

# Nyankod Magz

Love it and Earn It

Emping Boolean





# **Nyankod Magz Edisi 8**

## **“Emping Boolean”**

Love it and Earn It

# Sapa Nyankod

Goooooaaaaa!!!!!!

Akhirnya goal juga cuy... hehehe... Maksudnya terbit juga. Maklum lagi Piala Eropa jadi semuanya dikait-kaitin sama bola. Bahkan temen-temen Nyankodist Jakarta sampe pada beli kostum bola dari tim kesayangannya masing-masing dan hampir setiap minggu maen futsal. Ya biar ikutan ngerasain serunya Piala Eropa githu. Hehe.. btw ngomong-ngomong Piala Eropa 2012, nyankoders sekalian pilih negara apa nih di Piala Eropa 2012 ini?? Jerman?? Inggris?? Italy? Spanyol?? Portugal? Apa aja boleh deh, yang penting jangan pada berantem sama yang beda tim kesayangan...



Oh ya, udah 3 minggu lagi nih berlalu setelah edisi ke-7 Dodol Query terbit. Semoga Dodolnya enak ya nyankoders sekalian. Di edisi kali ini, kita juga punya sesuatu yang nggak kalah enakya lho, yaitu Emping Boolean. Mau tau gimana rasanya emping yang satu ini?? Ntar cobain aja sendiri ya. Cuma saya mau kasih gambaran aja nih tentang Emping yang satu ini. Di Emping Boolean ini bakal ada sebuah rubrik baru lho, yaitu NyanTerview. Kita udah nyiapin hasil wawancara kita dengan salah satu startup yang lagi berkembang nih, mau tau startup apakah itu?? Hehe... Biar lebih seru dan asyik, ntar baca aja sendiri ya.... :D

Namun sebelum mulai mencicipi, seperti biasa jangan lupa siapin kopi hangat, beberapa gorengan dan list musik dangdut. Hehehe... ya biar belajarnya makin asyik.

Delapan memang angka yang luar biasa coy, sebuah angka yang nggak ada ujungnya, bentuknya aja kayak borgol. Keren bukan?? Mangkanya saking kerennya akhirnya pasukan pemberantas teroris di Indonesia disebut dengan nama Densus 88. Tuh kan ada 88-nya. Katanya sih, supaya bisa memberantas terorisme tanpa henti dan tanpa ujung, seperti angka 88. Pemain bola yang hebat-hebat juga banyak yang nomor punggungnya 8. Nggak percaya?? Nih kan lagi Piala Eropa, coba deh ntar cek sendiri siapa aja pemain bola hebat yang nomor punggungnya 8 itu siapa aja. Ya

begitulah angka 8. Namun selain itu juga, angka 8 ini penuh dengan rintangan yang luar biasa juga. Doakan saja kawan2 nyankoders sekalian semoga kami diberikan kekuatan supaya terus tanpa menyerah buat menulis dan mengembangkan Nyankod.

Seperti kata pepatah, berakit-rakit ke hulu, naek speedboat kemudian. Hehehe... soalnya saya nggak bisa renang coy. Kalo berenang-renang kemudian, ya bakalan repot coy. Maklum saya pemakan buah iblis jadi nggak bisa berenang. Hehehe... Saya adalah pemakan buah iblis unyu-unyu dengan kemampuan khusus teleportasi. #ngarangBanget. Jadi pasti dalam segala sesuatu itu, kita pasti bakal ngalamin yang namanya masa sulit, tapi kalo kita tetep konsisten dalam menjalaninya pasti kita bakal nemu jalannya dan akan merasakan manisnya kemudian. #sokBijak

Oke coy, daripada ribed, mendingan langsung cicipi aja deh, hidangan kami kali ini, yaitu Emping Boolean, rasakan tendangannya. Hehehe.... Selamat belajar dan selamat menempuh hidup baru. Salam coding!!

***Love it and earn it***

**Kresna Galuh D. Herlangga  
Pimpinan Redaksi NyankodMagz**

NB:

*Orang yang memakan buah iblis dalam anime One Piece akan mempunyai kemampuan khusus, seperti tubuhnya bisa melar seperti karet, bisa mengeluarkan api, atau bisa mengendalikan es, ato bisa kekuatan-kekuatan yang lainnya, namun konsekuensinya adalah nggak akan pernah bisa berenang*

Bila ada pertanyaan seputar Nyankod, atau mau bertanya tentang pemrograman kepada Nyankodist Team, atau hanya sekedar silaturahmi juga boleh, asalkan jangan spam, silahkan kontak kami ke email kami:  
[mail@nyankod.com](mailto:mail@nyankod.com).

Anda juga dapat berkomentar langsung seputar konten majalah di post artikel di situs Nyankod atau langsung menghubungi nyankodist pada kontak yang telah disediakan.

Jangan lupa follow twitter kami ya!



**@nyankodTWEET**

## Di dalam sini, ada..

Sapa Nyankod	Halaman 3
Kontak Nyankod	Halaman 5
Seru-seruan nyankodTWEET	Halaman 7
Nyankoders Menyapa...	Halaman 8
C - Pointer	Halaman 12
Ruby - JIKA AKU ME-NYANKOD!!!	Halaman 20
nyanTerview - jago{coding}	Halaman 33
Perl - [Ulangan Episode 1]	Halaman 39
JavaScript - Menutup Celah Error	Halaman 51
Python - Pakai Fungsi dan Prosedur Biar Kodenya Gak Redundant	Halaman 60
ActionScript { Penggunaan NextFrame dan PrevFrame}	Halaman 72
#nyanKomik - Kamu Dikatakan Maniak Apabila:	Halaman 81
Agenda Terbit Edisi ke-9	Halaman 86

## Seru-seruan nyankodTWEET

Buat temen-temen yang pengen belajar Drupal atau lagi belajar Drupal (sebuah CMS buat bikin web), mendingan ikutin deh twit-twitnya kang Toni Haryanto ([@ToHaryan](#)) di hashtag #BelajarDrupal.



Lumayanlah, twitteran tapi juga plus dapet ilmu. Temen-temen juga bisa nanya-nanya lho seputar Drupal n JavaScript ke kang Toha (panggilan akrabnya).

Selaen itu juga, jangan lupa ya buat follow @nyankodTWEET....!!!! Dapetin informasi tentang Nyankod, plus twit-twit berkaitan dengan pemrograman.

# Nyankoders Menyapa...

Dari **azza dha** ([aqrowya@gmail.com](mailto:aqrowya@gmail.com))

akang nyankod yg unyu2...hehe  
kang klo bisa rilis majalah jangan terlalu lama yaaa...  
sya selalu nunggu ni,,hehe

---

ade yg unyi2....  
buat nyankodMagz emang 3 minggu sekali. tp tenang, qt lg ngegarap  
nyankodPaper, yg ini g terbit 3 minggu kok. kita usahain ntr tiap minggu  
ada yg terbit. doain yg ade yg unyi2... hehe.

---

Dari **azza dha** ([aqrowya@gmail.com](mailto:aqrowya@gmail.com))

aamiin,,,semakin cepat semakin baik kang #YusufKallasay

oia kang boleh minta stu permintaan lagi yaaa,,  
klo bisa setiap majalahnya ada stu yg diselipkan untuk bikin stu  
project dri biasanya,heheh(ya itung2 agar para pemula seperti ini  
semakin meraskan manfaatnya seperti saya ini) ,,bebas deh kang  
projectnya menggunakan bahasaa apa,,yang penting gak jadi programmer  
gaptek,he  
makasih yaa buat akang2 nyankod yang unyu2 untuk semua ilmu yang dshare  
ke kita2 khususnya ke saya...

maju trus dunia IT indonesia,,

---

Hehehe... boleh deh ntr dicoba buat diselipin tutorial bikin project tertentu. Tapi sebenarnya rencana kita sih ntr kalo emang materi pokok udah habis dibahas semua, rencananya malah setiap edisinya bakal kita bahas project-project dari kasus-kasus tertentu. Ya semisal bikin aplikasi sederhana gitu. Tapi sabar dlu ya, mau tamatin materi dulu.

---

Dari **Farid Hidayat** ([faridh05@yahoo.com](mailto:faridh05@yahoo.com))

Salam Sejahtera,  
Saya sangat berharap lamaran magang saya diterima, Terima Kasih.

Salam,

Farid

Attachments:

- Surat Magang
  - CV
  - Portofolio
  - Sertifikat
- 

Ade Farid yg baik, kita udah baca surat magang kamu dan udah cek CV kamu. Luar biasa, kreatif banget... kita sih mau-mau aja nerima kamu buat magang, tapi sayangnya buat saat ini, kita blom punya kantor sendiri, jadi blom bisa nerima lamaran magang kamu. Mudah-mudahan tahun depan kita udah punya kantor sendiri. hehehe... Sekali lagi sorry ya....

Oh ya, photo kamu di CV lucu juga tuh....  
Tapi sayang Portofolio.swf kamu kayaknya rusak tuh, muter2 terus nggak karuan. Padahal kita penasaran pengen liat...

Salam Hangat

---

Kalo ada pertanyaan, seputar nyankod silahkan jangan ragu-ragu untuk mengirimkan pertanyaan kamu melalui email ke [mail@nyankod.com](mailto:mail@nyankod.com). Sebisa mungkin akan kami jawab. Dan buat temen-temen yang pengen nanya tentang pemrograman tertentu, silahkan hubungi penulisnya ya... cemunguddhhh eaaaa.....

[C]



Ade Kurniawan  
@adekurniawan  
noadekur@yahoo.com



# Pointer

Kira-kira siapa yah juara Euro kali ini, hehehehe mumpung lagi demam sepakbola Euro 2012. Spanyol lagi atau Inggris yah, hahaha ngarep aja padahal Inggris datang dengan kekuatan seadanya karena banyak pemain yang cedera seperti Lampard, Cahill, Barry, berharap sih kayak Chelsea walau gak diunggulkan eh bisa jadi juara Liga Champion dengan perjuangan heroik. Lho malah jadi ngomongin bola, sedikit intermezzo aja biar gak pusing pas lagi belajar coding, oke kita tinggalkan sejenak mengenai perhelatan akbar sepakbola dan saatnya ke membahas mengenai programming, yupss sekarang ini kita akan membahas mengenai pointer (makhluk apa lagi ini) yups pointer adalah seorang anak kecil bertubuh tambun,,hehehehe emangnya si gopur tambun. Hehehehe moga dia gak baca tulisan ini. Pointer adalah yaudah biar lebih jelas lagi, langsung liat ke tkp.

## Pointer

---

Pointer merupakan suatu variabel yang berisi alamat dari suatu variabel lain, jadi pointer bukan berisi dengan suatu nilai data tetapi berisi dengan suatu alamat. Hubungan antara variabel dan pointer dalam kehidupan sehari-hari bisa dimisalkan seperti alamat dan rumah, misalkan alamat anda bambu apus rt 001/09 Jakarta maka akan mengacu kepada rumah anda, rumah bisa mempunyai banyak bentuk dan ukuran sedangkan alamat biasanya mempunyai ukuran yang sama ( jalan, kota, propinsi). Yups seperti pointer yang mengacu kepada variable lain, alamat juga mengacu kepada rumah, jadi bagaimanapun bentuk rumahnya mau kecil atau besar, alamat tetap mempunyai bentuk yang sama yaitu terdiri dari nama jalan, kota,

propinsi. Contoh lain misalnya hati anda yang mengacu kepada kekasih anda, pada saat itu hati anda berperan sebagai pointer yang berisi alamat dari hati kekasih anda, hehehe tar ada yang galau lagi kalo denger hal-hal seperti ini, terutama "forever alone".

Pointer dapat berisi alamat yang mengacu kepada jenis data apapun, tetapi waktu variable pointer dideklarasikan harus dituliskan jenis(tipe) data yang akan diacunya. Seperti pada variable biasa, sebuah variable harus dideklarasikan sebelum digunakan, deklarasi pointer dilakukan dengan menambahkan karakter bintang(\*) sebelum nama pointer.

### Deklarasi pointer :

```
1 | Tipe *nama_variabel_pointer
2 |
3 | Int *fussy
4 | Float *ps
5 | Float * gh
```

Karakter bintang (\*) pada pointer boleh dituliskan menyatu atau terpisah dengan nama variable pointer, seperti pada contoh diatas penulisan variable pointer float \*ps yang berarti pointer ps menunjuk sebuah float, artinya pointer ps berisi suatu alamat dengan memori yang terdapat di alamat tersebut berupa float.

Setiap variable tentunya memiliki operator yang akan digunakan, pada bahasa C terdapat 2 operator yang berhubungan dengan operasi pointer, diantaranya :

- a. & operator ini bersifat unary( mengambil satu operan, bukan operan umpan silang dari Ronaldo y), digunakan untuk mendapatkan alamat memori operan tersebut. Biasanya alamat memori ini selanjutnya diberikan( di-assign) kepada variable pointer.

- b. \* operator ini bersifat unary. Operator ini digunakan untuk mengakses data yang alamatnya ditunjuk oleh pointer.

### Program 8.1 :

```
1  #include <stdio.h>
2  int main()
3  {
4  int thing_var; /* define a variable for thing */
5  int *thing_ptr; /* define a pointer to thing */
6  thing_var = 2; /* assigning a value to thing */
7  printf("Thing %d\n", thing_var);
8  thing_ptr = &thing_var; /* make the pointer point to thing */
9  *thing_ptr = 3; /* thing_ptr points to thing_var so */
10 /* thing_var changes to 3 */
11 printf("Thing %d\n", thing_var);
12 /* another way of doing the printf */
13 printf("Thing %d\n", *thing_ptr);
14 return (0 );
15 }
```

Output dalam konsol :

```
Thing 2
Thing 3
Thing 3
```

Dalam program diatas merupakan contoh simple penggunaan pointer, output dari program diatas yaitu, Thing 2 nilai pertama ketika variable thing diberikan nilai 2 pada baris ke 6, kemudian Thing 3 ketika pointer thing diberikan nilai pada baris ke 9.

### Program 8.2 :

```
1  #include <stdio.h>
2  int array[] = {4, 5, 8, 9, 8, 1, 0, 1, 9, 3};
3  int *array_ptr;
4  int main()
5  {
6  array_ptr = array;
7  while ((*array_ptr) != 0)
8  ++array_ptr;
9  printf("Number of elements before zero %d \n",
10 array_ptr - array);
11 return (0);
12 }
```

Output dalam konsol :

```
Number of elements before zero 6
```

Contoh program diatas merupakan contoh penggunaan pointer didalam array, perhatikan ketika kita ingin memeriksa data didalam array kita menggunakan operator (\*) yang merupakan operator pengakses data yang alamatnya ditunjuk oleh pointer. Dalam data array diatas terdapat 10 data, program diatas bertugas mencari berapa elemen sebelum nilai elemen 0, ekspresi while((\*array\_ptr) !=0) berarti array akan mencari sampai ditemukan data 0, kemudian untuk menghitung berapa jauh array\_ptr didalam array digunakan ekspresi array\_ptr - array.

### Program 8.3 :

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  /***** *****/
5  * my_strchr --Finds a character in a string. *
6  * Duplicate of a standard library function, *
7  * put here for illustrative purposes *
8  * *
9  * Parameters *
10 * string_ptr --String to look through. *
11 Kurakura
12 239
13 * find --Character to find. *
14 * *
15 * Returns *
16 * pointer to 1st occurrence of character *
17 * in string or NULL for error. *
18 *****/
19 char *my_strchr(char * string_ptr, char find)
20 {
21 while (*string_ptr != find) {
22 /* Check for end */
23 if (*string_ptr == '\0')
24 return (NULL); /* not found */
25 ++string_ptr;
26 }
27 return (string_ptr); /* Found */
28 }
29 int main()
30 {
31 char line[80]; /* The input line */
32 char *first_ptr; /* pointer to the first name */
33 char *last_ptr; /* pointer to the last name */
```

```
34 fgets(line, sizeof(line), stdin);
35 /* Get rid of trailing newline */
36 line[strlen(line) - 1] = '\0';
37 last_ptr = line; /* last name is at beginning of line */
38 first_ptr = my_strchr(line, '/'); /* Find slash */
39 /* Check for an error */
40 if (first_ptr == NULL) {
41     fprintf(stderr,
42     "Error: Unable to find slash in %s \n", line);
43     exit (8);
44 }
45 *first_ptr = '\0'; /* Zero out the slash */
46
47 ++first_ptr; /* Move to first character of name */
48 printf("First:%s Last:%s\n", first_ptr, last_ptr);
49 return (0);
50 }
```

Pointer merupakan suatu variabel yang berisi alamat dari suatu variabel lain, dalam penggunaan pointer harus memperhatikan tipe data yang diacu pada pointer, missal variable yang diacu oleh pointer bertipe integer maka harus mempunyai tipe integer juga pointer tersebut, dan juga jangan memanipulasi pointer sebelum pointer tersebut mengacu kepada alamat memori yang semestinya. Dan jangan lupa pointer mengacu kepada alamat jadi tidak berisi suatu data, kecuali kita telah memanipulasi ke alamat memori yang dituju, seperti hati anda kepada kekasih anda,,hehehe buat "forever alone" maaf anda saya singgung lagi...:)

**Referensi :**

- Practical C – O'reilly Publisher.
- Algoritma dan Struktur Data Bahasa C  
Thompson Susabda Ngoen

# [Ruby]



Muhammad Singgih Zulfikar Anshori  
@hirokakaoshi  
m.singgih.za@gmail.com  
<http://mszacompany.wordpress.com>



## JIKA AKU ME-NYANKOD!!!

Hohohoho... Hello back ladies and gentleman...  
Welcome back to Nyankod Magazine 8th Edition...

Now, we will speak english at the whole time when u learn ruby in this edition.  
Tapi gajadi deh..repot juga pake bahasa inggris wong yang baca orang endonesa haaha..

Edisi sekarang ini bahasan kita mengenai *Control Structures* yang artinya Struktur Kontrol...“??? klo itu mah ane juga tau bang(-\_\_-”)”

hehe maksudnya penggunaan *kondisi dan syarat dalam mengeksekusi aksi yang kita inginkan, aksi tersebut baru dapat dijalankan apabila memenuhi syarat yang ditentukan.*

Untuk sekarang kita bahas mengenai **IF** , haha bagi yang sudah tau tolong pura2 ga tau aja yah :P IF artinya JIKA... ini merupakan syarat ketika akan mengeksekusi perintah tertentu sesuai dengan yang kita tentukan.

Nyok kita review dulu apa aja sih kemaren tuh, ada Latihan juga kan kalau ga salah..ada source code jawaban dari latihan kemarin di edisi ini. Poin yang sudah dipelajari kemarin ini nih :

- a) *Belajar menggunakan Variabel lokal di Class*
- b) *Belajar enkapsulasi dan type kontrol akses method dalam kelas, dengan kata lain melakukan pendefinisian tingkat keamanan dari methods dalam class.*

Baiklah kita mulai sodara sodara \.\.^o^))\*!

## Level - 8

---

Berdo'a Dulu yuks...semoga dapat membawa kebaikan belajar sekarang ini... bakal kepeke banget soalnya (perasaan apapun kepeke deh (^\_^) ).

Tahap persiapan :

1. Buka **terminal** bila kamu pakai linux, **cmd** kalau kamu pakai windows.
2. Masuk ke folder RubyDOJO di terminal :

```
cd RubyDOJO/
```

dan CMD dengan perintah :

```
cd C:\Ruby193\bin\RubyDOJO\
```

versi Ruby nya disesuaikan dengan yang kamu instal ya.

3. Buka editor plain teks favorit kamu, kalau saya di GNU/Linux Ubuntu pakai **GEDIT**. Kalau kamu pakai Windows gunakan Notepad.

Get readyy.... Gooo

### I. IF - ELSE - END

---

IF merupakan perintah kondisional, contoh :

**Saya mau beli Sepatu Nike dengan walk Recognitions nya,**  
**JIKA uangnya cukup saya bisa beli sepatu,**  
**JIKA tidak, saya tidak bisa beli Sepatu itu.**

Nah sekarang, kita coba buat programnya deh di Ruby :

### Program 8.1: Jika ada Uang (IFSepatu.rb)

```
1  #!/usr/bin/ruby
2  #Belajar IF pertama, pengen beli sepatu ceritanya
3  puts "Harga Sepatu = 600.000 Rupiah"
4  printf "Uang kamu berapa? : " #bertanya
5  @uangSaya = gets.to_i #inputan dari user gets.to_i mengubah string
6  jadi integer
7
8  if @uangSaya >= 600000 #awal kondisi #jika uang lebihbesar/samadengan
9  600.000
10     puts "SEPATU JADI MILIKKU CULAMANYAH!!! 0__o" #maka dapat
11 else #kalau tidak
12     puts "Sudah, Mroyek Dulu Sanah!!" #ga dapat :(
13 end #akhir dari if
```

Lihat deh, kode program di atas adalah bentuk dari penggunaan IF kondisional. Simpan di folder *RubyDOJO*. Ketika di eksekusi akan tampil seperti berikut :

**terminal :**

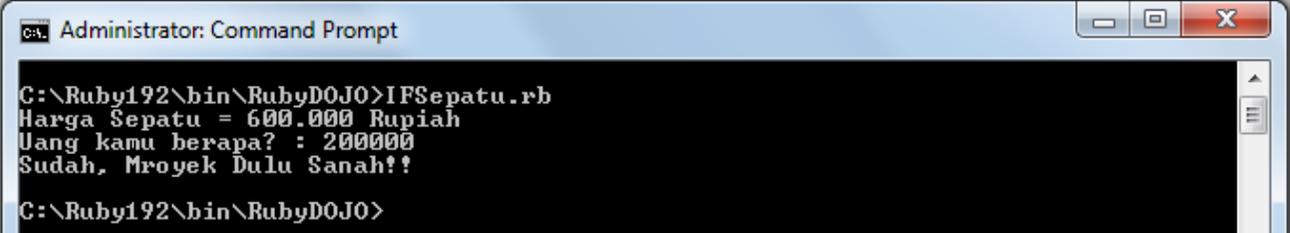


```
hirokaoshi@hirokaoshi-Lenovo-G450: ~/Project/RubyDOJO
File Edit View Search Terminal Help
hirokaoshi@hirokaoshi-Lenovo-G450:~/Project/RubyDOJO$ chmod +X IFSepatu.rb
hirokaoshi@hirokaoshi-Lenovo-G450:~/Project/RubyDOJO$ ./IFSepatu.rb
Harga Sepatu = 600.000 Rupiah
Uang kamu berapa? : 4000000
SEPATU JADI MILIKKU CULAMANYAH!!! 0__o
hirokaoshi@hirokaoshi-Lenovo-G450:~/Project/RubyDOJO$ █
```

Folder RubyDOJO saya simpan di dalam folder Project, jadi tak perlu bingung ya, kalo kamu kan nyimpenya di RubyDOJO langsung... ^\_\_^

saya menginputkan 400.000 sedangkan sepatu 600.000, ya ga cukup lah jadi weh...harus Mroyek lagi :(

**CMD :**



```
Administrator: Command Prompt
C:\Ruby192\bin\RubyDOJO>IFSepatu.rb
Harga Sepatu = 600.000 Rupiah
Uang kamu berapa? : 200000
Sudah, Mroyek Dulu Sanah!!
C:\Ruby192\bin\RubyDOJO>
```

Nah disini pake CMD di windows, coba lihat langsung saja kan eksekusi tanpa di Chmod +X dulu, terus hasilnya saya punya uang 700.000 sedangkan harga sepatu 600.000, jadi ya sepatu ajdi milik saya..hahaha

Kemudian contoh lainnya nih, Kita buat program tebak tahun lahir :

*Program menerima **input nama** kamu, dan **umur** kamu. Kemudian program memberikan keterangan kamu masih remaja kalau umur kamu kurang dari 20 tahun dan juga mencetak tebakan tahun lahir kamu. Kalau tidak memenuhi syarat maka program bilang kamu udah dewasa dan mencetak tebakan tahun lahir kamu.*

### **Program 8.2: Tebak Tahun Lahir (IFLahir.rb)**

```
1  #!/usr/bin/ruby
2
3  #menebak tahun lahir
4  time1= Time.new #Variabel berisi waktu terkini dari komputer
5  puts time1.inspect #mencetak waktu secara detail
6
7  puts "Nama kamu siapa?  : "
8  gets #inputan string, namun tidak dimasukan ke var
9
```

```

10 puts " Umur Kamu Berapa? : "
11 @umur = gets.to_i #inputan string diubah menjadi integer
12
13 @tahun = time1.year - @umur #menghitung tahun sekarang dikurang umur
14
15 if @umur < 20 #kalau kurang dari 20 tahun
16   puts " Kamu masih Remaja ya ^__^"
17 else #kalau lebih dari 20 tahun
18   puts "Kamu udah Dewasa, pasti menawan deh!!!"
19 end
20
21 puts "pasti lahir antara tahun " + String(@tahun-1) + " - #@tahun
22 Yaaa :D"

```

Kita mencoba menebak tahun lahir kamu nih, program di barusan bisa nebak tapi memunculkan hasil yang range, misal dari tahun ke berapa ke tahun berapa. Yaudah deh lihat aja yuk hasil eksekusinya :

#### TERMINAL :

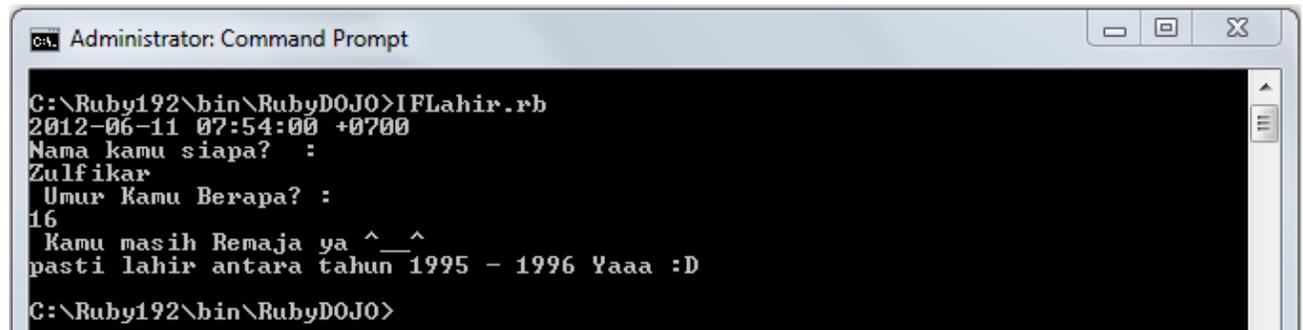
```

hirokakaoshi@hirokakaoshi-Lenovo-G450: ~/Project/RubyDOJO
File Edit View Search Terminal Help
hirokakaoshi@hirokakaoshi-Lenovo-G450:~/Project/RubyDOJO$ chmod +x IFLahir.rb
hirokakaoshi@hirokakaoshi-Lenovo-G450:~/Project/RubyDOJO$ ./IFLahir.rb
Mon Jun 11 07:49:57 +0700 2012
Nama kamu siapa? :
Singgih
Umur Kamu Berapa? :
22
Kamu udah Dewasa, pasti menawan deh!!!
pasti lahir antara tahun 1989 - 1990 Yaaa :D
hirokakaoshi@hirokakaoshi-Lenovo-G450:~/Project/RubyDOJO$ █

```

Di terminal saya memasukkan umur saya 22 Tahun, jadi hasilnya saya dewasa dan menawan hihhi.. ☺ ☺ ☺ 🌸

## CMD :



```
Administrator: Command Prompt
C:\Ruby192\bin\RubyDOJO>IFLahir.rb
2012-06-11 07:54:00 +0700
Nama kamu siapa? :
Zulfikar
Umur Kamu Berapa? :
16
Kamu masih Remaja ya ^__^
pasti lahir antara tahun 1995 - 1996 Yaaa :D
C:\Ruby192\bin\RubyDOJO>
```

kalau yang di CMD saya inputnya umur 16 tahun, jadi remaja deh hasil keluarannya.

Penerima inputan menggunakan **gets** ke var umur artinya, menerima inputan berupa string.

Kemudian **gets** diikuti dengan **to\_i** jadi **gets.to\_i** artinya inputan String tadi di ubah ke dalam bentuk Integer.

Perintah seperti ini dikenal dengan istilah **CASTING**.

Sama hal nya dengan **String(@tahun)**, artinya mengubah bentuk var umur yang tadinya *Integer menjadi String*.

Untuk penggunaan **IF - ELSE - END**, merupakan kondisional dimana ketika di if dinyatakan syaratnya, apabila memenuhi syarat maka eksekusi perintah yang pertama di definisikan, apabila tidak memenuhi maka loncat ke **ELSE** yang memuat perintah lain yang telah didefinisikan juga.

Lihat program [IFLahir.rb](#) , else memuat perintah apabila tidak memenuhi syarat untuk remaja, maka dinyatakan dewasa.

NAMUN!!! coba kalau kita input umur bukan dengan angka??? misal dengan "Tujuhbelas" atau "@#@#" ??? masa hasilnya tetep dibilang dewasa??

Nah ELSE adalah handler / simpangan terakhir suatu syarat..so sebenarnya syaratnya bisa banyak....

bagaimana caranyaaa??? kita pakai **ELSIF** atau dikenal dengan nama *IF Bertingkat (nested if)*



## II. IF - ELSIF - ELSE - END

Bagian ke-2 ini kita akan berlatih menggunakan IF bertingkat (*nested if*) yaitu dengan *elseif*.

Maksudnya adalah, apabila kita memasukan syarat di **IF** pertama, namun tidak terpenuhi maka kita dapat menambah syarat lain di if kedua menggunakan **ELSIF**. **ELSIF** adalah kode ruby untuk *Else IF*.

*Contoh kasus nya kita kembangkan dari IFLahir sebelumnya. Sekarang kita tambahkan apabila umur lebih dari 40 tahun berikan keterangan Sudah tua dan tebak tahun lahirnya, apabila umur lebih dari 100 tahun cetak "Banyak bertobat dari dosa ya! :D". apabila inputan umur bukan angka maka berikan keterangan umur harus angka lebih dari 0. apabila umur benar maka cetak prediksi tahun lahir, apabila bukan angka jangan cetak tahun lahir.*

kita buat programnya..yeaahh :::

### Program 7.2: Tebak tahun lahir season 2 (IFLahir2.rb)

```
1 | #!/usr/bin/ruby
2 |
3 | #menebak tahun lahir
4 | timel= Time.new #Variabel berisi waktu terkini dari komputer
5 | puts timel.inspect #mencetak waktu secara detail
```

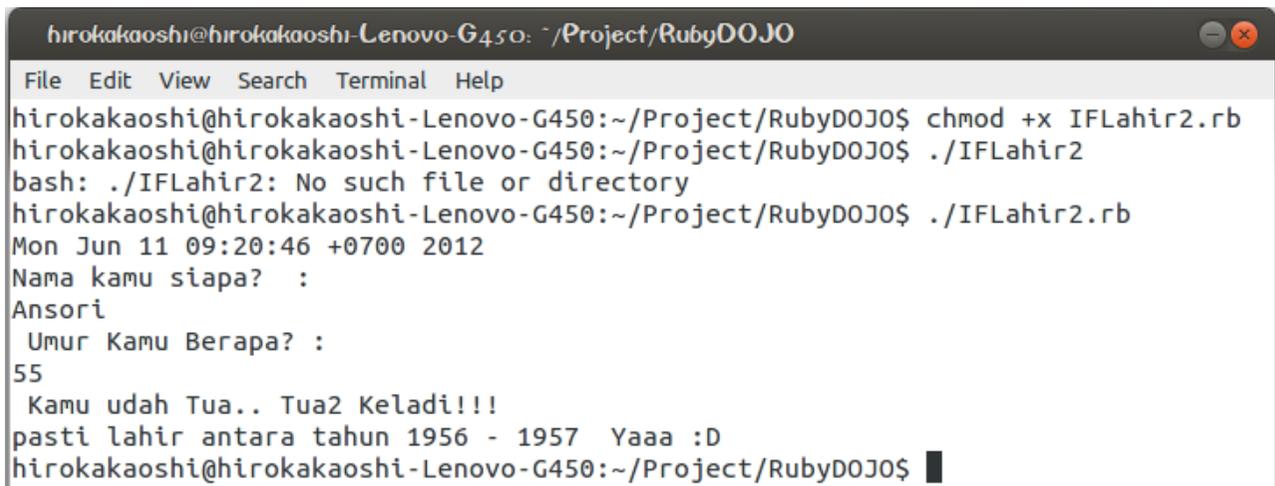
```

6
7 puts "Nama kamu siapa? : "
8 gets #inputan string, namun tidak dimasukan ke var
9
10 puts " Umur Kamu Berapa? : "
11 umur = gets.to_i #inputan string diubah menjadi integer
12
13 @tahun = time1.year - umur #menghitung tahun sekarang dikurang umur
14
15 if umur > 0 and umur <= 20 #kalau kurang dari 20 tahun
16   puts " Kamu masih Imut ya ^__^"
17 elsif umur > 20 and umur <= 40 #kalau kurang dari atau 40 tahun
18   puts "Kamu Dewasa, pasti menawan deh!!!"
19 elsif umur > 40 and umur <= 100 #kalau lebih dari 40 tahun
20   puts " Kamu udah Tua.. Tua2 Keladi!!!"
21 elsif umur > 100 #kalau lebih dari 100 tahun
22   puts "Kamu Segera Banyak bertobat yaa !!! *_*"
23 else #kalau bukan dari semua syarat
24   puts " Hey umur nya angka positif dari angka 1, ga boleh yang
25   laen!!!"
26 end
27
28 if umur != 0 #casting membuat inputan jadi 0
29   puts "pasti lahir antara tahun " + String(@tahun-1) + " - #@tahun
30   Yaaa :D"
31 else
32   puts " Salah inputannya juga yey, ulang lagi!"
33
34 end

```

Mari kita eksekusi kemudian saya jelaskan ^\_\_^

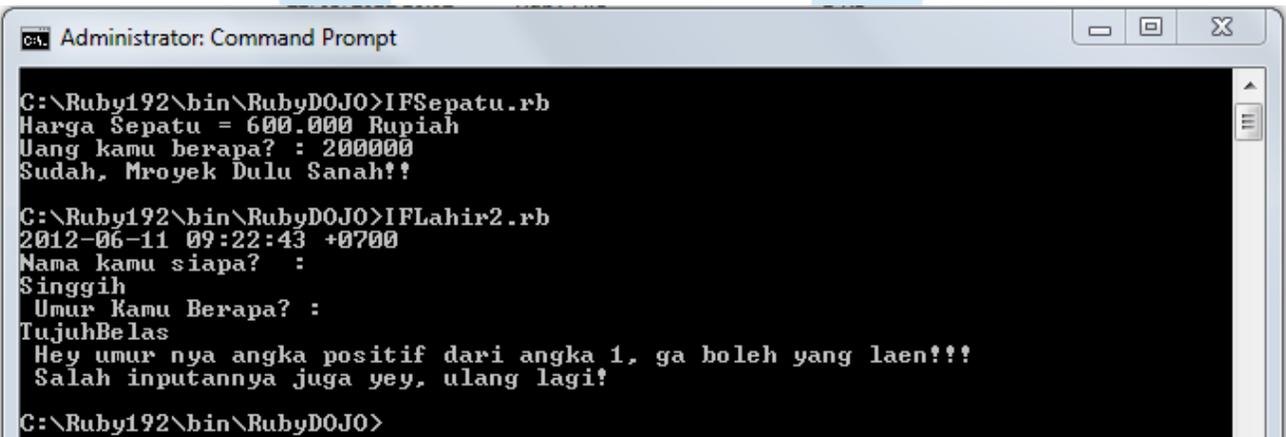
## Terminal :



```
hirokakaoshi@hirokakaoshi-Lenovo-G450: ~/Project/RubyDOJO
File Edit View Search Terminal Help
hirokakaoshi@hirokakaoshi-Lenovo-G450:~/Project/RubyDOJO$ chmod +x IFLahir2.rb
hirokakaoshi@hirokakaoshi-Lenovo-G450:~/Project/RubyDOJO$ ./IFLahir2
bash: ./IFLahir2: No such file or directory
hirokakaoshi@hirokakaoshi-Lenovo-G450:~/Project/RubyDOJO$ ./IFLahir2.rb
Mon Jun 11 09:20:46 +0700 2012
Nama kamu siapa? :
Ansori
Umur Kamu Berapa? :
55
Kamu udah Tua.. Tua2 Keladi!!!
pasti lahir antara tahun 1956 - 1957 Yaaa :D
hirokakaoshi@hirokakaoshi-Lenovo-G450:~/Project/RubyDOJO$
```

terminal saya isi dengan angka dan mencetak TUA dan tebakkan tahun lahir.

## CMD :



```
Administrator: Command Prompt
C:\Ruby192\bin\RubyDOJO>IFSepatu.rb
Harga Sepatu = 600.000 Rupiah
Uang kamu berapa? : 200000
Sudah, Mroyek Dulu Sanah!!

C:\Ruby192\bin\RubyDOJO>IFLahir2.rb
2012-06-11 09:22:43 +0700
Nama kamu siapa? :
Singgih
Umur Kamu Berapa? :
TujuhBelas
Hey umur nya angka positif dari angka 1, ga boleh yang laen!!!
Salah inputannya juga yey, ulang lagi!

C:\Ruby192\bin\RubyDOJO>
```

isi dengan umur yang bukan angka, maka program mencetak ini bukan umur dan syaratnya harus positif dan lebih dari 0.

Program pertama-tama meminta data nama dan umur, kemudian masuk ke bagian pengecekan di **IF**, di **IF** pertama atau syarat pertama mengecek apakah var umur lebih dari 0 dan kurang dari atau sama dengan 20 tahun, bila terpenuhi maka cetak perintah "masih Imut".

*Apabila tidak terpenuhi, maka masuk ke syarat kedua yaitu pengecekan di **ELSIF** apabila umur lebih dari 20 dan kurang dari sama dengan 40 tahun maka cetak "Dewasa".*

*Apabila tidak terpenuhi, masuk ke syarat ke-3, apabila umur lebih dari 40 dan kurang dari 100, maka cetak "Udah Tua".*

*Apabila tidak terpenuhi, masuk ke syarat/pengecekan ke-4, apabila umur lebih dari 100 maka cetak "Segera Bertobat :D".*

*nah apabila seluruh syarat tidak terpenuhi, maka masuk ke handler terakhir, bahwa inputan umur bukan salah satu dari ke empat syarat di atas. Maka cetak ketentuannya.*

Untuk mencetak tebakan tahun lahir, diberikan syarat apabila inputan tersebut bukan angka maka jangan cetak tebakan.

Tapi dengan perintah **umur = gets.to\_i** artinya mengubah seluruh inputan ke var umur yang tadinya *String menjadi Integer*. Jadi apapun yang di input akan berubah jadi 0, maka syarat dalam mengetahui inputan umur angka atau bukan dengan perintah umur != 0.

Begitulah kawan... Masih banyak rahasia yang belum terungkap. Dan kalau kamu kritis pasti bertanya-tanya kenapa aturannya seperti itu..hhi sengaja ah biar pusing hahaha...

Baiklah sekian saja untuk Ruby sekarang, edisi lanjutannya akan membahas berbagai rahasia lagi di **IF** :P . Penggunaan @ sebelum variable di sesuaikan, ada yang memang dibutuhkan dan ada yang tidak :).  
( $\cong \nabla \cong$ )y ( $\cong \nabla \cong$ )y ( $\cong \nabla \cong$ )y SELAMAT BELAJAR...



Referensi :  
Programming Ruby, 2nd Edition | pragmatic Programmer's Guide  
[http://www.tutorialspoint.com/ruby/ruby\\_if\\_else.htm](http://www.tutorialspoint.com/ruby/ruby_if_else.htm)

---

### III. Rangkuman

Penggunaan **IF - ELSE - END** dan **IF - ELSIF - ELSE - END** di Ruby tidak beda dengan bahasa pemograman lainnya, sederhana dan berfungsi sama sebagai pengecekan dengan syarat tertentu (kondisional). Apa saja sih yang sudah kita pelajari di artikel kali ini :

- ⤴ Control Structures bagian IF.
- ⤴ IF sederhana dan IF bertingkat.

- ▲ Penggunaan Time dari system.
- ▲ Mengubah tipe variable dengan Casting.

Selanjutnya, saya niatnya sih membahas kembali penggunaan **If** dengan memberikan kasus yang cukup semakin menyulitkan. Melatih Logic dulu deh hhe  
≥ ^∪ ^≤

**BERLATIH**

**8**

**BERLATIH**

Horrey, saatnya latihan...berikut ini bagian latihan yang kita perlukan untuk memantapkan penguasaan algoritma dan Ruby. Berikut latihan menggunakan IF dan nested IF.

- Buat Program yang menerima inputan/masukan dari pengguna/user berupa Nama dia, kemudian cek panjang nama nya. Apabila panjangnya lebih dari 30 karakter maka cetak " Jumlah Karakter nama kamu lebih dari 30 karakter", apabila tidak maka cetak namanya, selanjutnya hitung jumlah huruf vokal yang ada di dalam namanya.

*Petunjuk : gunakan .length, dan array.*

**CHALLENGE ACCEPTED**



---

Apabila penulis melakukan ketelodoran baik kata-kata maupun kode yang kurang baik mohon dimaafkan dan diingatkan.

Penulis berharap semua ilmu yang disebar dapat bermanfaat dan mempermudah dalam mempelajari bahasa pemrograman dan algoritma pembaca.

*Salam, hirokakaoshi | nyankoder's Ruby.*

*Masih dengan IF ... :D dan mungkin Case :P*

# //NyanTerview

# jago{coding}



Nyankoders sekalian. Mulai edisi kali ini, kita punya sebuah rubrik baru lho, yaitu NyanTerview. Di rubrik ini, kita bakal mewawancarai founder-founder dari sebuah startup yang lagi berkembang di Indonesia yang berkaitan dengan dunia IT, atau orang-orang yang telah berkarya dalam bidang tertentu di dunia It.

Nah, pada kesempatan kali ini, kita bakal ngewawancarai founder dari JagoCoding.com yaitu kang Cecep Yusuf, S.Kom atau biasa akrab dipanggil Kang Cheyuz. Mau tau lebih banyak tentang JagoCoding.com, yuk ikutin wawancaranya.



## Cecep Yusuf, S.Kom

Founder situs Jagocoding.com



### Kapan sih berdirinya Jagocoding.com?

Jagocoding.com berdiri pada tanggal 3 Mei 2012. Jadi usia sampai saat ini baru satu bulan. Suatu perkembangan yang baik dengan satu bulan ini Jagocoding.com dapat memiliki visitor dengan IP berbeda 90-160 orang per harinya. Dengan fanspage yang sudah memiliki fans 2000 orang lebih.

### Inspirasi dibuatnya Jagocoding.com ini darimana?

Desain website Jagocoding.com terinspirasi dari suatu website bernama css-tricks.com yang mana situs ini menyediakan tutorial-tutorial CSS yang berkualitas dan mempunyai "snippets" (kode siap pakai) sama seperti Jagocoding.com. Namun, di situs ini tutorial hanya dibuat oleh admin, bukan oleh pengguna lain. Untuk jumlah view, komentar, dan love terinspirasi dari situs dribbble.com, tetapi secara konseptual sangat berbeda dengan jagocoding.com. Dan situs dribbble.com itu sendiri adalah situs komunitas desain grafis. Untuk fitur lainnya seperti follow dan lain-lain, itu mengikuti social media yang sedang marak saat ini.

### Boleh dijelaskan gak Kang, Ada fitur apa saja di Jagocoding.com ini?

Fitur utama pada situs Jagocoding.com tentu saja membuat tutorial. Tutorial dibuat oleh pengguna yang sudah mendaftarkan diri di situs Jagocoding.com, yang kemudian rela membagikan ilmunya ke publik untuk dibaca oleh pengguna/pengunjung jagocoding.com lainnya. Adapun fitur-fitur lainnya adalah:

- **Follow**, yaitu sejenis subscribe, berlangganan tutorial terhadap orang yang kita follow. Jika kita memfollow "James", maka kita akan selalu mendapatkan notifikasi bahwa James telah membuat tutorial baru.
- **Comments and Discussion**, pengguna dapat memberikan komentar dan diskusi di setiap tutorial.
- **Love**, pengguna dapat menyukai tutorial dan dilihat siapa yang menyukai tutorial tersebut.
- **Download PDF**, pengguna dapat mendownload versi PDF dari setiap tutorial.
- **Message**, fitur untuk mengirim pesan sesama pengguna yang difollow atau memfollow user.
- **Notifications**, fitur untuk menampilkan notifikasi ketika ada perubahan/update pada suatu tutorial, komentar, dan sebagainya.

- **Ratings**, fitur untuk memberikan rating terhadap suatu tutorial. Kenapa dibuat? Karena pengguna ingin tahu mana tutorial yang paling berkualitas dan mana tutorial yang kurang begitu bagus.
- **Snippets**, yaitu koleksi kode-kode siap pakai (tinggal copy paste).
- Dan beberapa fitur kecil lainnya yang tidak dapat disebutkan satu persatu.

### **Keunikannya Jagocoding.com dibanding dengan website – website belajar pemrograman lainnya apa nih Kang?**

Keunikannya sebenarnya banyak, cuma saya sebutkan yang paling terlihat berbeda dengan tutorial yang lainnya, yaitu: user dapat berlangganan tutorial dari user lain, dan dapat berdiskusi secara dinamis pada suatu tutorial. Fitur lainnya yang membedakan dan murni ide dari Jagocoding.com adalah download PDF pada setiap tutorial. Dengan begitu, user dapat mempelajari tutorial-tutorial Jagocoding.com secara offline. User juga dapat mengkolleksi tutorial-tutorial yang berkualitas dengan download PDF yang diinginkan.

### **Founder Jagocoding.com ini selain Kang Cheyuz siapa lagi Kang?**

Selain saya sebagai Founder, Co-founder Jagocoding.com adalah Titan Firman, dialah yang memberi ide-ide unik dan membantu pembuatan program untuk Jagocoding.com. Dan saya juga merekrut Irvan Riswanto sebagai orang ketiga yang berperan penting untuk situs Jagocoding.com.

### **Visi dan misi Jagocoding.com sendiri apa Kang?**

#### **Visi:**

Membuat para programmer Indonesia semakin cerdas dan berkembang di dalam dunia IT, khususnya programming. Selain itu juga Jagocoding.com ingin membuat bangsa Indonesia selalu bangga akan hasil karya sendiri dan selalu mengikat silaturahmi antar orang di bidang yang sama.

#### **Misi:**

- Membuat fasilitas untuk membuat tutorial yang interaktif, dapat didiskusikan, dibagikan, dan dinilai oleh semua pengguna.

- Menyediakan fitur yang unik dan terbaik bagi pengguna supaya pengguna nyaman menggunakan situs Jagocoding.com.
- Membangun sosial yang baik antar programmer melalui diskusi interaktif pada suatu tutorial.

**Rencana pengembangan kedepan Jagocoding.com sendiri seperti apa Kang? Kan yang udah like Jagocoding.com udah sekitar 2500-an nih. hehe**

Sudah banyak ide-ide brilian yang sudah kami persiapkan untuk ke depannya. Kalo dikasi tau sekarang ga seru dong ga kejutan. Hehe..

**Kenapa Jagocoding.com ini membahas PHP, Javascript, HTML, CSS, dan JAVA?**

Jagocoding.com tidak hanya berfokus pada bahasa pemrograman di atas, tetapi juga bahasa pemrograman yang lain, bahkan semua bahasa pemrograman. Mungkin karena Jagocoding.com baru berumur 1 bulan, jadi kontributor yang baru terdaftarlah yang expert dibidang itu. Mungkin para kontributor Nyankod juga bisa berkontribusi di Jagocoding.com. Saya rasa di Nyankod juga banyak ahli perl, python, yuk berkontribusi.. :D dan share ilmunya..

**Target Jagocoding.com ini sendiri siapa Kang ? Pelajar, Mahasiswa, Dosen atau Pengguna Umum?**

Target Jagocoding.com adalah semua kalangan dengan segala usia, pengguna umum yang ingin belajar memprogram dan belajar bahasa pemrograman.

**Kira – kira kapan nih mau launching di Kampus atau institusi lainnya ? Atau sudah pernah melakukan presentasi atau promosi sebelumnya Kang?**

Karena produk Jagocoding.com ini masih beta, maka kami belum pernah melakukan launching. Akan tetapi, diperkirakan dalam waktu dekat ini kami akan melakukan launching. Untuk bulan pastinya, belum bisa ditentukan, soalnya kami masih harus mengumpulkan user dan konten terlebih dahulu sebelum launching, hehe.

**Bagaimana nih tanggapan Jagocoding.com dengan munculnya berbagai website – website serupa Jagocoding.com?**

Tidak masalah bagi kami. Digitalisasi itu sangat mudah dijiplak, dicopy, apalagi dengan media internet yang begitu bebasnya untuk melakukan sebuah plagiarisme. Hal tersebut tidak masalah bagi kami, karena kami akan selalu memberikan ide original untuk user. Ide-ide inilah yang akan menjadi sesuatu yang unik bagi kami yang tidak ada di website lainnya.

**Bagaimanakah cara Jagocoding.com mengelola pembaca dan kontributor?**

Kami dari tim jagocoding mengelola pembaca dan kontributor dengan cara selalu meluncurkan fitur-fitur yang unik, dan selalu menerbitkan tutorial-tutorial yang fresh dan original (dan tentunya berguna bagi pembaca).

**Kalau yang paham coding sebelumnya kan gak masalah yah Kang, langsung belajar apa yang dia suka yang disediakan Jagocoding.com. Kalau yang belum pernah coding sebelumnya tapi ingin belajar hal – hal yang dibahas di Jagocoding.com, bagaimana memulainya Kang?**

Tujuan dari situs jagocoding bukan untuk mengajarkan pemrograman dari awal sampai dengan akhir. Jika pengajaran, mungkin bisa diperkuliahan saja atau di pendidikan formal lainnya. Di Jagocoding.com ilmu dasar tersebut dikembangkan, kebanyakan tutorial-tutorial di Jagocoding.com bukan teori yang ada di pendidikan formal tapi berdasarkan pengalamandan kasus yang biasa ditemukan di dunia kerja.

**Boleh memberikan pesan dan saran untuk rekan sejalur Jagocoding.com?**

Tetaplah berkreasi dengan ide-ide yang original (tanpa plagiarisme)

**[Perl]**



Kresna Galuh D. Herlangga  
@kresnagaluh  
kresnagaluh@gmail.com  
<http://kresnagaluh.com>



# Ulangan Episode 1

Pada suatu hari, ada seorang raja yg mempunyai seorang anak, dan menjelang malam hari mau tidur anak raja itu pengen diceritain oleh ayahnya yang seorang raja. Mau tau ceritanya???

Ceritanya gini, pada suatu hari, ada seorang raja yang mempunyai seorang anak, dan menjelang malam hari mau tidur, anaknya pengen diceritain oleh ayahnya yang seorang raja. Mau tau ceritanya??

Pada suatu, ada seorang raja yg punya seorang anak, dan menjelang malam hari mau tidur anak raja itu pengen diceritain oleh ayahnya yang seorang raja. Mau tau ceritanya???

Berapa lapis?? Ratusan.... Lebih....  
Berapa lapis?? Ratusan.... Lebih....  
Berapa lapis?? Ratusan.... Lebih....

Hehehe... kok diulang-ulang?? Kok diulang-ulang??

Malem nyankoders sekalian (klo disini sih malam), gimana kabarnya nih?? Nggak kerasa udah 3 minggu lagi berlalu semenjak edisi ke-7 terbit. Gimana belajarnya coy?? Jangan sampe lupa ya, sama pembahasan edisi sebelumnya.

Oh ya coy, pernah nggak diceritain dongeng kayak di atas?? Yang raja n anaknya itu lho?? Yang minta diceritain itu... Gimana rasanya pas pertama kali diceritain dongeng kayak gitu?? Pasti sebel banget ya... Ya begitulah, saya juga pas

pertama kali diceritain, cerita kayak gitu agak sebel sih. Saya pertama kali diceritain itu pas kelas 4 SD. Dah lama juga ya...

Yang saya mau tekankan bukanlah dongengnya, tapi pengulangannya. Ntar deh, saya ceritain lebih lanjut tentang ini. Tapi sebelum kita mulai membahas materi, kita review dulu deh tentang materi sebelumnya. Biar nggak lupa coy...

1. Apa aja sih, bentuk-bentuk statment percabangan??
2. Gimana sih bentuk statment darei elsif?
3. Fungsi unless buat apa sih??

Seperti biasa ya, jangan dulu lanjutin pelajaran klo belum bisa jawab pertanyaan-pertanyaan di atas. Selamat berjuang.

## Ulangan Apaan??

---

Ulangan apaan coy?? Ulangan harian?? Hehehe... Sederhananya sih, yang dimaksud dengan pengulangan ya suatu tindakan yang berulang-ulang. Ato kalo di dalam pemrograman yang disebut dengan pengulangan atau looping adalah suatu aksi yang dilakukan secara berulang dalam kondisi tertentu.

Dalam looping, biasanya program akan terus berputar selama keadaan tertentu, dan akan keluar dari putaran tersebut bila telah memenuhi syarat tertentu atau tidak lagi memenuhi syarat tertentu. Ada loop yang berhenti ketika keadaan telah bernilai false, namun ada juga loop yang akan berhenti jika keadaan telah bernilai true.

Oh ya, dalam prakteknya, kadang ada sebuah program akan terus berputar tanpa henti sehingga butuh dikeluarkan secara paksa untuk menghentikan

perputarannya. Selain itu juga ada program yang memiliki loop, namun sama sekali tidak pernah masuk dalam loop tersebut.

Untuk loop yang ada di Perl, terdiri dari beberapa bentuk, yaitu: while, until, for, foreach, do while dan do until.

## While

---

Baik. Mari kita mulai dengan while terlebih dahulu. Sebenarnya mungkin walaupun kita belum pernah kenal sama struktur pengulangan while, tapi mungkin dari arti kata while itu sendiri juga kita udah paham kalau ini akan digunakan sebagai bentuk pengulangan.

Adapun struktur dari pengulangan while adalah seperti berikut:

```
while ( condition ) { action }
```

Secara struktur mungkin agak sedikit mirip dengan syntax dari percabangan if yang udah kita bahas di edisi sebelumnya. Cuma mirip doang kok. Adapun kalo dari segi penggunaan jelas berbeda banget.

Di dalam pengulangan while, pertama-tama akan dilakukan pengecekan terhadap suatu kondisi. Apabila kondisi memenuhi persyaratan untuk masuk ke dalam loop, maka akan langsung masuk ke dalam loop, kemudian akan dijalankan sebuah aksi, setelah itu akan kembali dilakukan pengecekan kondisi lagi, bila masih memenuhi syarat pengulangan, maka akan dilakukan aksi. Begitu seterusnya, akan terus diulang dan dilakukan aksi selama kondisi masih memenuhi persyaratan pengulangan/loop. Pengulangan hanya akan berhenti bila kondisi tidak sesuai lagi dengan persyaratan loop. Atau dalam bahasa yang lebih keren yaitu, selama pengulangan akan dilakukan selama kondisi bernilai true, dan akan dihentikan setelah kondisi bernilai false.

Biar mudah dipahami, mari kita pelajari lewat contoh. So, coba tulis dan eksekusi program berikut ini:

### Program: while1.pl

```
1  #!/usr/bin/perl
2  # nama program : while1.pl
3
4  $countdown = 7;
5
6  while ($countdown > 0) {
7      print "Hitung mundur: $countdown\n";
8      $countdown--;
9  }
10
11 print "Ini di luar loop";
12
```

Output:

```
$ perl while1.pl
Hitung mundur: 7
Hitung mundur: 6
Hitung mundur: 5
Hitung mundur: 4
Hitung mundur: 3
Hitung mundur: 2
Hitung mundur: 1
Ini di luar loop
$
```

Penjelasannya gini:

Pertama-tama kita punya sebuah variabel bernama \$countdown yang bernilai 7. Selanjutnya dilakukan pengecekan kondisi oleh while dengan persyaratan, pengulangan/loop hanya akan dilakukan jika variabel \$countdown bernilai lebih besar dari 0. Karena saat ini variabel \$countdown bernilai 7, artinya program memenuhi persyaratan buat masuk ke dalam loop. Selanjutnya adalah program akan menampilkan tulisan "Hitung mundur: " dan menampilkan nilai variabel \$countdown saat ini, yaitu 7. Setelah itu akan dilakukan pengurangan nilai dari variabel \$countdown oleh syntax \$countdown--, yang artinya nilai variabel \$countdown akan dikurangi satu. Atau bisa juga ditulis: \$countdown = \$countdown - 1. Setelah itu maka nilai \$countdown adalah 6. Selanjutnya akan kembali ke atas, yaitu melakukan pengecekan terhadap nilai dari \$countdown. Bila masih memenuhi persyaratan loop, maka akan kembali memasuki loop, namun bila tidak maka akan keluar dari loop dan mengeksekusi syntax selanjutnya. Begitulah selanjutnya. Intinya, program akan terus berada di dalam loop, jika kondisi masih bernilai true. Artinya program akan bernilai false apabila \$countdown sudah tidak lagi lebih besar dari 0. Kesimpulannya program akan keluar dari loop jika nilai \$countdown bernilai 0.

Karena pengecekan terhadap kondisi dilakukan di awal, maka akan ada saatnya dimana kita tidak akan pernah masuk ke dalam pengulangan. Ya, hal itu terjadi dimana kondisi sejak awal tidak memenuhi persyaratan buat masuk ke dalam loop. Atau dalam kata lain, kondisi bernilai false.

Sekarang, mari kita modifikasi sedikit program tadi, jadi seperti berikut ini:

### **Program: while2.pl**

```
1 | #!/usr/bin/perl
2 | # nama program : while1.pl
3 |
4 | $countdown = 7;
5 |
6 | while ($countdown < 0) {
7 |     print "Hitung mundur: $countdown\n";
```

```
8     $countdown--;  
9     }  
10  
11     print "Ini di luar loop";  
12
```

Output:

```
$ perl while1.pl  
Ini di luar loop  
$
```

Tuh kan, program sama sekali nggak masuk ke dalam loop. Tau kan alasannya?? Ya, karena kali ini persyaratannya adalah loop akan dijalankan bila \$countdown bernilai kurang dari 0. Sedangkan saat ini \$countdown bernilai 7. Artinya syarat kondisi loop tidak terpenuhi. So, akan langsung lanjut ke perintah di bawahnya yaitu menulis kalimat "Ini di luar loop".

## Until

---

Bisa dibilang kalau until adalah lawan dari while. Hehehe... Kok bisa?? Bisa dong. Sebenarnya sih klo secara struktur penulisan, antara while dan until itu nggak ada bedanya. Hanya saja, kalau di while itu, loop akan dijalankan selama kondisi bernilai true dan akan berhenti ketika kondisi bernilai false. Sedangkan kalau di until, loop akan dijalankan selama kondisi bernilai false, dan akan berhenti setelah kondisi bernilai true. Adapun format penulisannya adalah seperti berikut:

```
until ( condition ) { action }
```

Biar lebih mudah buat dipahami, silahkan praktekin kode berikut ini:

### Program: k-until\_anak.pl

```
1  #!/usr/bin/perl
2  # nama program : k-until_anak.pl
3
4  $countdown = 7;
5
6  until ($countdown <= 0) {
7      print "Hitung mundur: $countdown\n";
8      $countdown--;
9  }
10
11 print "Ini di luar loop";
12
```

Output:

```
$ perl k-until_anak.pl
Hitung mundur: 7
Hitung mundur: 6
Hitung mundur: 5
Hitung mundur: 4
Hitung mundur: 3
Hitung mundur: 2
Hitung mundur: 1
Ini di luar loop
$
```

Tuh kan, outputnya sama sama program while1.pl

Intinya, secara struktur logika samalah antara while dengan until. Hanya yang membedakan disini adalah, loop akan dieksekusi selama kondisi bernilai false. Terlihat

kan dri program di atas, \$countdown bernilai 7. Sedangkan syarat kondisi adalah \$countdown sama dengan atau lebih kecil dari 0. Karena kondisi masih bernilai false, maka loop akan dieksekusi. Intinya gitu. Sederhana bukan??

## For

---

Kalau kamu pernah belajar bahasa pemrograman yang laen semisal PHP, C, C++ atau Java, mungkin kamu pernah tau tentang pengulangan For. Nah, Perl juga punya pengulangan for. Secara umum logikanya sama kok.

Adapun buat kamu yang baru belajar, selow aja ya, pasti bakalan dijelasin kok. Jadi nggak usah takut nggak ngerti. Nggak ribed-ribe amat kok.

Begini nih format pengulangan for:

```
for (init_expression; test_expression; step_expression) {  
    action  
}
```

Init expression adalah ekspresi pertama, atau bisa dibilang keadaan pertama. Test expression adalah pengecekan kondisi (true atau false), dan juga batas kondisi apakah akan melanjutkan pengulangan atau pengakhirinya. Step expression adalah pernyataan yang dieksekusi ketika test expression bernilai true. Selanjutnya action adalah aksi-aksi yang nantinya bakal dilakukan ketika pengulangan.

Biar gampang, mari kita pelajari dengan contoh. Sebelumnya, coba perhatikan dulu program while1.pl dan k-until\_anak.pl. Nah, program itu kalo ditulis dalam bentuk for adalah seperti berikut:

### Program: for\_loop.pl

```
1  #!/usr/bin/perl
2  # nama program : for_loop.pl
3
4  for ($countdown = 7; $countdown > 0; $countdown--) {
5      print "Hitung mundur: $countdown\n";
6  }
7
8  print "Ini di luar loop";
9
```

### Output:

```
$ perl for_loop.pl
Hitung mundur: 7
Hitung mundur: 6
Hitung mundur: 5
Hitung mundur: 4
Hitung mundur: 3
Hitung mundur: 2
Hitung mundur: 1
Ini di luar loop
$
```

Sama kan hasilnya, bahkan lebih sederhana. Ekspresi pertama menyatakan bahwa \$countdown bernilai 7. Ekspresi kedua menyatakan bahwa looping akan dieksekusi jika \$countdown bernilai lebih besar dari 0. Ekspresi ketiga menyatakan bahwa setiap kali syarat ekspresi kedua terpenuhi maka akan dilakukan pengurangan 1 nilai \$countdown yaitu dengan \$countdown--. Apabila ekspresi kedua tidak lagi dipenuhi maka program akan keluar dari loop dan akan mengeksekusi perintah di bawah loop, yaitu "Ini di luar loop".

# Foreach

---

Satu lagi nih loop yang ada di Perl, yaitu foreach. Foreach ini adalah pengulangan yang menggunakan list dan array. Masih inget kan tentang list dan array??

Baik, silahkan praktekin program berikut ini:

## Program: for\_loop.pl

```
1  #!/usr/bin/perl
2  # nama program : foreach_nih.pl
3
4  $number;
5
6  foreach $number (1 .. 7) {
7      print "Number: $number\n";
8  }
9
10 print "Ini di luar loop";
11
```

Output:

```
$ perl foreach_nih.pl
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5
Number: 6
Number: 7
$
```

Gimana gampang kan coy?? Jangan lupa, materi ini harus dipahami bener-bener ya jou, ini bakalan bermanfaat banget buat kedepannya. Mungkin buat edisi ke-8 ini sampe disini dulu coy. Kita ketemu di edisi-9 mendatang. Tentunya dengan melanjutkan pembahasan kita di Ulangan Episode 2.

Oke cong, sampe ketemu lagi di edisi ke-9 nanti. Tetep semanget, dan jangan lupa sarapan pagi, biar nggak sakit perut. Cemungudh eaaa...

***Every time is learning....***

@KresnaGaluh  
KresnaGaluh.com

Sumber:

Begining Perl Third Edition; James Lee; Apress  
Learning Perl; Randal L. Schwart, Tom Phoenix & Brian D Foy; O'REILLY

Oh ya, nih soal ulangnya. Dikerjain ya, biar nguji pemahaman di edisi ini.

Buatlah sebuah program buat untuk menuliskan angka ganjil dari 1 sampai dengan n. Adapun nilai n akan dimasukan oleh user dengan <STDIN>. Misalnya user memasukan nilai 100. Maka, akan tampil di layar barisan bilangan ganjil dari 1 sampai 99 yang dipisahkan dengan spasi. Untuk membuat pengulangannya, silahkan pilih yang mana aja yang sesuai selera kamu dari beberapa bentuk pengulangan yang ada di edisi ini. Sebagai bantuan, gunakanlah pemilihan dan operator sisa hasil bagi buat membedakan ganjil dan genap.

# [JavaScript]



Toni Haryanto  
@toHaryan  
toha.samba@gmail.com  
<http://toniharyanto.cs.upi.edu>



# Menutup Celah Error

*Pernah Hal yang paling menyenangkan bagi seorang programmer adalah ketika ia menemukan error pada program dan berhasil memperbaikinya. Setuju?*

Bicara tentang pemrograman pasti tidak akan lepas dengan error. Sebagai programmer, kita mesti mampu membuat aplikasi yang stabil. Artinya tidak ada kesalahan yang muncul dari sistem. Setiap peluang kesalahan atau error (untuk selanjutnya saya pake istilah error aja ya) mesti sudah diperhitungkan oleh kita sang pembuat program. Kalaupun memang ada error yang muncul disebabkan input dari pengguna yang tidak sesuai dengan permintaan program, maka pesan error yang muncul mesti dapat dimengerti oleh pengguna. Oke, mungkin pembahasan ini nampaknya akan cukup rumit, tapi percayalah, pembahasan ini sangat sederhana. Kamu hanya perlu mengenal 3 istilah baru disini. Ditambah lagi, kamu sudah mempelajari dasar-dasar pemrograman JavaScript di majalah NyanKod edisi sebelumnya. Pokoknya gampang pisan lah! \*alamatSusahIniPembahasan

# "Coba kode, dan tangkap errornya"

Sintaks pertama yang akan kita bahas adalah try..catch. Mari kita lihat struktur penulisannya.

```
try {  
    //baris kode yang akan dicek errornya  
}  
catch(err) {
```

```
    //baris kode untuk penanganan error  
}
```

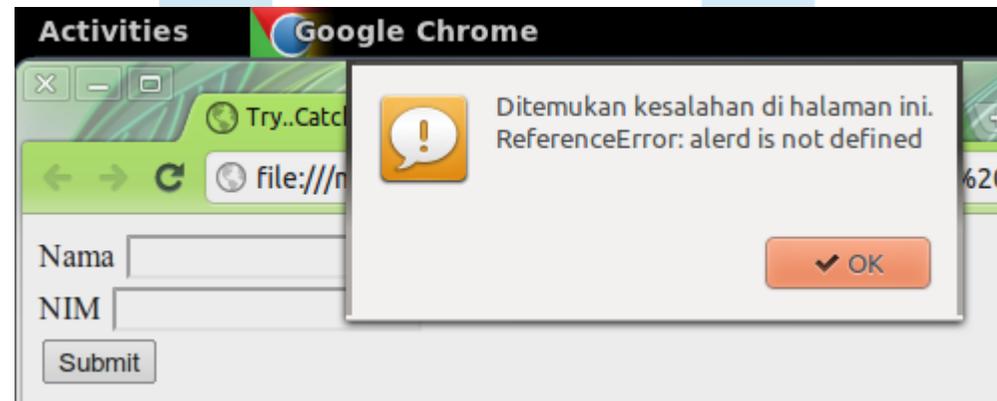
Pada penulisan try..catch, terdapat dua blok area yang dapat kita isikan kode. Blok pertama yaitu blok kode try, yaitu blok dimana kita mengisi baris-baris kode yang ingin dicek error. Blok kedua adalah blok catch yaitu blok dimana kita menuliskan baris kode yang akan dieksekusi apabila ditemukan baris kode yang error di blok try.

Untuk lebih jelasnya mari kita lihat contoh program berikut.

```
1 <html>  
2 <head>  
3 <title>Try..Catch</title>  
4 <script type="text/javascript">  
5  
6 function submitData(){  
7     var nama = document.getElementById('nama').value;  
8     var nim = document.getElementById('nim').value;  
9  
10    try  
11    {  
12    alert("Nama : " + nama + "\nNIM : " + nim);  
13    }  
14    catch(err)  
15    {  
16    txt="Ditemukan kesalahan di halaman ini.\n";  
17    txt+=err;  
18    alert(txt);  
19    }  
20 }  
21 </script>  
22 </head>  
23 <body>  
24 Nama <input type="text" id="nama" />  
25 <br />
```

```
26 NIM <input type="text" id="nim" />
27 <br />
28 <input type="submit" value="Submit" onclick="submitData()" />
29 </body>
30 </html>
31
```

Pada baris ke-11 kita memanggil fungsi alert tapi ceritanya salah ketik, jadi alerd. Disini kita mencoba membuktikan apakah error bisa ditangkap oleh mekanisme try..catch ini. Ketika aplikasi dijalankan dan Kamu menekan tombol Submit, mestinya pesan yang muncul adalah Nama dan NIM yang kamu inputkan di dalam field. Tapi, karena tadi kita salah menetik fungsi, mestinya alert jadi alerd, maka otomatis fungsi tersebut tidak dikenali oleh JavaScript dan dianggap sebagai error. Maka blok kode catch-lah yang dijalankan. Pada contoh di atas, kita diberi tahu, bahwa fungsi alerd tidak dikenali.



Paham kan? Sudah, begitu saja pembahasan tentang try..catch. Hehehe.. Yang bener? Iya bener. Tapi ada satu pembahasan lagi untuk melengkapi penanganan error di JavaScript. Yaitu throw.

## # "Lemparkan Errornya"

Pada try..catch, JavaScript mencari baris kode yang error sesuai dengan aturan JavaScript. Adakalanya kita memerlukan penanganan suatu kondisi yang tidak dianggap sebagai suatu error oleh JavaScript padahal menurut kita itu mesti dianggap error –setidaknya dari sudut pandang user. Misalnya tentang validasi form. Pada contoh sebelumnya kita bebas mengisi apapun di kolom Nama dan NIM. Padahal mestinya ada aturan yang mesti kita terapkan untuk pengisian kolom tersebut. Misalnya kolom Nama hanya boleh diisi oleh huruf saja, dan NIM hanya boleh diisi oleh angka saja. Diluar aturan itu kita anggap sebagai error. Maka kita dapat mendefinisikan sendiri error kita dengan menggunakan throw. Contohnya adalah sebagai berikut.

```
1 <html>
2 <head>
3 <title>Throw</title>
4 <script type="text/javascript">
5 function submitData(){
6     var nama = document.getElementById('nama').value;
7     var nim = document.getElementById('nim').value;
8
9     try
10    {
11        if(nama==null || nama=="")
12            throw "namaKosong";
13        else if(nim==null || nim=="")
14            throw "nimKosong";
15        else if(! isNaN(nama))
16            throw "namaSalah";
17        else if(isNaN(nim))
18            throw "nimSalah";
19        else
20            alert("Nama : " + nama + "\nNIM : " + nim);
21    }
```

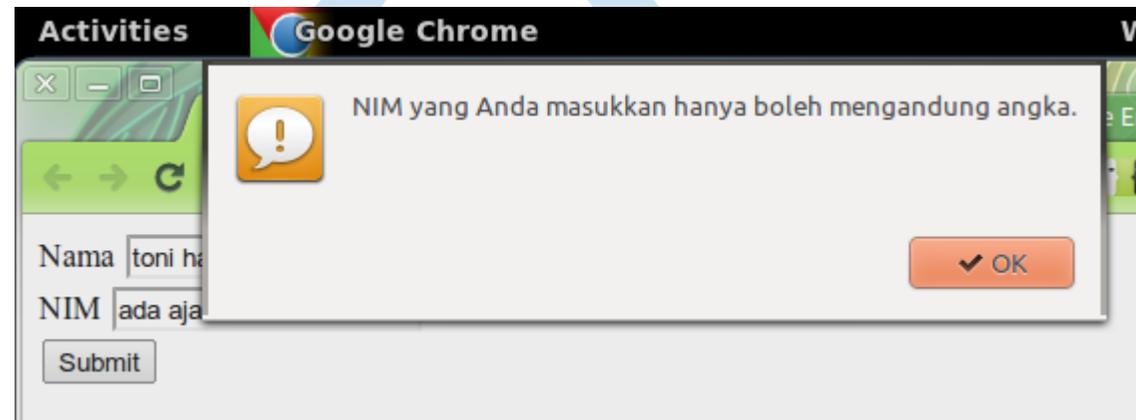
```

22     catch(err)
23     {
24         txt = err;
25         if(err == "namaKosong")
26             txt="Nama tidak boleh kosong.\n";
27         else if(err == "nimKosong")
28             txt="NIM tidak boleh kosong.\n";
29         else if(err == "namaSalah")
30             txt="Nama yang Anda masukkan tidak boleh mengandung
31 angka.\n";
32         else if(err == "nimSalah")
33             txt="NIM yang Anda masukkan hanya boleh mengandung
34 angka.\n";
35         alert(txt);
36     }
37
38 }
39 </script>
40 </head>
41 <body>
42 Nama <input type="text" id="nama" />
43 <br />
44 NIM <input type="text" id="nim" />
45 <br />
46 <input type="submit" value="Submit" onclick="submitData()" />
47 </body>
48 </html>
49

```

Pada baris ke-11 sampai 18 kita mendefinisikan error kita. Contohnya pada baris ke-11 kita mengecek apabila kolom nama bernilai null atau kosong, maka kita melemparkan error dengan nama "namaKosong". Begitu juga dengan pengecekan selanjutnya. Nanti apabila salahsatu kondisi terpenuhi, maka error akan langsung dilemparkan oleh sintaks throw ke blok kode catch.

Di blok kode catch, pertama-tama caption error diisi dahulu oleh error yang terdeteksi oleh JavaScript, seperti pada baris ke-24. Baris ke-25 sampai 32 adalah pemilahan error yang telah kita definisikan di blok kode try. Dan di baris ke-33, apapun pesan errornya, maka tampilkan di dalam jendela alert.



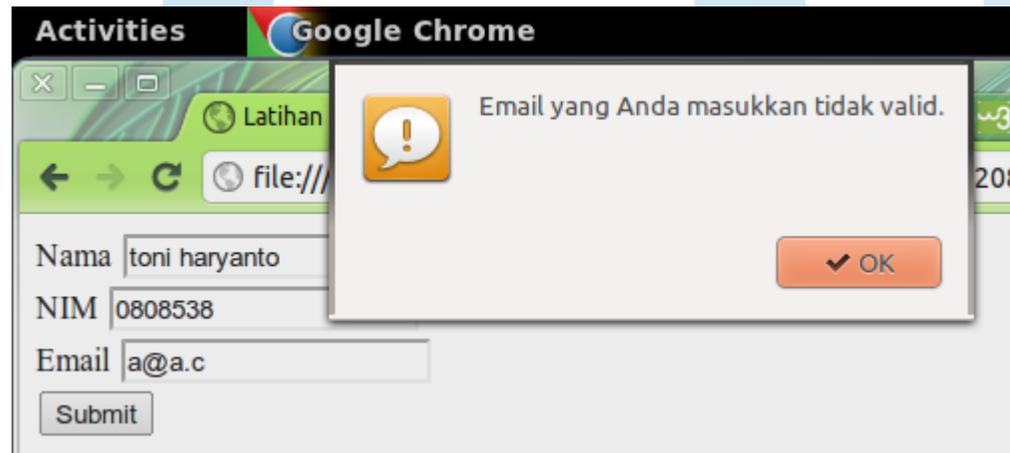
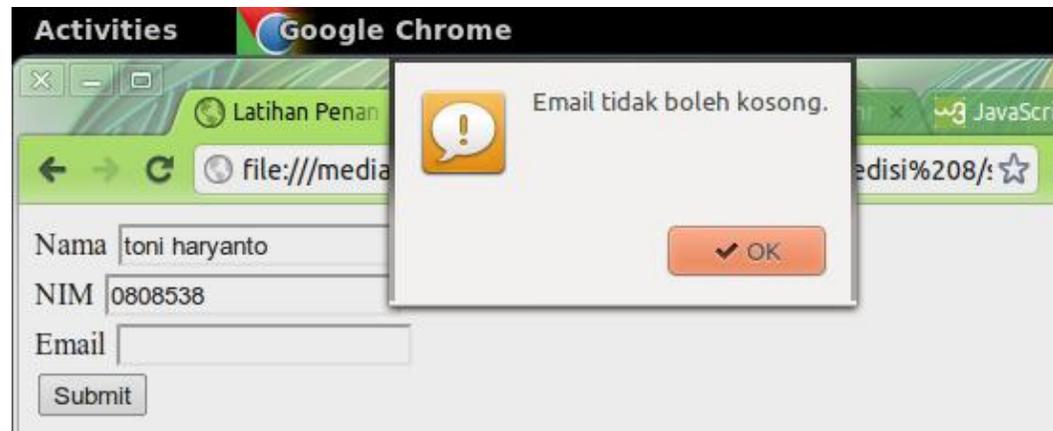
Sip, gampang kan? Apa saya bilang. Hehe.. Intinya sesederhana apapun program kita, sedapat mungkin kita mempersempit peluang terjadinya error. Paling tidak, ketika error tidak dapat dihindari, kita sudah menunjukkan bahwa kita sudah memperhitungkan setiap error dengan menampilkan pesan kesalahan.

```
//Pesan pinggir  
// Baik try..catch maupun throw, mesti ditulis dengan huruf kecil semua. Kalo  
tidak maka jadi error deh. Masa penangkap error-nya malah error juga.. hehe..
```

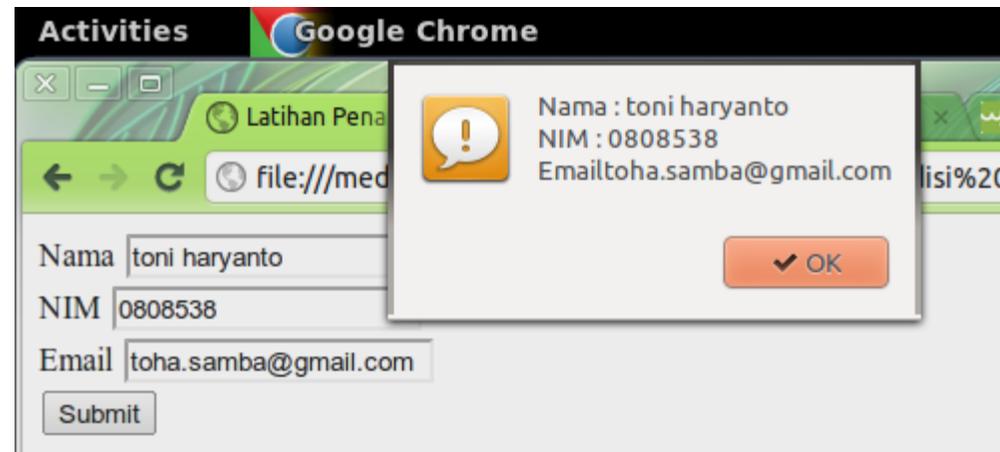
# Latihan Lagi

Sebelum Kamu mengakhiri pembahasan ini, ada baiknya mumpung masih panas, boleh dicoba ini latihannya.

Soalnya yaitu, kamu coba tambahkan satu kolom lagi yaitu email, dan definisikan error dari kolom email tersebut. Hasilnya kurang lebih seperti ini:



dan apabila tidak ada error lagi:



***Happy Coding!! :D***

# [Python]



Ridwan Fadjar Septian  
@ridwanbejoBlog  
ridwanbejo@gmail.com  
<http://ridwanbejo.wordpress.com>



# Pake Fungsi dan Prosedur Biar Kodenya Gak Redundant ^\_^

Sebelumnya di Python Nyankodz ini, kita sudah belajar Hello World. Terus belajar pemilihan kondisi, terus pengulangan juga. Nah sekarang kita bakal bahas nih bagian pemrograman yang gak kalah serunya yaitu **FUNGSI DAN PROSEDUR**. Terkadang kita selalu menyalin bagian sebuah kode untuk dipakai lagi di dalam sebuah program. Fungsi adalah sebuah potongan kode yang diberikan label dan parameter dan mempunyai fungsional tertentu. Tapi masih abstrak kan maksudnya apa ? Lebih jelasnya mah pantengin terus yah artikelnya.

## Let's PyNyankod.

Isi Artikel :

- a) Prosedur tanpa parameter
- b) Prosedur dengan parameter
- c) Fungsi tanpa parameter
- d) Fungsi dengan parameter
- e) Ruang lingkup variabel dalam pemrograman
- f) Referensi
- g) Latihan

Prosedur itu apa yah ? Kalau fungsi itu apa ? Kapan harus pakai salah satunya. Pertanyaan – pertanyaan ini muncul sewaktu penulis masih kuliah algoritma dan

pemrograman. Dulu hampir sering ngoding banyak sekali baris. Perintah yang sama saya ulang – ulang untuk membuat sebuah proses dalam program. Misal waktu itu saya ditugasi membuat sebuah program yang menampilkan data mahasiswa. Ya sudah , saya ketik ulang bagian kode yang dibuat untuk mencetak data mahasiswa sebanyak 10 kali dan tiap bagian datanya menampilkan data yang berbeda. Misal saya buat contoh kode program tadi seperti ini.

```
id = 1
nama = 'ridwan'
kelas = ' A'
prodi = 'ilmu komputer'
'
print "Nama : ", nama
print "Kelas :", kelas
print "Prodi : ", prodi

.....
# terus aja nyalin sampai 100 kali
.....

print ""

id = 100
nama = 'samsudin'
kelas = ' C'
prodi = 'ilmu komputer'
print "Nama : ", nama
print "Kelas :", kelas
print "Prodi : ", prodi
```

Dan alhamdulillah jadi deh programnya. Tapi capek kan ? Kalau 1000 data seperti diatas disalin terus bagaimana jadinya ?. Selain filenya bakal berukuran besar, pasti capek nulisnya juga. Gimana caranya yah biar kita memanggil proses yang sama

tapi data – datanya berbeda. Jadi walaupun sampai banyak sekali data yang akan ditampilkan, setidaknya kita tidak mengulang bagian kode yang sama. Kalau untuk peristiwa diatas kode yang sering diulang adalah bagian kode untuk mencetak nama, kelas dan prodi.

*Solusinya adalah dengan membuat **prosedur dan fungsi**.*

Sepanjang pengalaman penulis belajar tentang pemrograman sih, prosedur dan fungsi adalah bagian dari program yang dinamakan subroutine. Keduanya adalah potongan kode yang bisa dipakai ulang di baris kode utama atau baris kode dari program kita dimana prosedur atau fungsi tersebut dipanggil. Kita bisa memanggil potongan kode khusus tersebut melalui nama yang kita definisikan untuk potongan kode tersebut. Sebuah subroutine bisa mempunyai parameter atau tidak. Ibaratnya prosedur atau fungsi tersebut membutuhkan bahan – bahan dari luar prosedur tersebut untuk menghasilkan output sesuai dengan proses yang ada didalamnya. Kalau prosedur itu hanya memproses. Sedangkan fungsi itu memproses dan mengeluarkan nilai yang akan ditamping dalam variabel. Nah bagaimana sih membuat prosedur dan fungsi. Penjelasan lengkapnya ada dibawah sini.

## Prosedur Tanpa Parameter

---

Prosedur tanpa parameter yah ? Apa bedanya dengan prosedur yang berparameter ? Kalem – kalem kita bahas dulu deh prosedur tanpa parameternya dulu. Kita lihat kode dulunya aja yah terus kita analisis kodenya.

Sebelumnya siapkan dulu text editor yang biasa Nyankoder pakai, misal kalau penulis sih pakainya Geany atau Notepad ++. Nah Nyankoder buat sebuah file

bernama **prosedur\_tanpa\_parameter.py** dan salin kode berikut ke dalam file tersebut.

*Note : jangan lupa yah indentasi.*

```
# Membuat sebuah prosedur
def cetak_mahasiswa_berprestasi():
    # isi prosedur
    nim = '0501123'
    nama = 'Yanti Nur Kusuma'
    kelas = 'B'
    prodi = 'Teknik Informatika'
    print "Prosedur cetak mahasiswa dipanggil..."
    print ""
    print "NIM : ", nim
    print "Nama : ", nama
    print "Kelas : ", kelas
    print "Prodi : ", prodi

# cara memanggil prosedur
cetak_mahasiswa_berprestasi()

# memanggil prosedur untuk kedua kalinya
cetak_mahasiswa_berprestasi()
```

Setelah kalian membuat file bernama **prosedur\_tanpa\_parameter.py** eksekusi file tersebut melalui command line di windows atau terminal di linux. Dengan cara berikut :

*note : dibawah ini adalah cara eksekusi file python melalui terminal linux atau command line windows*

**Masuk ke direktori tempat file prosedur\_tanpa\_parameter.py disimpan.**

**Kemudian eksekusi file tersebut dengan perintah :**

`python nama_file.py.`

**Kalau pada kasus ini kita panggil perintah :**

`python prosedur_tanpa_parameter.py`

Kalau berhasil akan tercetak dua kali informasi yang sama. Yaitu informasi tentang mahasiswa teladan yang ada di dalam prosedur diatas. Enak kan ? Kita tidak perlu menyalin isi kode dari prosedur seperti contoh kode di awal pembahasan. Begitulah kira – kira contoh pemakaian prosedur dipython. Tapi kan statik ? Tidak ada parameter. Selanjutnya kita akan buat prosedur yang berparameter.

## Prosedur Dengan Parameter

---

Apa bedanya yah ? Waktu Nyankoder membuat prosedur formatnya seperti ini kan :

```
def nama_prosedur () :  
    isi prosedurnya disini
```

Nah disana ada tanda kurung buka dan kurung tutup kan ? Nah kedua tanda itu berfungsi untun menyimpan atau menentukan parameter apa saja yang akan dipakai untuk prosedur tersebut.Dengan demikian format penulisan prosedur yang berparameter akan menjadi seperti ini.

```
def nama_prosedur (arg ke – 1, arg ke -2, ,arg ke - n) :  
    isi prosedurnya disini
```

Sebagai contoh kita coba lagi yah buat file baru bernama **prosedur\_berparameter.py**. Kemudian salin kode berikut kedalam file **prosedur\_berparameter.py** Nyankoder.

```
# membuat sebuah prosedur dengan parameter
def cetak_mahasiswa(nim, nama, kelas, prodi):
    print "Prosedur cetak mahasiswa dipanggil..."
    print ""
    print "NIM : ", nim
    print "Nama : ", nama
    print "Kelas : ", kelas
    print "Prodi : ", prodi

# cara panggilnya adalah sama seperti prosedur tanpa parameter
# hanya saja kita melewatkan (istilah di pemrograman untuk
# memberikan
# sebuah parameter ke dalam prosedur) nilai kedalam prosedur
# tersebut

cetak_mahasiswa ('0901939', 'Ridwan F.S.', 'C', 'Ilmu Komputer')
cetak_mahasiswa ('0801023', 'Fadjar R.S.', 'A', 'Pendidikan Ilmu
Komputer')
```

Kemudian eksekusi file tersebut dengan cara seperti pada pembahasan *prosedur tanpa parameter*. Kalau berhasil pada layar akan tampil dua informasi yang melakukan proses sama tapi datanya berbeda. Karena datanya berasal dari parameter yang dipakai oleh prosedur yang kita panggil.

## Fungsi Tanpa Parameter

---

Lalu bagaimana dengan fungsi tanpa parameter apakah sama dengan kedua contoh diatas ? Sama sih, hanya saja bagaimana cara memanggilnya dan mendefinisikannya saja. Hanya saja ada sedikit hal yang ditambahkan untuk membuat sebuah fungsi. Formatnya kurang lebih seperti ini.

```
def nama_fungsi () :  
    isi fungsinya disini  
    return nilai yang dihasilkan oleh fungsi ini
```

Sebagai percobaan kita buat lagi file baru bernama **fungsi\_tanpa\_parameter.py** dan salin kode berikut ke dalam file **fungsi\_tanpa\_parameter.py**.

```
# membuat sebuah fungsi tanpa parameter  
def contoh_konstanta():  
    number = 100  
    konstanta = 10  
    # melakukan proses di dalam fungsi  
    hasil = number * konstanta  
    # mengeluarkan nilai dari variabel yang akan kita pakai  
    # sebagai hasil proses dari fungsi yang kita buat  
    return hasil  
    # cara panggil fungsi di python adalah dengan memasukkanya ke  
    dalam  
# sebuah variabel  
konstanta_saya = contoh_konstanta()  
# Kemudian kita cetak isi dari variabel yang kita gunakan untuk  
# menyimpan hasil keluaran fungsi yang kita panggil  
print "Konstanta yang telah saya buat : ", konstanta_saya
```

Kemudian eksekusi dengan cara seperti yang sudah dilakukan di pembahasan *prosedur\_tanpa\_parameter*.

## Fungsi Dengan Parameter

---

Kalau fungsi yang ini hanya tinggal menambahkan parameter seperti prosedur berparameter. Format umumnya seperti berikut :

```
def nama_fungsi (arg ke - 1, arg ke -2, ,arg ke - n) :  
    isi fungsinya disini  
    return nilai yang dihasilkan oleh fungsi ini
```

Sudah lebih paham kan ? Sekarang agar lebih paham lagi kita tinggal coba nih contoh fungsi berparameter. Nyankoder buat file baru lagi yah dengan nama **fungsi\_berparameter.py**. Salin kode berikut :

```
# membuat sebuah fungsi berparameter  
def hitung_luas_persegipanjang(panjang, lebar):  
    luas = panjang * lebar  
    return luas  
  
# variabel yang akan dipakai untuk parameter fungsi yang akan  
kita pakai  
panjang = 50  
lebar = 60  
  
# memanggil fungsi berparameter yang telah kita buat  
luas = hitung_luas_persegipanjang(panjang, lebar)  
  
# menampilkan hasil nya  
print "Panjang persegi panjang : ", panjang  
print "Lebar persegi panjang : ", lebar  
print "Luas persegi panjang : ", luas  
  
# kemudian kita panggil lagi dengan isi parameter yang berbeda
```

```
panjang = 30
lebar = 48
luas = hitung_luas_persegipanjang(panjang, lebar)

print "Panjang persegi panjang : ", panjang
print "Lebar persegi panjang : ", lebar
print "Luas persegi panjang : ", luas
```

Kalau udah buat filenya eksekusi lagi dengan cara seperti pada pembahasan *prosedur tanpa parameter*. Rame kan ? Pada akhirnya kita tidak perlu menyalin bagian kode proses yang sama untuk memproses data. Jangan sampai kita mengulang bagian kode seperti peristiwa di awal pembahasan. Sebenarnya sih kalau Nyankoder perhatikan. Di contoh bagian *fungsi dengan parameter* penulis masih melakukan pengulangan baris kode yang mempunyai fungsi sama. Jadi yah memang kita perlu pikirkan dan mendesain dengan baik bagaimana program kita akan dibuat.

## Ruang Lingkup Variabel Dalam Pemrograman

---

Berikutnya bahasan terakhir kita adalah ruang lingkup variabel atau *variable scope*. Jadi variabel itu mempunyai daerah kerjanya sendiri. Jika berada diluar *subroutine (prosedur dan fungsi)* maka dinamakan variabel global. Sedangkan jika berada di dalam *subroutine (prosedur dan fungsi)* dinamakan variabel lokal. Sebuah *subroutine* akan memakai variabel yang ada di dalam *subroutine* tersebut. Jika ingin memakai variabel dari luar harus dilewatkan sebagai parameter. Lebih jelasnya buat sebuah file baru bernama **variabel\_lokal\_global.py**. Kemudian salin kode berikut:

```
# variabel x yang dibawah ini adalah variabel global
# karena tidak disimpan di dalam subroutine
x = 10
```

```

def cetak_x():
    # kalau variabel x yang ini adalah variabel lokal yang ada di
    # dalam
    # subroutine
    x = 10
    print "x yang di dalam prosedur cetak_x() bernilai : ", x

# melihat nilai x sebelum cetak_x() dipanggil
print "x yang di luar prosedur cetak_x() bernilai : ", x
cetak_x()
# melihat nilai x sesudah cetak_x() dipanggil
print "x yang di luar prosedur cetak_x() bernilai : ", x
# setelah baris kode diatas dipanggil nilai x masih tetap 10.Lalu
?

# kita ubah lagi sedikit kode yang ada di dalam prosedur
cetak_x()
def cetak_x():
    # untuk menggunakan variabel x yang bersifat global
    # kita panggil kata kunci global dan nama variabel global
    # yang akan kita pakai
    global x
    x = 99
    print "x yang di dalam prosedur cetak_x() bernilai : ", x

# melihat nilai x sebelum cetak_x() dipanggil
print "x yang di luar prosedur cetak_x() bernilai : ", x
cetak_x()
# melihat nilai x sesudah cetak_x() dipanggil
print "x yang di luar prosedur cetak_x() bernilai : ", x
# setelah baris kode diatas dipanggil nilai x berubah menjadi 99

```

Eksekusi lagi yah file nya seperti cara di bahasan prosedur *tanpa parameter*.  
 Jelaskan ? Bagaimana sebuah variabel saja memiliki daerah kerjanya sendiri. Untuk itu

Nyankoder perlu hati – hati dalam mendeklarasikan dan menggunakan sebuah variabel. Yeah...Coding is Pulen...

**Referensi :**

- Python Tutorial, <http://www.java2s.com>
- Python Documentation, <http://www.python.org>
- Python Cookbook, O'Reilly Publisher
- Python Programming Fundamental, Springer Publisher

## Latihan

---

Untuk lebih memantapkan lagi latihannya. Buatlah sebuah prosedur atau fungsi untuk menghitung luas dan keliling persegi panjang, persegi, segitiga, trapesium, dan jajar genjang. Nanti file python nya kirim ke [ridwanbejo@gmail.com](mailto:ridwanbejo@gmail.com) atau [ridwanbejo@nyankod.com](mailto:ridwanbejo@nyankod.com) yah dengan judul :

Nyankod\_Python\_<< Edisi Nyankod >>\_Instansi\_Asal\_Nama. Nanti saya berikan feedback tentang latihan yang Nyankoders buat. Ditunggu yah, jangan lupa kirim

Akhir kata terima kasih telah mengikuti artikel Python ini di Nyankod Magazine.

*Bahasa pemrograman itu beragam, tapi mempunyai satu dasar yaitu algoritma. Tapi tidak salah juga mempelajari lebih dari satu bahasa pemrograman karena setiap bahasa pemrograman mempunyai gaya dan tujuan yang berbeda.*

Sampai jumpa di edisi berikutnya Nyankoder.

# [ActionScript]



Tarom Apriyanto  
@tarompey  
tarompey@gmail.com



# Penggunaan NextFrame dan PrevFrame

Asalamualaikum Wr. Wb. Salam super buat kita semua (kayak pak Mario Teguh lagi), seperti biasa tak henti-hentinya untuk menanyakan kabar para Nyankoder yang ada di rumah, yg di kamar, di kantor di ruang makan, di toilet atau di manapun anda berada yang pastinya sedang menghadap layar kompi atau Lapi masing-masing maupun minjem temennya??. So pasti baik semua lah dan Nyankodist selalu doain yang terbaik buat Nyankoder semuanya selama masih menikmati suguhan ActionScript ini (doa yang bersyarat hahahaha).

Gimana, sudahkah sobat Nyakod mencoba materi kemarin??. gampang kan dan juga bahan latihan pun gampang banget kan buatnya?? Nah kalau ada yang masih merasa susah, jangan segan untuk menanyakan langsung di email saya.

Nah Sob, materi yang bakal kita bahas sekarang ini adalah materi lanjutan dari materi edisi 7 kemarin, masih berhubungan dengan tombol tapi dengan ActionScript yang berbeda. Materi kali ini bisa dikatakan materi yang paling sederhana karena hanya menggunakan beberapa frame untuk membuat animasi slide foto mirip dengan kemarin yang kita buat, namun nanti kita akan membuat aplikasi slide photo yang agak dinamis (dinamis di sini slide photo bisa kita tampilkan dengan slide show dan tampil satu persatu).

Nah sob seperti biasa, sebelum kita mulai menelusuri materi di edisi kali ini, alangkah baiknya sobat Nyankod atau Nyankoder siapkan posisi yang enak biar materi yang disampaikan dapat dicerna dengan mudah. Supaya menambah kenikmatan membaca, saya sarankan siapkan secangkir teh hangat dan beberapa cemilan,

cepuluh, dan sebelas (berhitung ni) di dekat posisi anda membaca Hehehee. Oke Check this out.

## NextFrame dan PrevFrame

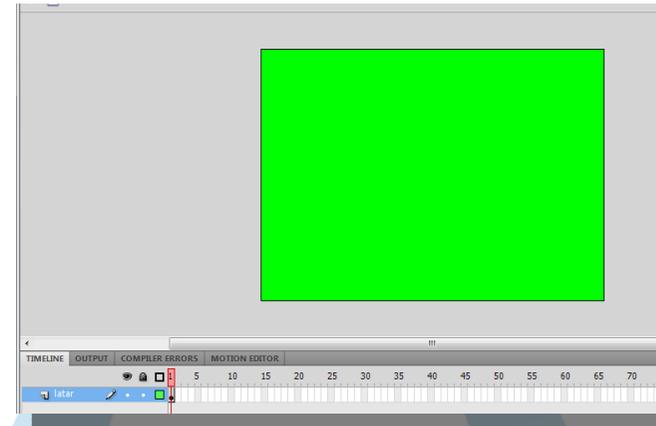
---

Perintah *NextFrame* berfungsi untuk menuju ke frame berikutnya, kalau *PrevFrame* tentunya berfungsi sebaliknya atau digunakan untuk menuju ke frame sebelumnya. Cukup jelas kan??? Jadi bisa disimpulkan bahwa perintah *NextFrame* dan *PrevFrame* adalah sebuah perintah yang digunakan untuk melakukan sebuah aksi menuju frame selanjutnya (*NextFrame*) dan menuju frame sebelumnya (*PrevFrame*).

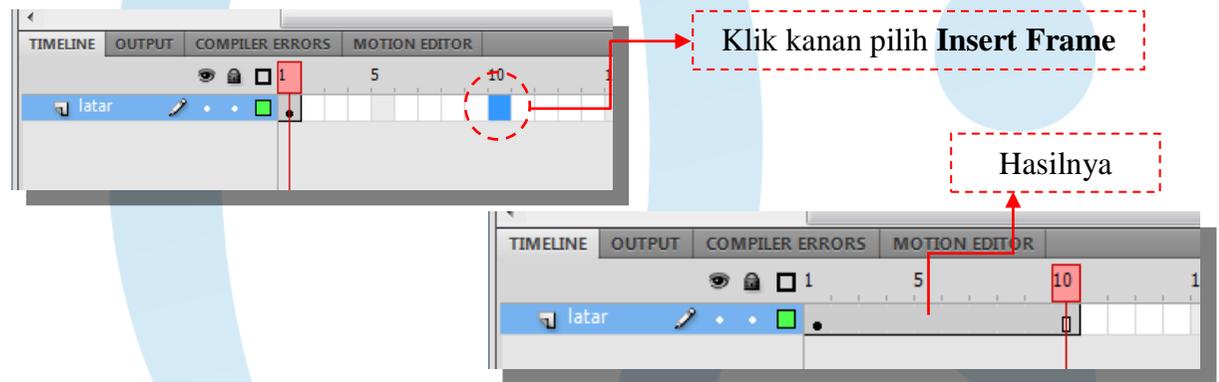
Nah agar lebih gampang ngingetnya gini aja, liat kata pada perintah tersebut yaitu *NextFrame*, kata *next* adalah bahasa inggris dari "selanjutnya" dan frame berarti ya frame di flash, jadi kalau mau inget script ini mesti bisa bahasa inggris dulu pasti Nyankoders jago dong bahasa inggris hehehehehe. Terus kalau *PrevFrame*, *Prev* itu singkatan dari *Previous* yang berarti "sebelumnya" frame sama kayak yang tadi udah dijelasin, jadi pokoknya kurang lebih kayak gitu lah biar gampang inget.

Nah biar tambah mudah ingetnya adalah dengan memberikan contoh dan mempraktikannya. Jadi langsung saja kita praktikkan oke?? Kita bakal buat sebuah slideshow photo sederhana dengan menerapkan perintah *NextFrame* dan *PrevFrame*, Check this out:

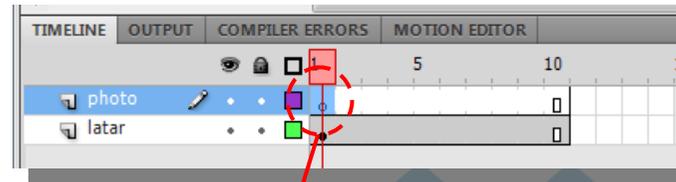
1. Pastinya sobat Nyankod buka aplikasi Adobe Flashnya dengan tipe ActionScript 2.0. ga perlu lama-lama...hahahaha
2. Klik 2x pada layer 1 yang muncul secara default di Timeline, ganti namanya dengan nama "latar".
3. Klik tombol **Rectangle tool**  , buat objek kotak di lembar kerja secara penuh sebagai background, **like this yoo:**



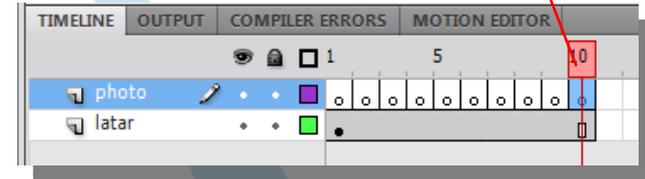
4. **Next**, klik kanan pada frame 10 kemudian pilih **Insert Frame** untuk manjangan tampilan framanya. **Like this yoo :**



5. Buat layer baru dengan klik tombol **New layer**, kemudian klik 2x ganti namanya menjadi "photo".
6. Klik keyframe pertama di layer "photo", kemudian tekan tombol **F6** sebanyak 9 kali untuk nambahin keyframe sehingga frame 1 sampai dengan frame 10 terisi keyframe kosong. **Like this yoo:**

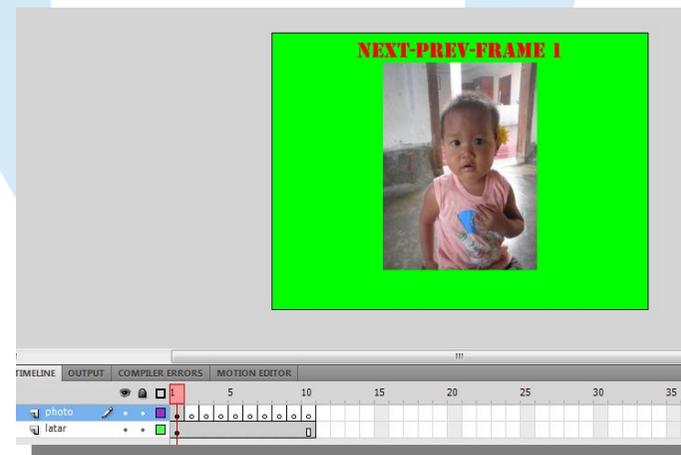


Klik Keyframe 1  
lalu tekan F6  
sebanyak 10 kali



Hasilnya

7. Klik di frame 1 layer "photo", terus impor photo yang akan di tampilkan ke stage (lembar kerja) dengan menekan tombol **Ctrl + R** di keyboard kompi/lapi sobat Nyankod sendiri (jangan punya orang lain yang dipecet hahaha).
8. Atur posisi dan ukuran objek photo yang di tampilkan dengan menggunakan **Free Tranform tool** . **Like this yoo :**



9. Tambahkan tulisan di atas objek photo terserah apa saja, seperti pada contoh di gambar langkah no 8. "NEXT-PREV-FRAME".
10. Dengan cara yang sama, masukan gambar yang berbeda pada setiap frame yaitu frame 2-10 pada layer "photo" (ulangi langkah no 7 & 8).
11. Selanjutnya tambahkan layer baru, klik 2x pada layer tersebut dang anti namanya dengan "tombol".
12. Klik menu **Window > Common Libraries > Buttons** untuk menampilkan panel **Library-Buttons**.
13. Pilih beberapa tombol yang diperlukan untuk slideshow photo ini, dengan cara drag contoh tombol yang ada pada panel **Library-Buttons**. **Like this yoo :**



14. Klik tombol sebelah kiri atas, kemudian tekan **F9** untuk membuka panel **Action** dan ketik ActionScript SBB :



```
on(release) {  
    prevFrame();  
}
```

15. Klik tombol sebelah kanan atas, kemudian tekan **F9** untuk membuka panel **Action** dan ketik ActionScript SBB :



```
on(release) {  
    nextFrame ();  
}
```

16. Klik tombol tengah bawah, kemudian tekan **F9** untuk membuka panel **Action** dan ketik ActionScript SBB :



```
On(release) {  
    play();  
}
```

17. Klik tombol kiri bawah, kemudian tekan **F9** untuk membuka panel **Action** dan ketik ActionScript SBB :



```
on(release) {  
    stop();  
}
```

18. Klik tombol kanan bawah, kemudian tekan **F9** untuk membuka panel **Action** dan ketik ActionScript SBB :

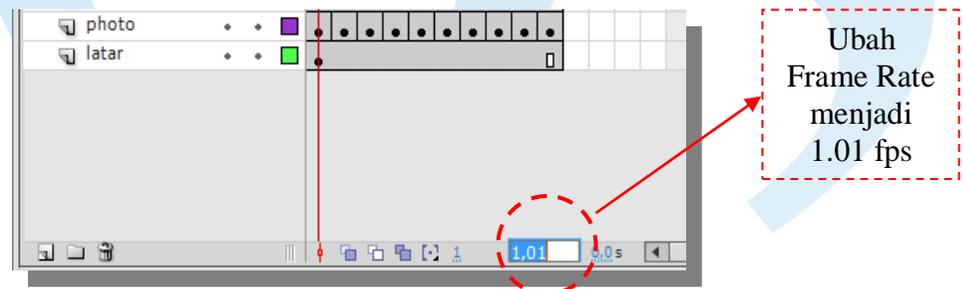


```
on(release) {  
    gotoAndStop(1);  
}
```

19. Klik tombol **New Layer** untuk membuat layer baru, kemudian klik 2x dan anti namanya menjadi "action".
20. Klik frame 1 pada layer "action", kemudian tekan **F9** dan tuliskan perintah SBB :

```
Stop();
```

21. Kemudian ubah kecepatan tampilan slide pada icon **Frame Rate** menjadi 1.01 fps. Agar tampilan animasi slide dijalankan 1 frame per detik. **Like this yoo :**



22. Selesai, Simpan file dan jalankan (tekan **Ctrl + Enter**).

## PR

---

Oke setelah kita mencoba coba praktik seperti di atas, coba sekarang lanjutkan praktik tersebut dengan menambahkan 3 tombol, masing-masing digunakan untuk :

1. **Tombol 1** digunakan untuk berhenti pada slide pertama.
2. **Tombol 2** digunakan untuk berhenti pada slide terakhir.
3. **Tombol 3** digunakan sebagai link menuju suatu situs internet (blog Nyankoder juga boleeeh).

Gampangkan, PR ini berguna untuk mengingat materi yang telah di sampaikan di edisi-edisi sebelumnya. Cayooooo.....

## Penutupan

---

### Referensi book:

Kupas Tuntas Adobe Flash CS5  
Madcoms & Penerbit Andi

Nah segitu aja materi dan praktiknya untuk edisi 8 kali ini semoga bisa bermanfaat bagi Sobat Nyankod semuanya terutama yang baru belajar dan mau belajar, sebelumnya mohon maaf apabila ada kekurangan dalam penulisan atau kata-kata yang kurang sopan dalam edisi ini. Kalau mau bertanya sekali lagi jangan segan-segan untuk meng e-mail (kirim via e-mail maksudnya) ke saya langsung.

Jangan lupa untuk mencoba, mencoba jangan takut salah, kalau salah cepet cari cara supaya benar karena kalau salah terus-menerus nanti bisa masuk penjara, tapi bukan hanya orang yang salah aja yang di penjara, tapi orang bener juga dipenjarakan. Intinya di mana kita ada kemauan untuk mencoba sesuatu yang bermanfaat di situ ada jalan untuk menggapainya, **Like this yoo** "there is a will there is a way" jangan pipis di depan busway (nanti bisa ketabrak). Oke sampai jumpa di edisi-9 sobat Nyankod.

#nyanKomik



## Kamu Dikatakan Maniak Apabila:

Karena kedekatannya dengan komputer, kadang-kadang seorang programmer itu bisa jadi seorang yang sangat maniak. Begini nih kira-kira kalo udah maniak:

### Tidur Bareng Sama Laptop

Kadang saking karena nggak pengen pisah jauh dari laptopnya, beberapa orang yang maniak kadang-kadang tidur bareng sama laptopnya.



## Suka Kayak Orang Autis Kalo Diajak Ngobrol

Saking asiknya kalo lagi bareng komputernya, beberapa orang yang kelewatan kadang jadi kayak orang autis kalo lagi sama komputernya. Susah diajak ngobrol, bahkan kadang ceweknya sendiri aja dicuekin.



## Nggak Tau Waktu

Kalo lagi sama komputer, kadang-kadang sampe nggak inget waktu. Pagi, siang, sore atau malem kayak nggak bisa dibedain. Kadang bahkan lupa sama lingkungannya, sama kesehatannya. Mungkin sampe nggak sadar kalau ade-adenya udah pada gede-gede, udah nikah dan udah punya anak.



## Serasa Hampa Setelah Mematikan Komputer

Saat-saat bersama komputer adalah saat-saat yang sangat menyenangkan baginya. Dan ada kalanya ia harus mematikan komputernya, mungkin karena mau berangkat sekolah atau ke kampus. Itulah saat-saat yang paling mengharukan, serasa akan pergi jauh, mendadak menjadi galau dengan perasaan hampa.



## Ngomong Dengan Komputer

Mungkin karena kesetrum atau karena kena hipnotis dari program komputer, beberapa orang yang sangat maniak bahkan ngobrol dengan komputer. Bahkan berpamitan saat ke kamar mandi.



Bagi Kamu yang merasa Nyankod Magz bermanfaat, dan Kamu ingin memberikan apresiasi, maka silahkan kirimkan Testimoni Kamu ke [mail@nyankod.com](mailto:mail@nyankod.com). Oh ya, jangan lupa dishare ya ke temen-temen yang laennya, supaya semakin banyak yang dapet manfaat dari Nyankod.

**Edisi ke-9 akan terbit: Rabu, 4 Juli 2012**



**Coding is pulen bro!!!**