya tidak tahu Isn'i itu siapa.

C:\ruby\bin>
```

5. Apabila tidak ada error maka program yang kita compile berhasil.

P7.3 Latihan

Latihan 1

Berikut ini merupakan program yang menggunakan bahasa ruby versi 2.6 untuk menginput nama dan usia, serta menampilkan hasil inputan nama dan usia tersebut secara *ascending*, dikerjakan sesuai dengan konsep HASH. Pada code editor di ruby 186 ketikkan program berikut.

(Lengkapi kode program berikut dengan mengisi titik-titik yang berwarna merah)

```
age_hash = { } # start with an empty hash

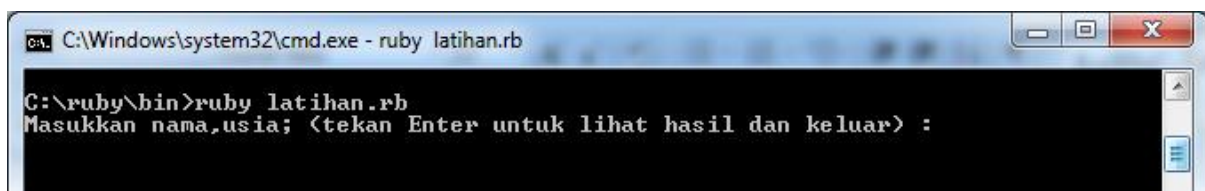
input = .....
name = .....
age = 0
keylist = []
print "Masukkan nama,usia; (tekan Enter untuk lihat hasil dan
keluar) : "
input = .....
while input != "" do
    (name, age) = .....
    .....

    .....

print "Masukkan nama,usia; (tekan Enter untuk lihat hasil dan
keluar) : "
.....
.....

puts "Berikut adalah konten dari HASH dan sudah diurutkan :"
keylist = age_hash.keys.sort
.....
puts "#{key} berusia #{age_hash[key]} tahun."
.....
```

TAMPILAN OUTPUT PROGRAM LATIHAN 1



(data nama dan usia diinput oleh user)

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\User>cd\ruby\bin

C:\ruby\bin>ruby latihan.rb
Masukkan nama,usia; (tekan Enter untuk lihat hasil dan keluar) : Anita, 22
Masukkan nama,usia; (tekan Enter untuk lihat hasil dan keluar) : Dani, 22
Masukkan nama,usia; (tekan Enter untuk lihat hasil dan keluar) : Sisca, 21
Masukkan nama,usia; (tekan Enter untuk lihat hasil dan keluar) : Jennifer, 20
Masukkan nama,usia; (tekan Enter untuk lihat hasil dan keluar) : Alya, 19
Masukkan nama,usia; (tekan Enter untuk lihat hasil dan keluar) : Ridwan, 23
Masukkan nama,usia; (tekan Enter untuk lihat hasil dan keluar) :
Berikut adalah konten dari HASH dan sudah diurutkan :
Alya berusia 19 tahun.
Anita berusia 22 tahun.
Dani berusia 22 tahun.
Jennifer berusia 20 tahun.
Ridwan berusia 23 tahun.
Sisca berusia 21 tahun.

C:\ruby\bin>
```

KOREKSI PROGRAM LATIHAN 1

```
latihan.rb * SciTE
File Edit Search View Tools Options Language Buffers Help
1 latihan.rb *
1  age_hash = { } # start with an empty hash
2
3  input = ""
4  name = ""
5  age = 0
6  keylist = []
7  print "Masukkan nama,usia; (tekan Enter untuk lihat hasil dan keluar) : "
8  input = gets.chomp
9  - while input != "" do
10     (name, age) = input.split(",")
11     age = age.to_i
12
13     age_hash[name] = age
14
15     print "Masukkan nama,usia; (tekan Enter untuk lihat hasil dan keluar) : "
16     input = gets.chomp
17     end
18
19     puts "Berikut adalah konten dari HASH dan sudah diurutkan :"
20     keylist = age_hash.keys.sort
21     - keylist.each do |key|
22         puts "#{key} berusia #{age_hash[key]} tahun."
23     end
24     |
```

Latihan 2

Berikut ini merupakan program fungsi pada ruby yang digunakan untuk mencari luas dan keliling dari bangun datar persegi dan persegi panjang dari menu yang disediakan. Pada code editor di ruby 186 ketikkan program berikut.

(Lengkapi kode program berikut dengan mengisi titik-titik yang berwarna merah)

```
def luas()
w=0
while w!=3
puts "1. Persegi"
puts "2. Persegi Panjang"
puts "3. Exit"
print "masukkan Pilihan: "
.....
if w==1
persegil
elsif w==2
ppl
elsif w==3
puts ''
else
puts "Anda Salah Memasukkan Pilihan"
end
end
.....
def keliling()
w=0
while w!=3
puts "1. Persegi"
puts "2. Persegi Panjang"
puts "3. Exit"
print "Masukkan Pilihan: "
w=gets.to_i
.....
persegik
.....
ppk
.....
puts ''
else
puts "Anda Salah Memasukkan Pilihan"
end
end
end

.....
print "masukkan sisi= "
```

```

b=a*a
print "luasnya adalah",b
puts ''
end

.....
print "masukkan panjang= "
p=gets.to_i
print "masukkan lebar= "
l=gets.to_i
.....
print "luasnya adalah ",m
puts ''
end

.....
print "masukkan sisi= "
.....
.....
print "keliling=",b
puts ''
end

.....
print "masukkan panjang= "
p=gets.to_i
print "masukkan lebar= "
l=gets.to_i
m=2*(p+l)
.....
puts ''
end

z=0
while z!=3
puts "1. Luas"
puts "2. Keliling"
puts "3. Exit"
print "Masukkan Pilihan: "
.....
if z==1 then
luas
.....
keliling
.....
puts ''
else
puts "Anda Salah Memasukkan Pilihan"
end
.....

```


TAMPILAN OUTPUT PROGRAM LATIHAN 2

```
C:\ruby\bin>ruby latihan2.rb
1. Luas
2. Keliling
3. Exit
Masukkan Pilihan: 1
```

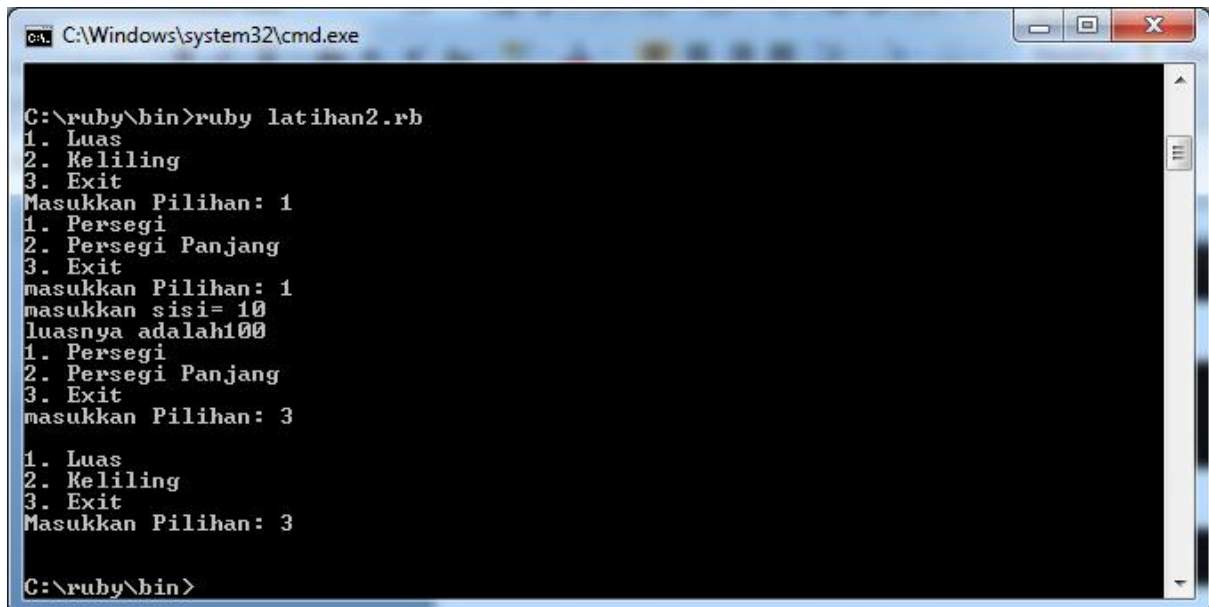


```
1. Persegi
2. Persegi Panjang
3. Exit
masukkan Pilihan: 1
```



```
masukkan sisi= 10
luasnya adalah100
```

Tampilan keseluruhan :



```
C:\Windows\system32\cmd.exe
C:\ruby\bin>ruby latihan2.rb
1. Luas
2. Keliling
3. Exit
Masukkan Pilihan: 1
1. Persegi
2. Persegi Panjang
3. Exit
masukkan Pilihan: 1
masukkan sisi= 10
luasnya adalah100
1. Persegi
2. Persegi Panjang
3. Exit
masukkan Pilihan: 3
1. Luas
2. Keliling
3. Exit
Masukkan Pilihan: 3
C:\ruby\bin>
```

KOREKSI PROGRAM LATIHAN 2

```
def luas()
  w=0
  while w!=3
    puts "1. Persegi"
    puts "2. Persegi Panjang"
    puts "3. Exit"
    print "masukkan Pilihan: "
    w=gets.to_i
    if w==1
      persegi
    elsif w==2
```

```

ppl
elsif w==3
puts ''
else
puts "Anda Salah Memasukkan Pilihan"
end
end
end

```

```

def keliling()
w=0
while w!=3
puts "1. Persegi"
puts "2. Persegi Panjang"
puts "3. Exit"
print "Masukkan Pilihan: "
w=gets.to_i
if w==1 then
persegik
elsif w==2 then
ppk
elsif w==3 then
puts ''
else
puts "Anda Salah Memasukkan Pilihan"
end
end
end

```

```

def persegil()
print "masukkan sisi= "
a=gets.to_i
b=a*a
print "luasnya adalah",b
puts ''
end

```

```

def ppl()
print "masukkan panjang= "
p=gets.to_i
print "masukkan lebar= "
l=gets.to_i
m=p*l
print "luasnya adalah ",m
puts ''
end

```

```

def persegik()
print "masukkan sisi= "
a=gets.to_i
b=a*4

```

```

print "keliling=",b
puts ''
end

def ppk()
print "masukkan panjang= "
p=gets.to_i
print "masukkan lebar= "
l=gets.to_i
m=2*(p+l)
print "keliling=",m
puts ''
end

z=0
while z!=3
puts "1. Luas"
puts "2. Keliling"
puts "3. Exit"
print "Masukkan Pilihan: "
z=gets.to_i
if z==1 then
luas
elsif z==2 then
keliling
elsif z==3 then
puts ''
else
puts "Anda Salah Memasukkan Pilihan"
end
end
end

```

P7.4 DaftarPustaka

- <http://www.ruby-lang.org/id/>
- <http://evc-cit.info/cit020/beginning-programming/index.html>
- <http://free-coding.blogspot.com/>

