Nyankod Magz

Love it and Earn It

Edisi 7

Nyankod Magz Edisi 7
“Dodol Query”

Love it and Earn It

Sapa Nyankod

Hi Nyankoders semua, yang ada di seluruh penjuru Nusantara. Kembali rapatkan barisan, bersiaplah untuk menerima guncangan kami kali ini. Guncangan yang sangat luar biasa, yang bisa membuat perut mendadak menjadi lapar dan mata berkunang-kunang, yaitu dodol ajaib.

Tapi sebelum kami jelaskan lebih banyak tentang dodol yang satu ini, seperti biasa, siapkan secangkir kopi hangat, beberapa gorengan dan list lagu dangdut yang siap untuk menemani, biar nggak tegang cong...

Oh ya Jou, pernah nggak kebayang dalam benak kamu gimana keadaanya kalau negara kita juga punya pahlawan super, ya sekelas Superman gitu lah, pasti Indonesia akan lebih aman. Ya, seenggaknya kejadian jatuhnya pesawat Sukhoi Superjet 100 kemarin bisa teratasi dengan mudah. Ketika pesawat mulai oleng, tiba-tiba pahlawan super datang dan menghentikan laju pesawat, atau malah mengantarkannya ke tempat tujuan, so nggak jadi deh jatohnya. Ya kalau pun nggak superman, minimalnya kayak Great Saiyaman-lah. Lumayanlah.... Atau ketika ada demo mahasiswa dan berakhir dengan kerusuhan, kan pahlawan super bisa dateng buat melerai. Tuh kan bermanfaat. Atau ketika lagi ada pertandingan sepak bola, terus karena sifat panatik dari suporter yang nggak bisa nerima kekalahan jagoannya, akhirnya pada ribut deh, malah sampe bakar stadiun segala. Kan kalo ada super hero semua itu bisa mudah diatasi. Ah pokoknya mah banyak untungnya deh kalo ada super hero mah...

Lho kok jadi ngomongin super hero??? Nggak nyambung tau!! Haha biarin aja. Eh tapi kalo dipikir-pikir dengan cermat dan pandangan yang maksain, ternyata seorang programer juga mirip kayak super hero lho. Nggak percaya?? Coba aja deh renungin, saya yakin pasti ada kesamaannya. Kalo udah ketemu jawabnnya langsung ngetwit dan mention ke @nyankodTWEET ya?? 10 orang yang beruntung akan mendapatkan hadiah yang menarik lho...

Tapi ngomong-ngomong super hero ntar nyankod bikin super hero ah. Siapa tau bisa berguna bagi nusa dan bangsa.

Oh ya, untuk edisi Cingcau Looping sebelumnya, kami mohon maaf, ada sedikit kesalahan penulisan. Hehehe... maaf ya, editor kita lagi galau jadi salah ketik deh. Hehehe... Yang bener python, bukan phyton. Maaf ya nyankoders sekalian.

Oke deh, dari pada berpanjang-panjang lebar ntar malah jadi bingung, langsung aja nih cicipin dodol jenis baru, yaitu "Dodol Query". Rasanya nggak kalah lho dari Dodol Garut. Nggak percaya?? Cobain aja nih...

Love it and earn it

Nyankodist Team

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.

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..

Seru-seruan nyankodTWEET

Kadang-kadang di twitter ada nyankoders yang mention lucu-lucu atau ngebales twit dari Nyankod dengan jawaban yang lucu-lucu. Nih beberapa diantaranya.

Hadeuuuuh, ini ada-ada aja nih abang @Wildan_Shidiq. Kok laper mesti banget laporan ke Nyankod. Emang dikirain Nyankod buka warteg. Cape deh... Kalo laper, cobain deh ke kamar mandi, nah di situ ada sabun mandi, coba gigit. Lumayan buat sarapan.

Yang ini lebih parah lagi nih, @hirokakaoshi. Seolah-olah Nyankod buka jasa konsultasi skripsi apa?? Hmmm.... Kalo pengen beres skripsinya ya, dikerjain bang, coba ambil sapu, ambil minyak tanah, terus ambil korek api, abis itu bakar deh. Pasti deh beres. Ya beres semuanya...

Curang nih si abang @plathitambiasa. Dia udah punya bini, ya enak dong ngejawabnya milih laptop, tanpa ada rasa galau dan dilema sedikitpun. #guenggakterima

Si abang @dhebonno ketauan tukang makan. Dia stress kalo nggak ada makanan. Awas bang ntar gendut lho....

Mau ikutan seru-seruan bareng @nyankodTWEET?? Buruan follow akun @nyankodTWEET....!!!!

Nyankoders Menyapa...

Dari **David Sanjaya** (david.teknik@yahoo.com)

Dear Nyankod Team..

Pertama tau nyankod yah dari grub blogger palembang di Facebook.. Dan kebetulan juga ane kuliah di jurusan Teknik Informatika yang sangat akrab sekali dengan namanya "Coding"/Pemrograman. Awal buka websitenya sih biasa aja, tapi setelah baca "Cingcau loopingnya" Jadi kepengen download yg lainnya.. hehe :)

Oiya nanti kalo bisa dimasukin juga yah untuk pemrograman C++ dan JAVA nya... Soalnya ane lgi belajar yg itu tuh.. :)

Oke skian testi dari saya, smoga dibaca dan ditanggapi..

TTD : pureheart99

Blog : http://bidargoblog.wordpress.com

Salam kenal blogger palembang....
Ouuh.. udah baca Cingcau Looping ya, gimana rasanya??
syukurlah kalo bermanfaat.

untuk C++ dan Java, kami dalam pencarian kontributor penulisnya. Untuk saat ini belum ada gan, insya Allah kedepannya akan kami usahakan terus buat menambah bahasa pemrograman lainnya, biar makin seru...

Dari **Moch. Faisal Arief** (faisalarief@gmail.com)

Hi admin nyankod

mohon diberitahu cara mendownload ebook nyankodMagz edisi sebelumnya, saya klik menu download tidak bisa, demikian

---

Untuk download edisi sebelumnya, silahkan abang klik tulisan:
Download edisi sebelumnya...
Yang ada di bawah jadwal penerbitan edisi selanjutnya dan di bawah cover nyankodMagz edisi terbaru. Nanti ikuti seperti biasa aja ya bang. Hehehe..

Kalo ada pertanyaan, seputar nyankod silahkan jangan ragu-ragu untuk mengirimkan pertanyaan kamu melalui email ke 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

Tipe Data Tingkat Lanjut

Setelah sekian rehat sejenak mengerjakan kewajiban, akhirnya sms dari "forever alone" pun tiba, inga – inga deadline hari sabtu ?*%^_" harus kumpul semua artikelnya !! wajib dengan font bold sebagai penekanan, aye aye laksanakan 86 komando diterima.

Setelah membaca artikel bahasa C dari edisi 1 sampai 6 kita tahu bahwa Bahasa C mempunyai lima macam tipe data dasar, yaitu int, double, float, char, dan void, selain tipe data tersebut Bahasa C juga menyediakan tipe data yang lain, yaitu tipe data struktur, enumerasi, bit field dan union. Oke lets begin "_"

Struktur

Struktur atau structure selanjutnya disebut struktur, jika kita mengartikan dengan bahasa sederhana (menurut saya) struktur adalah suatu bentuk yang saling berhubungan dalam satu kesatuan, yups itu menurut saya tetapi didalam Bahasa C struktur adalah kumpulan data yang berbeda tipe yang bernaung dalam nama yang sama, jika sebelumnya jika mempelajari mengenai array atau larik adalah kumpulan data yang mempunyai tipe yang sama, kali ini kumpulan data mempunyai tipe yang berbeda.

Jika didalam larik sebelumnya kita menyebut tipe data homogen tetapi kali ini tipe data bersifat heterogen (horee gak maho lagi "_"). Elemen – elemen pembentuk struktur biasa disebut dengan field. Suatu struktur terdiri dari elemen – elemen yang

saling berhubungan dan bukan asal dikumpulkan, contoh sebuah struktur mahasiswa terdiri dari field, nik, nama, tgl_lahir, jenis kelamin.

Penulisan struktur didalam Bahasa C terdiri atas tipe data struktur dan variable struktur. Deklarasi struktur menggunakan kata kunci struct dan elemen – elemen pembentuk struktur dituliskan didalam {}. Untuk mengakses field sebuah struktur digunakan operator titik yang dituliskan diantara nama struktur dan field.

Penulisan struktur seperti dibawah ini

Deklarasi struktur

```
1  Struct[tag]{
2    Type mvar_name;
3    Type mvar_name;
4  }svar_name;
```

struct : wajib dituliskan jika ingin mendeklarasikan struktur
type : jenis data
mvar_name : nama field
svar_name: nama variable dari struktur

Untuk pengaksesan elemen pada struktur dapat dilakukan dengan menyebutkan nama variabel struktur diikuti dengan operator titik dan nama dari elemen strukturnya

```
1  variabel_struktur.nama_field
```

seperti pada contoh dibawah ini

Program 7.1 :

```
1   #include <stdio.h>
2   main()
3
4   {
5   struct{ /*struktur bersifat lokal */
6   float Panjang; /*field struktur*/
7   float Lebar; /*field struktur*/
8   float Luas;/*field struktur*/
9   }Persegi_panjang;
10
11  printf("Panjang");
12  scanf("%f",&Persegi_panjang.Panjang);
13  printf("Lebar");
14  scanf("%f",&Persegi_panjang.Lebar);
15
16  Persegi_panjang.Luas       =       Persegi_panjang.Panjang       *
17  Persegi_panjang.Lebar;
18  printf("Luas Persegi Panjang = %f\n",Persegi_panjang.Luas);
19  }
```

Output :

```
Panjang = 5
Lebar = 6
Luas Persegi Panjang = 30.000
```

Pada program diatas struktur ditulis tanpa menggunakan tag, elemen struktur terdiri atas 3 elemen yaitu panjang, lebar dan luas. Untuk mengakses elemen dari

struktur digunakan operator titik sehingga untuk mengakses elemen struktur diatas ditulis.

```
1    Persegi_panjang.panjang
```

Jika anda belajar java, pengaksesan data seperti ini akan banyak dijumpai. ☺

Program 7.2 :

```
1   #include <stdio.h>
2   #define MAXTITL   40
3   #define MAXAUTL   40
4   #define MAXBKS   100
5   struct book {                   /*struktur bersifat global   */
6       char title[MAXTITL];
7       char author[MAXAUTL];
8       float value;
9   };
10  int main(void)
11  {
12       struct book libry[MAXBKS]; /* variabel struktur */
13       int count = 0;
14       int index;
15
16       printf("Please enter the book title.\n");
17       printf("Press [enter] at the start of a line to stop.\n");
18       while (count < MAXBKS && gets(libry[count].title) != NULL
19                         && libry[count].title[0] != '\0')
20       {
21            printf("Now enter the author.\n");
22            gets(libry[count].author);
23            printf("Now enter the value.\n");
24            scanf("%f", &libry[count++].value);
25            while (getchar() != '\n')
```

```
26              printf("Enter the next title.\n");
27          }
28          printf("Here is the list of your books:\n");
29          for (index = 0; index < count; index++)
30          printf("%s by %s: $%.2f\n", libry[index].title,
31              libry[index].author, libry[index].value);
32          return 0;
    }
```

Output :

```
Please enter the book title.
Press [enter] at the start of a line to stop.
My Life as a Budgie
Now enter the author.
Mack Zackles
Now enter the value.
12.95
Enter the next title.
    ...more entries...
```

Pada program diatas struktur bersifat global, sehingga data struktur dapat diakses dimanapun didalam program. Kemudian struktur kembali diakses melalui struct book libry[MAXBKS] untuk menempatkan struktur sebagai array. Untuk mengakses struktur tersebut dapat dituliskan libry[0].value .

Bit Field

Kadang dalam pengolahan data kita hanya membutuhkan data yang mempunyai dua nilai seperti ganteng atau jelek, maho atau banci, hahaha becanda y, gak serius, soalnya sambil mikirin final Champions, yups contohnya yang sesuai misal benar atau salah, laki – laki dan perempuan, lulus dan tidak lulus, biasanya untuk menampung data ini kita menggunakan tipe data karakter. Satu karakter mempunyai nilai satu byte atau delapan bit. Padahal untuk efisiensi memori kita dapat menggunakan cukup satu bit saja.

Sebuah struktur terdiri atas beberapa elemen atau field. Field suatu struktur boleh dinyatakan dengan ukuran bit. Field seperti inilah yang disebut dengan bit field

Program 7.3 :

```
1   #include <stdio.h>
2   typedef struct{
3   unsigned int f_1:10;
4   unsigned int f_2:12;
5   }tipe1;
6
7   int main(){
8   tipe1 record;
9   printf("%d byte", sizeof(record));
10  return 0;
11  }
```

Output

```
4 byte
```

Yups, akhirnya sekian dulu artikel mengenai Struktur, Struktur merupakan kumpulan tipe data yang berbeda dalam satu kesatuan objek, berbeda dengan array yang merupakan kumpulan tipe data yang sama. Dalam hal ini struktur banyak digunakan untuk mendefinisikan suatu objek yang memiliki kumpulan data yang berbeda. Jika kita menguasai penggunaan struktur dalam Bahasa C, mungkin jika anda sedang mempelajari Java akan lebih mudah mengerti karena Java merupakan pemrograman yang berorientasi objek. Terima kasih.

Referensi :

- C Primer Plus, Fourth Edition
  Stephen Prata.
- Algoritma dan Struktur Data Bahasa C
  Thompson Susabda Ngoen

[Ruby]

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

Good MORNING *CLASS* (Session 2)

Selamat #@waktu semuanya :) apakabar nih? Semoga baik dan sehat ya Gimana yang edisi kemarin sudah bisa dan dicoba program pemutar lagunya??Hehe... Mudah kan kemarin mengenai pengenalan Class dan Method? Kalo ada yang bingung, tanyain aj langsung ya kontak saya saja di twiter atau chat.

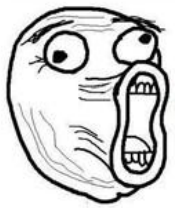
Oya, untuk edisi sekarang akan melanjutkan pengenalan Class dan Method, di dalam sini kita akan membahas mengenai Variabel lokal, dan Tipe-tipe kontrol akses Class (Protected, Private, dan Public). Tadinya mau dibahas mengenai inheritance namun sepertinya belum perlu untuk pengenalan karena selanjutnya kita akan membahas algoritma dasar dulu :)

well, kita review dulu apa yang sudah dipelajari minggu kemarin

a) *Sudah membahas mengenai pengertian class dan method*
b) *Ciri-ciri Class sudah dijabarkan*
c) *Teknik dasar menggunakan Class juga sudah cukup dikuasai*

Berhubung kemarin ada Latihan, dan lumayan sulit jadi di edisi sekarang sudah saya lampirkan kunci jawaban beserta penjelasannya. Jangan lupa download Source Code nya juga ya kawan.

Akhirnya....KITA MULAI....



Level - 7

Mari kita mulai..berdoa dulu ya semoga edisi ini bisa lebih memantapkan kamu mengenai OOP di Ruby | the real Son of OOP.

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.

Senjata tempur kita sudah siap, persiapan sudah matang pula...lets LOOP...

I. Variabel Lokal dalam Class

Variabel lokal merupakan variabel yang hanya dapat digunakan dalam area dimana variabel tersebut di deklarasikan. Misal ilustrasinya begini :

*Zahira punya **kucing** anggora, **Kucing** itu bisa akrobat dan lucu tingkahnya. Namun sayangnya dia hanya bisa bertingkah kalau **di dalam rumah** saja, kalau di ajak **keluar rumah** dia diam dan tak bertingkah.*
Sayang sekali yaa....
Namun Lucu banget kucingnya....

Cerita di atas membawa kucing dan rumah, kucing itu ibarat Variabel Lokal, dan Rumah adalah Class. Hayoo..ngerti kan maksudnya ??? (^__^)9 . Kalau belum paham, baca lagi deh ya ceritanya hheu.. Kalau sudah, perhatikan kode berikut ini :

Program 7.1: Contoh variabel lokal di Ruby (VarClass.rb)

```
1  #!/usr/bin/ruby
2
3  #belajar menggunakan variabel class:: variabel yg hanya berlaku di
4  #kelas dimana var trsbut dideklarasikan.
5  class Lagu
6    @@varClass = 0 #deklarasi variable class diisi 0
7    def initialize (judul, artis, durasi) #initialisasi var global
8  secara private
9      @judul = judul
10     @artis = artis
11     @durasi = durasi
12     @diputar = 0
13   end
14
15   def putar #method aksi
16     @diputar += 1 #ingat +=1 sama ssprti diputar = diputar + 1
17     @@varClass += 1
18    puts "Lagu #{@judul} diputar #{@diputar} kali, total #{@@varClass}
19 kali putar lagu."
20   end
21 end
22
23 if __FILE__ == $0
24   lagu1 = Lagu.new("Honey","Laruku", 3.34) #lagu pertama
25   lagu2 = Lagu.new("You got a Friend","Ken Hirai", 4.42)
26   lagu1.putar
27   lagu2.putar
28   lagu1.putar
29   lagu1.putar
```

```
30    lagu2.putar
31  end
```

Kode di atas merupakan contoh kode penggunaan variable Lokal dalam Class. **Variabel lokal dalam mendeklarasikannya menggunakan tanda** @@. Seperti di atas *@@varClass* merupakan variabel lokal yang terus dan hanya bisa digunakan di dalam class tersebut. Kita lihat hasil eksekusinya di dalam **terminal** :

```
hirokakaoshi@hirokakaoshi-Lenovo-G450:~/RubyDOJO$ sudo chmod +x VarClass.rb
hirokakaoshi@hirokakaoshi-Lenovo-G450:~/RubyDOJO$ ./VarClass.rb
Lagu Honey diputar 1 kali, total 1 kali putar lagu.
Lagu You got a Friend diputar 1 kali, total 2 kali putar lagu.
Lagu Honey diputar 2 kali, total 3 kali putar lagu.
Lagu Honey diputar 3 kali, total 4 kali putar lagu.
Lagu You got a Friend diputar 2 kali, total 5 kali putar lagu.
```

Perintah *sudo chmod +x* untuk mengubah priviledge dari file *Varclass.rb* supaya mendapat izin di eksekusi. Cukup lakukan sekali saja di setiap file rb yang baru kita buat. Perintah *./VarClass.rb* untuk eksekusi file tersebut.

Kemudian lihat hasil eksekusi di CMD :

```
C:\Users\Muhammad Singgih Z.A>cd C:\Ruby192\bin\RubyDOJO

C:\Ruby192\bin\RubyDOJO>VarClass.rb

Lagu Honey diputar 1 kali, total 1 kali putar lagu.

Lagu You got a Friend diputar 1 kali, total 2 kali putar lagu.

Lagu Honey diputar 2 kali, total 3 kali putar lagu.

Lagu Honey diputar 3 kali, total 4 kali putar lagu.
```

```
Lagu You got a Friend diputar 2 kali, total 5 kali putar lagu.
```

Perhatikan perintah di method **def putar**, variabel lokal dan variable global di increment 1 setiap kali di eksekusi. Perhatikan hasil eksekusi, mengapa nilai dari var *@diputar* tiap objek baru berbeda-beda sedangkan variabel lokal @@varClass terus bertambah???

Itu karena var @diputar merupakan variabel global yang berlaku untuk objek instance baru atau variabel objek yang dideklrasikan dengan tipe *class* ***Lagu***. Jadi, setiap ada Variabel objek yang dideklarasikan, maka variabel global yg dideklarasikan dalam class berlaku hanya untuk variabel objek tersebut. Namun berbeda hal nya dengan variabel Lokal Class, dia terus berlaku sama untuk setiap variabel objek yang menggunakan *Class Lagu*. Maka hasilnya seperti di atas deh … (^__^)Y

II. Kontrol Akses dalam Class(Access Control)

Berikut ini ..... adalah.... Type Hak Akses dalam Class... YEAHH

Di edisi berapa gitu, saya lupa pernah menyinggung mengenai ini , ada 3 jenis hak / perlindungan akses dalam Class di OOP.

1. **Public methods**
   Bisa digunakan oleh apa saja, deklarasi tipe akses kontrol tidak diperlukan. Method secara otomatis atau default merupakan public.

2. **Protected methods**
   Hanya bisa digunakan oleh objek dalam class(induk class) dan subclass(anak Kelas). Hanya untuk anggota keluarga nya saja.

3. **Private methods**
   Tidak dapat digunakan oleh siapa saja, hanya bisa digunakan dalam class itu sendiri. Namun tidak untuk subclass dan sebagainya. Tidak dapat dipanggil oleh var objek bertipe class tersebut.

Di kode sumber sebelumnya, setiap method yang dibuat merupakan method bertype Public sehingga dapat digunakan oleh siapa saja yang berlaku sebagai Class tersebut. Namun tetap, di Ruby akses kontrol pun ada flexibiltasnya, namun belum akan dibbahasa sekarang :)

Masih belum ngerti ya?? baiklah kalau begitu Akang Caritakeun nya Ilustrasina Kieu *nyundaEuyBaraya

Deklarasi Tokoh:
Kancil sebagai **Public Methods**
Tikus sebagai **Protected Methods**
Marmut sebagai **private Methods**

Kancil, Tikus dan Marmut

Pada suatu ketika ada 3 hewan yang begitu terkenal se antero hutan kalimantan. Yaitu *Kancil si Dermawan*, *Tikus si Hemat* dan *Marmut si Kikir*.

Sang Kancil merupakan hewan yang paling disenangi dan disegani di Hutan, hewan-hewan lain begitu menghormati dan senang ketika berada di dekatnya selain baik dia juga begitu ramah dan dermawan, ketika ada hewan lain yang membutuhkan dia selalu menolong dan memberikan apa yang bisa dia berikan.

Hingga pada suatu saat, datang *sang Tikus* meminta pertolongan ke *sang Kancil*. Dia meminta agar *kancil* mau memberikannya beberapa Buah Kacang tanah dari kebunnya untuk mereka budidayakan di halamannya.

Lalu *sang kancil* memberikannya dengan senang hati, kemudian *Tikus* pulang ke rumah dengan riang dan memanggil semua anggota keluarganya untuk menanam kacang bersama-sama di halaman mereka.

Setelah beberapa lama, akhirnya tiba waktunya untuk panen kacang. Keluarga *tikus* pun riang gembira, mereka menikmati kacang hasil dari halaman mereka namun tidak membagikannya ke hewan sebelah meskipun hewan lain memintanya untuk berbagi.

Sang tikus hanya perduli kepada keluarganya saja.

Lalu, *Sang Marmut* datang. Marmut merupakan saudara jauh dari tikus jadi *sang Tikus* pun Membagikan hasil panen kacang sebanyak 1 kantong ke *Marmut*.

Kemudian *sang Marmut* pulang ke rumahnya, namun ketika smpai depan rumah dia

menyembunyikan kantong kacang tersebut sehingga ketika dirumah tak ada yang tahu apabila *sang Marmut* mempunyai kacang.

Tak lama kemudian *sang Marmut* ketahuan sedang memakan kacang oleh adik dan temanya, lalu mereka meminta sedikit kacang tersebut, namun *sang Marmut* tidak mau berbgai dengan adiknya.

----------TAMAT----------

Sudah selesai baca kan? Kalau belum baca yang betul ya sampai mengerti maksud dari cerita tersebut. Kalau sudah, saatnya kita keee.... KODE PROGRAM YEEEEYYY

Program 7.2 : Contoh penggunaan Access Control di Ruby (sampleCA.rb)

```
1   #!/usr/bin/ruby
2
3   class MyClass
4       def method1  # Kntrol Akses method default sebagai 'public'
5         #...
6       end
7     protected      # methods berikutnya akan menjadi 'protected'
8       def method2  # method2 jadi 'protected'
9         #...
10      end
11    private        # methods berikutnya akan menjadi 'private'
12      def method3  # method3 jadi 'private'
13        #...
14      end
15    public         # methods berikutnya akan menjadi 'public'
16      def method4  # method3 jadi 'public'
17        #...
```

```
18      end
19  end
```

Sebenarnya ada cara lain untuk mendefinisikan type pengamanan atau akses kontrol terhadap methods, lihat cara berikut :

Program 7.3: Contoh penggunaan Access Control di Ruby (sampleCA_2.rb)

```
1   #!/usr/bin/ruby
2
3   class MyClass
4     def method1  # Kontrol Akses method default sebagai 'public'
5       #...
6     end
7     def method2
8       #...
9     end
10    def method3  # method3 jadi 'private'
11      #...
12    end
13    def method4  # method3 jadi 'public'
14      #...
15    end
16    public :method1, :method4#definisikan method1 dan method4 sbg public
17    protected :method2 # definisikan method2 sbg protected
18    private :method3  # definisikan method3 sbg private
19  end
```

Buehehehe...gampang kan wkwkwk...

Kalau program 7.2 mendefisikan type method sebelum membuat method, sedangkan program 7.3 itu mendefinisikan type method setelah method di deklarasikan sebelumnya.

Coba aja deh ya kode-kode di atas, ya meskipun gakan keluar apapun haha..
Ah udah ah sekarang ke contoh yang sebenarnyah berisi dan beraksi..YEAAHHH

Program 7.4: akses kontrol method (AKMethod.rb)

```
1  #!/usr/bin/ruby
2
3  class Ganteng #deklarasi class induk yaitu ganteng
4    def initialize #
5      @angka = 10
6    end
7
8    def cek_angka
9      if @angka%2 == 0
10       puts "#{@angka} adalah Genap"
11     end
12   end
13
14   def tampil_angka
15     puts cek_angka #memanggil method cek_angka yang Private
16     print "angka dipangkat 2 hasilnya", @angka=@angka*@angka
17   end
18
19   protected :tampil_angka
20   private :cek_angka
21 end
22
23 class Cantik < Ganteng
24   def tampil_anak
25     puts tampil_angka; #memanggil method tampil_anak yang Protected
26   end
27 end
28
29 if __FILE__ == $0
```

```
30    Jenny = Cantik.new
31    #Jenny.tampil_angka  #kalau  di  uncomment  bakal  error  lho  karena
32  protected tak bisa di akses dari luar
33    #Jenny.cek_angka  #juga  akan  error  deh  karena  Private  tak  bisa  di
34  akses diluar class
35    puts Jenny.tampil_anak
36  end
```

Coba deh Compile dan Run di Terminal atau CMD. Hasilnya sperti berikut di terminal

```
hirokakaoshi@hirokakaoshi-Lenovo-G450:~/Project/RubyDOJO$ chmod +x AKMethod.rb
hirokakaoshi@hirokakaoshi-Lenovo-G450:~/Project/RubyDOJO$ ./AKMethod.rb
10 adalah Genap
nil
angka dipangkat 2 hasilnya100nil
nil
```

Sekarang tampil di CMD :

```
C:\Ruby192\bin\RubyDOJO>AKMethod.rb

10 adalah Genap

angka dipangkat 2 hasilnya100
```

Cukup jelas bukan :D . Di **program 7.4** terdapat perpangkatan, dan pengecekan bilangan ganjil di beberapa method berbeda namun. Inilah yang disebut dengan enkapsulasi, yaitu pelindungan bertingkat dalam OOP.

Di kelas pertama terdapat 3 method | initialize, cek_angka, dan tampil_angka | , kemudian di akhir kelas di definisikan type method nya. Kemudian

dibuat kelas turunan dari *class Ganteng*, yaitu *class Cantik* yang merujuk ke induknya. Sehingga semua method di dalam *class Ganteng* dapat di akses oleh c*lass Cantik*, kecuali method yang bertipe private method.

Kemudian buat perintah untuk mencari tahu hasil dari kedua kelas tadi dengan membuat variable objek dari *class Cantik*. Lalu ekseperimen deh memanggil method dari *class Cantik* dan *class Ganteng*. Kemudiaannn....lihat hasilnya :D

Udah dulu aja ah, ini namnaya juga pengenalan Class season 2. Cukup untuk sementara ini. Thanks ^__^

Referensi :
Programming Ruby, 2nd Edition | pragmatic Programmer's Guide
http://rubylearning.com/satishtalim/ruby_access_control.html

III. Rangkuman

Pemirsa, tak terasa sekarang sudah di penghujung acara Ruby edisi 7 nih. Apa saja sih yang sudah kita pelajari tadi :

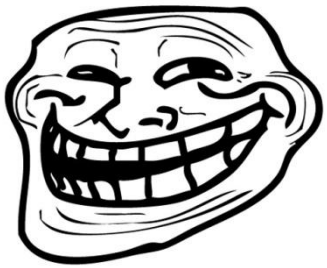
- Belajar menggunakan Variabel lokal di Class
- Belajar enkapsulasi dan type kontrol akses method dalam kelas, dengan kata lain melakukan pendefinisian tingkat keamanan dari methods dalam class.

Yaph hanya segitu lho di edisi ini, tapi lihat...panjang banget gilaaakkkss... Untuk edisi selanjutnya kita kembali mundur, kita belajar dulu mengenai algoritma dasar dalam pemograman, bahasa nya pakai ruby gitu belajarnya hheu (^o^)9

Untuk semakin mantap penguasaan class dan local variable, coba kerjakan latihannya ya dan kunci jawabannya minggu depan baru saya kasih.

1. Buat penghitungan Volume dari Balok, panjang lebar tinggi simpan di method type protected. Kemudian buat class turunan dan buat method public untuk melakukan penghitungan volume balok tersebut.
2. Buat penghitungan sebanyak 10 kali dengan nilai berbeda, kemudian setiap eksekusi volume selalu berikan keterangan penghitungan ke berapa yang dilakukan.

Sederhana Bukan :P hahahahahhaha....



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 pemograman dan algoritma pembaca.

Salam, hirokakaoshi | nyankoder's Ruby.

Selanjutnya ==> penggunaan statemen IF dalam Ruby (^_^)

[Perl]

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

Di Saat Harus Memilih

Ada saatnya dimana seorang super hero akan merasakan dilema antara memilih menjadi pribadinya yang seorang manusia biasa atau tetap menjadi pahlawan super untuk menumpas kejahatan. Wajarlah, super hero juga kan manusia, dia juga pengen ngerasain pacaran seperti manusia biasa, pengen ngerasain nonton sama ceweknya, pengen ngerasain punya keluarga, punya anak dan hidup aman, nyaman dan tentram tanpa harus mengurusi urusan orang lain. Namun seperti kata paman Ben sebelum ia meninggal, ***"With great power comes great responsibility"***, dalam kekuatan yang besar terkandung tanggung jawab yang besar. Maka, mau tidak mau diapun harus memilih. Setidaknya itulah yang pernah dirasakan oleh Peter Parker, Spiderman.

Tentu saja memilih dan saat-saat memilih tidak hanya dirasakan oleh super hero saja, siapa pun pernah, ya termasuk programer bahkan mungkin termasuk kamu. Ketika sang mantan datang dan meminta kamu buat balikan lagi sama dia, pasti kamu harus memilih apakah akan menerima tawaran menariknya itu atau kamu abaikan dan memilih untuk melanjutkan hidupmu tanpa dia. Ketika kamu punya dua orang gebetan, sebut saja misalnya D1 dan D2, kamu suka keduanya, tapi kamu bingung yang manakah yang harus kamu pilih, apakah D1, D2 ataukah tidak sama sekali, atau mungkin keduanya. Di saat kamu harus memilih apakah harus menerima project ataukah fokus mengerjakan skripsi. Atau ketika kamu harus memilih apakah harus memilih pacar baru atau laptop baru. Apa pun kondisi pilihannya, yang pasti pilihan kamu akan menentukan apa yang akan kamu hadapi kedepannya.

Oh ya jou, ngomong-ngomong masalah pemilihan saya punya kasus nih, misalnya kamu lagi ngerjain beberapa project sekaligus, anggap saja ada 3 project.

Nah ketika kamu lagi fokus buat nyelesein project-project itu, tiba-tiba ada klien baru yang nawarin project ke kamu dengan budget yang lebih besar dari project-project yang sebelumnya. Kira-kira bakal kamu terima nggak sih, sedangkan project-project sebelumnya hampir mendekati deadline, jadi harus kelar cepet dan nggak bisa ditunda.

Baik, sepertinya sudah saatnya untuk masuk ke materi kita. Oh ya, tunggu dulu, sebelum masuk ke materi, mari kita review dulu pembahasan di edisi sebelumnya. Di edisi sebelumnya kita udah membahas tentang Array dan segala ke-alay-annya. Buat mengingat edisi sebelumnya, coba deh jawab beberapa pertanyaan berikut ini.

1. Apa kaitannya antara list dan array?
2. Gimana cara mengakses array?
3. Bisakah sebuah array yang udah punya elemen, ditambahkan elemennya, kalau bisa, gimana caranya?

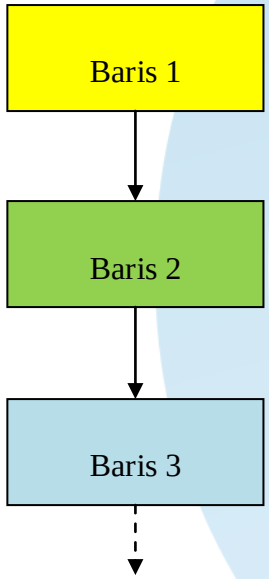
Jawab dulu pertanyaan itu ya. Kalau udah bisa jawab dengan baik dan benar baru deh boleh ngelanjutin bacanya. Hehehe.. Seperti kesepakatan kita di edisi sebelumnya, buat mempelajari hal baru bukan berarti kita harus ngelupain hal lama yang telah kita pelajari. Jangan mentang-mentang udah dapet yang baru, yang lama langsung dilupain begitu aja. Kan nggak keren jadinya.

Oke, kembali ke topik.

Tadi kita udah bahas tentang pemilihan yang terjadi di dunia nyata kita. Nah, ternyata pemilihan itu juga terjadi lho di dalam pemrograman. Nah lho.... Dalam pemrograman, biasanya pemilihan sering disebut juga percabangan. Ya konsepnya mirip-miriplah sama di dunia nyata, ada kondisi, ada pilihan. Mau tau tentang percabangan atau pemilihan?? Mari kita lanjut...

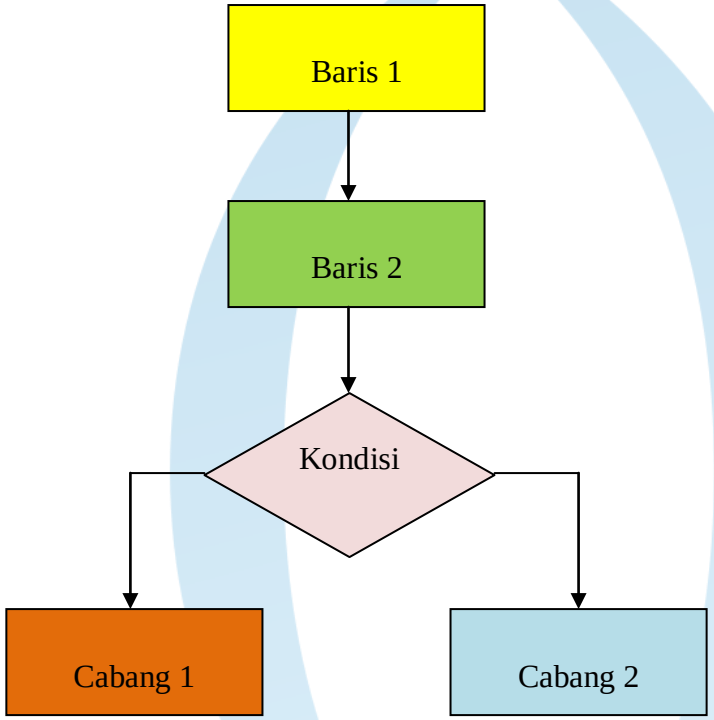
Konsep Dasar Percabangan

Coba deh perhatiin kode-kode program yang udah pernah kita tulis sebelumnya. Kalau kamu perhatiin baik-baik, pasti semuanya akan dijalankan baris per baris oleh interpreter. Jadi, yang pertama dikerjain adalah baris pertama, dan setelah baris pertama selesai, baru deh yang dikerjain selanjutnya adalah baris kedua. Begitulah seterusnya. Kalo digambarkan dalam flow chart kira-kira gini nih.



Kesimpulannya adalah semua baris atau semua statment yang ada di kode program akan dieksekusi secara keseluruhan tanpa terkecuali. Nah, itu adalah pernyataan biasa, dan bukan sebuah pemilihan/percabangan, ya karena nggak ada yang harus dipilih. Dalam percabangan, biasanya interpreter akan membaca terlebih dahulu kondisi mana yang sesuai, setelah itu akan menjalankan pernyataan yang

memenuhi syarat kondisi saja. Jadi nggak semua pernyataan akan dieksekusi di dalam program. Kalo digambarin gini nih.



Nah, kalo di gambar atas baru merupakan sebuah percabangan. Kita bisa liat, dari baris 1 dan 2 program dijalankan biasa aja. Nah kemudian setelah ada kondisi, maka program hanya menjalankan satu cabang aja, yaitu Cabang 1 atau Cabang 2, tergantung kondisi mana yang memenuhinya. Gampang bukan?? Sebenernya dalam prakteknya Cabang yang ada di dalam percabangan itu bisa hanya satu buah, dua buah ataupun lebih.

if

Pernyataan if hanya akan diproses bila kondisi sesuai dengan persyaratan, dan bila kondisi tidak sesuai maka pernyataan tidak akan dijalankan. Adapun bentuk dari statment if adalah seperti berikut:

```
If (kondisi) {
      aksi..
}
```

Baik, biar lebih gampang mari kita coba dengan contoh aja deh.

Program: pernyataan_if.pl

```
1   #!/usr/bin/perl
2   # nama program : pernyataan_if.pl
3
4   $angka = 9;
5
6   if ($angka == 9){
7        print "Pernyataan dieksekusi \n";
8   }
9
10  print "ini di luar if \n";
11  print "selesai \n";
```

Maka, apabila dijalankan hasilnya adalah sebagai berikut:

```
$ perl pernyataan_if.pl
Pernyataan dieksekusi
```

```
Ini di luar if
Selesai
$
```

Dari program di atas jelas kan terlihat kalo pernyataan di dalam if dijalankan karena persyaratan pada kondisinya terpenuhi, yaitu $angka == 9. Coba deh ganti angka 9 pada kondisinya, misalnya kita ganti jadi 7, seperti kode berikut:

Program: pernyataan_if_bukan.pl

```
1   #!/usr/bin/perl
2   # nama program : pernyataan_if_bukan.pl
3
4   $angka = 9;
5
6   if ($angka == 7){
7        print "Pernyataan dieksekusi \n";
8   }
9
10  print "ini di luar if \n";
11  print "selesai \n";
```

Maka, apabila dijalankan hasilnya adalah sebagai berikut:

```
$ perl pernyataan_if_bukan.pl
Ini di luar if
Selesai
$
```

Tuh kan, ternyata kalo kita ganti jadi angka 7, maka pernytaannya nggak akan dieksekusi. Oh ya, tanda == adalah sebuah operator perbandingan untuk numerik, adapun jika dikenakan pada string maka bisa menggunakan operator 'eq', masih inget kan?? Kalo lupa, silahkan cek kembali di edisi ke-3 dan ke-4. Pembahasannya ada di situ.

Program: pernyataan_if_string.pl

```
1   #!/usr/bin/perl
2   # nama program : pernyataan_if_string.pl
3
4   $warna = 'hijau';
5
6   if ($warna eq 'hijau'){
7        print "Pernyataan dieksekusi \n";
8   }
9
10  print "Ini di luar if \n";
11  print "Selesai \n";
```

Output:

```
$ perl pernyataan_if_string.pl
Pernyataan dieksekusi
Ini di luar if
Selesai
$
```

Gimana gampang kan?? Sebenernya kalo kamu mau kamu juga bisa kembangin kondisinya dengan menggunakan operator boolean.

if else

Kalo statment if di atas kan cuma buat satu kondisi aja, dimana kalo kondisi nggak tercapai maka statment nggak akan dijalanin. Nah untuk statment if else maka akan terdapat dua kondisi yang memungkinkan untuk dieksekusi tergantung bagaimana keadaan dan syarat kondisi mana yang akan terpenuhi. Bentuk adalah sebagai berikut:

```
If (kondisi) {
        Pernyataan...
} else {
        Pernyataan...
}
```

Pernyataan pertama adalah apabila syarat kondisi terpenuhi, sedangkan pernyataan kedua adalah jika syarat kondisi tidak terpenuhi. Biar lebih gampang, mari kita coba dengan contoh.

Program: pernyataan_if_else.pl

```
1   #!/usr/bin/perl
2   # nama program : pernyataan_if_else.pl
3
4   $angka = 9;
5
6   if ($angka == 9){
7       print "Pernyataan aksi \n";
8   } else {
9       print "Pernyataan alternatif \n";
10  }
11
```

```
print "Ini di luar if \n";
print "Selesai \n";
```

Output:

```
$ perl pernyataan_if_else.pl
Pernyataan aksi
Ini di luar if
Selesai
$
```

Pada contoh di atas, yang tereksekusi adalah pernyataan pertama, karena syarat kondisi yaitu $angka == 9 terpenuhi. Silahkan ganti saja kondisi yang ada atau mengganti nilai dari variabel $angka, pasti hasilnya akan beda.

if else if

Kadang-kadang kita juga butuh melakukan pemilihan dengan lebih dari dua kondisi, maka dalam kondisi itu kita bisa menggunakan if else if. Adapun formatnya adalah sebagai berikut:

```
If (kondisi 1){
        Aksi 1
} elsif (kondisi 2){
        Aksi 2
} else {
        Aksi 3
}
```

Aksi 1 hanya akan dieksekusi jika kondisi 1 terpenuhi, dan bila tidak terpenuhi maka akan dilakukan pengecekan untuk kondisi kedua, jika kondisi kedua terpenuhi maka Aksi 2 yang akan dieksekusi. Dan jika kondisi 2 pun tidak terpenuhi, maka yang akan dieksekusi adalah Aksi 3.

Biar lebih mudah mari kita coba dengan contoh program. Coba tulis kode yang ini jou.

Program: pernyataan_if_else_if.pl

```
1   #!/usr/bin/perl
2   # nama program : pernyataan_if_else_if.pl
3
4   $angka = 7;
5
6   if ($angka == 9){
7        print "Pernyataan aksi 1 \n";
8   } elsif ($angka == 7) {
9        print "Pernyataan aksi 2 \n";
10  } else {
11       print "Pernyataan aksi 3 \n";
12  }
13
14  print "Ini di luar if \n";
15  print "Selesai \n";
```

Output:

```
$ perl pernyataan_if_else_if.pl
Pernyataan aksi 2
Ini di luar if
Selesai
$
```

Gimana jou?? Ternyata yang dieksekusi adalah pernyataan 2, ya karena kondisi kedua cocok dengan variabel $angka, yaitu bernilai 7. Sederhana bukan?

unless

Ada satu statment percabangan lagi nih, yaitu unless. Sebenernya konsepnya sama dengan if. Hanya saja jika if, pernyataan aksi akan diekseskusi jika kondisi terpenuhi, nah buat unless, pernyataan aksi akan dieksekusi jika kondisi nggak terpenuhi. Ya kebalikannya gitu deh.

Format penulisan kodenya gini nih:

```
unless (kondisi) {
      Aksi
}
```

Baik, langsung aja coba deh praktekin program ini:

Program: pernyataan_unless.pl

```
1  #!/usr/bin/perl
2  # nama program : pernyataan_unless.pl
3
4  $angka = 7;
5
6  unless ($angka == 8) {
7      print "angkanya nggak sesuai dengan syarat kondisi \n";
8  }
```

Output:

```
$ perl pernyataan_unless.pl
angkanya nggak sesuai dengan syarat kondisi
$
```

Tuh kan, aksinya malah dieksekusi pas kondisinya salah. Hehehe...

Baik, mungkin untuk edisi ke-7 ini, materi kita sampe di sini dulu yua.. Jagan lupa buat terus belajar ya, karena yang namanya pemrograman itu emang mesti dicoba terus dengan berbagai keadaan. Semakin kita sering nyoba, semakin piawai kita dalam memanipulasi logika kode kita. Dan semakin mahir juga kita dalam membuat program. Oh ya, di edisi selanjutnya kita akan berhadapan dengan pengulangan, jadi jangan lupa nantikan edisi selanjutnya, dijamin nggak kalah seru.

Oh ya, kalau kamu jadi super hero, kamu bakal pilih siapa?? Superman, spiderman, batman, ironman, hulk, atau siapa?? Kalau saya sih lebih suka spiderman, tapi kalo boleh punya kekuatan tambahan, saya pengen bisa teleport, biar nggak usah bergelantungan di gedung-gedung. Soalnya kan di Indonesia gedung-gedunganya nggak tinggi-tinggi amat. Kalo pun ada yang tinggi-tinggi itu pun cuma sedikit, dan jarang-jarang, bahkan nggak di semua kota. Kan jadi lucu kalo spiderman naek angkot. Mangkanya kalo bisa teleport kan jadi nggak usah cape. Hehehe

Oke cong, sampe ketemu lagi di edisi ke-8 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, sebelum pulang (kayak sekolah aja), saya punya oleh-oleh nih. Silahkan dicicipi buat menguji pemahaman kamu di edisi ini:

Buatlah sebuah program pemilihan yang berisi beberapa kondisi. Detail programnya begini nih, buatlah sebuah variabel dengan nama $pilih yang diinput oleh user dengan menggunakan <STDIN>, kemudian selanjutnya buatlah statment if else if, dimana jika user memasukan input angka 1, maka akan tampil tulisan "Anda memilih menu 1", jika user memasukan angka 2, maka akan tampil tulisan "Anda memilih menu 2", dan jika user memasukan angka selain angka 1 dan 2 maka akan tampil tulisan "Anda memilih menu selain 1 dan 2".

[JavaScript]

Toni Haryanto
@yllumi
toha.samba@gmail.com
http://toniharyanto.cs.upi.edu

Events

Pernah nggak kamu perhatikan, ketika kamu bermain komputer, misalnya main game perang-perangan kayak CS, COD, atau lain sebagainya, disitu kamu berperan sebagai sudut pandang orang pertama seolah kamu berada di dalam game tersebut. Di dalam game, kamu bisa melakukan apapun, seperti menembak, mengambil senjata atau bom, dan membeli senapan. Lalu, darimana karakter game kita tahu kalo kita mau nembak, atau melempar granat, atau aksi-aksi lainnya, lalu melakukan aksi sesuai yang kita inginkan?? Owh, tentu saja, kita sebagai *game player* kan dapat mengendalikan karakter pemain kita di dalam game, dengan cara mengklik mouse dan keyboard, atau menekan tombol-tombol *joysick*! Masa iya kita ngendaliin karakter kita dengan pikiran secara langsung, lalu tiba-tiba saja karakter kita mengikuti kehendak kita?! Hmm, mungkin kedepannya cara bermain game akan sampai ke tahap semacam itu. Tapi, yang perlu kita garis bawahi, sebagai seorang programmer tentunya, adalah bahwa game tersebut bisa seasyik dan seinteraktif itu, karena kita dapat mengendalikan karakter pemain kita sendiri, dan itu tidak akan dapat dilakukan kalo si aplikasi atau game tidak dapat menangkap maksud yang ingin kita sampaikan melalui penekanan joystick, mouse dan keyboard. Nah, di sinilah peran mekanisme penangkapan input oleh aplikasi dari kita selaku pengguna, yang umumnya dikenal dengan istilah *event handler* atau penanganan event.

Konsepnya sederhana. Untuk dapat membuat interaktivitas di dalam sebuah aplikasi, maka aplikasi tersebut mesti dapat meng*handle* atau menangani setiap masukan yang dibutuhkan aplikasi untuk menjalankan program.

Oke, biar ga terlalu ribet ama konsep, mari saya jelaskan dengan contoh. Kalo kamu sudah baca pembahasan JavaScript di edisi kemarin (edisi 6 tentang fungsi),

saya membuat contoh yang menggunakan event. Yang mana cobaa?? Coba cek lagi cek lagi edisi sebelumnya.. :P

Program: fungsi_3.html

```
1   <html>
2   <head>
3       <title>Fungsi</title>
4   <script type="text/javascript">
5   function sapaan(nama)
6   {
7      document.write("Halo " + nama + "! Selamat belajar fungsi.");
8   }
9   </script>
10  </head>
11  <body>
12       Nama: <input type="text" id="nama" />
13       <button id="proses"
14        onclick="sapaan( document.getElementById('nama').value )" >
15       Proses</button>
16  </body>
17  </html>
```

Yups, di contoh pada program fungsi_3.html. Kalo kamu bisa nangkep maksud penjelasan di awal tentang *event handler*, maka alur berfikirnya mungkin kira-kira seperti ini: bagian dalam aplikasi yang bertugas menangani masukan atau input dengan menjalankan suatu tugas begitu inputan tersebut ditangkap oleh program. Yap, event handler yang kita maksud pada contoh di atas adalah baris kode **onclick** !

Oke, kita lihat baris kode ke-14 yang bertuliskan onclick=".. dst. Awalnya kita punya sebuah fungsi (yang masih bingung apa itu fungsi, kemungkinan besar melewatkan pembahasan di edisi ke-6, so silakan dibaca dulu) yang bernama sapaan(). Ketika program dijalankan maka setiap baris kode di dalam program akan dieksekusi. Tapi berbeda untuk fungsi. Ketika pembacaan program sampai pada baris

deklarasi fungsi, fungsi tersebut tidak langsung dieksekusi, tetapi disimpan dulu di dalam memori dan baru akan dijalankan ketika fungsi tersebut dipanggil. Nah, pada contoh di atas, fungsi dipanggil di dalam baris kode onclick, yang artinya program akan menjalankan fungsi sapaan() ketika button yang memiliki event handler onclick diklik. Atau dengan kata lain, ketika button diklik, maka event onclick akan mulai menjalankan tugasnya yang pada kasus ini memanggil fungsi sapaan().

Hmmm.. jadi sudah paham kan apa yang dimaksud event? Onclick adalah salahsatu event handler atau mekanisme penanganan ketika event 'klik' dilakukan.

Jadi Event Adalah onclick?

Hmmm.. lebih tepatnya, onclick adalah salahsatu event. Ada banyak event yang bisa kita berikan ke program kita, tergantung dari ketersediaan event handler di dalam bahasa pemrograman tersebut. Kalo kamu ngasih event 'berpikirSupayaProgramMenuliskanSemuaYangAdaDiPikiranSaya', sampai kapan pun program kamu ga akan pernah bereaksi terhadap event kamu itu karena tidak ada event handler yang tersedia untuk event semacam itu. So, untuk saat ini kita manfaatkan event handler yang ada dulu yaaa... #fiuhhh

Jadi, ada event apa saja di JavaScript? Berikut daftar event yang bisa kamu pelajari:

Attribut	event ditangkap ketika..	IE	F	O	W3C
Onabort	pemuatan gambar disela/dihentikan	4	1	9	Ya
Onblur	suatu elemen kehilangan fokus	3	1	9	Ya
onchange	isi dari suatu kolom isian berubah	3	1	9	Ya
Onclick	mouse mengklik sebuah elemen	3	1	9	Ya
ondblclick	mouse mengklik ganda sebuah elemen	4	1	9	Ya

Onerror	muncul kesalahan saat memuat dokumen atau gambar	4	1	9	Ya
Onfocus	suatu elemen dikenai fokus	3	1	9	Ya
onkeydown	keyboard ditekan	3	1	No	Ya
onkeypress	keyboard ditekan dan ditahan	3	1	9	Ya
Onkeyup	keyboard dilepaskan setelah ditekan	3	1	9	Ya
Onload	sebuah gambar atau halaman berhasil dimuat	3	1	9	Ya
onmousedown	tombol mouse ditekan	4	1	9	Ya
onmousemove	kursor mouse dipindahkan	3	1	9	Ya
onmouseout	kursor mouse dipindahkan dari suatu elemen	4	1	9	Ya
onmouseover	kursor mouse dipindahkan ke suatu elemen	3	1	9	Ya
onmouseup	mouse dilepaskan setelah ditekan	4	1	9	Ya
Onreset	button reset diklik	4	1	9	Ya
Onresize	jendela atau bingkai (frame) diubah ukurannya	4	1	9	Ya
Onselect	teks dipilih	3	1	9	Ya
onsubmit	button submit diklik	3	1	9	Ya
onunload	pengguna keluar dari jendela	3	1	9	Ya

Daftar di atas saya dapatkan dari website w3school.com. Kamu bisa belajar lebih dalam untuk setiap eventnya dengan mengunjungi website tersebut. Di dalam pembahasan kali kita hanya akan membahas beberapa event saja sebagai contoh penggunaan.

Studi Kasus Beberapa Event

Oke, untuk onclick saya pikir kita sudah pada-pada paham dengan contoh penjelasan di awal, so kita lanjutkan ke event yang lain, *Okeh*?!

Event onload dapat diterapkan pada elemen <body>, <frame>, <frameset>, <iframe>, <img> dan <link> pada HTML, dan object image, layer, window pada JavaScript.

Aturan penulisan event handler, adalah dengan menyematkan event tersebut sebagai sebuah atribut dari suatu elemen HTML. Kalo liat contoh yang di atas, event onclick disimpan di dalam elemen button, sehingga ketika button diklik, maka event onclick dijalankan. Selain itu, event juga bisa dipanggil sebagai sebuah property suatu object di JavaScript.

onLoad

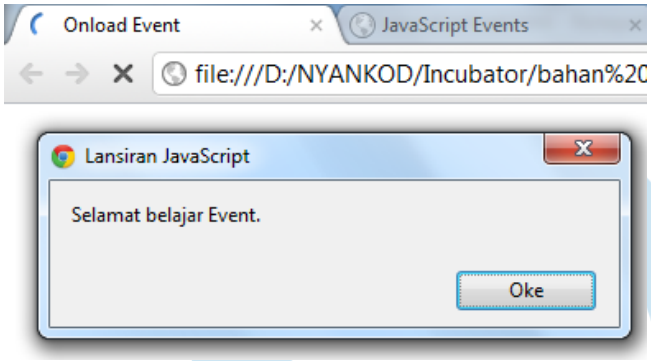
Event onload dijalankan ketika halaman atau gambar selesai dimuat.

Program: onload.html

```
1   <html>
2   <head>
3       <title>Onload Event</title>
4   <script type="text/javascript">
5   function sapa()
6   {
7      alert("Selamat belajar Event.");
8   }
9   </script>
10  </head>
11  <body onload="sapa()">
12  </body>
13  </html>
```

Kalo kita jalankan program di atas, maka akan muncul popup berisi teks sapaan `"Selamat belajar Event."`.

Event onfocus dan onblur dapat diterapkan pada elemen <a>, <acronym>, <address>, <area>, <b>, <bdo>, <big>, <blockquote>, <button>, <caption>, <cite>, <dd>, <del>, <dfn>, , <dl>, <dt>, , <fieldset>, <form>, <frame>, <frameset>, <h1> to <h6>, <hr>, <i>, <iframe>, <img>, <input>, <ins>, <kbd>, <label>, <legend>, <li>, <object>, <ol>, <p>, <pre>, <q>, <samp>, <select>, <small>, , , <sub>, <sup>, <table>, <tbody>, <td>, <textarea>, <tfoot>, <th>, <thead>, <tr>, <tt>, <ul>, <var> pada HTML, dan object button, checkbox, fileUpload, layer, frame, password, radio, reset, select, submit, text, textarea, window pada JavaScript.



onFocus dan onBLur

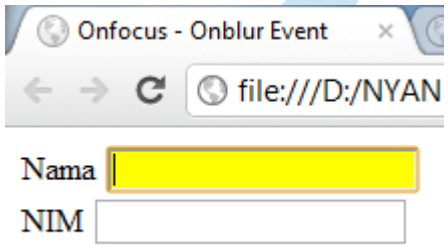
Event onfocus dijalankan ketika suatu elemen dikenai fokus dan event onblur dijalankan ketika suatu elemen kehilangan fokus.

Program: onfocus_onblur.html

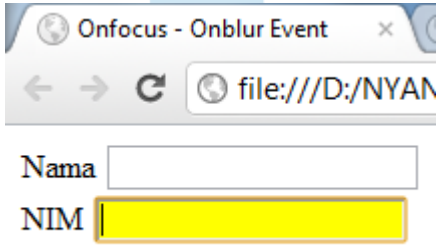
```
1   <html>
2   <head>
3   <title>Onfocus - Onblur Event</title>
4   <script type="text/javascript">
5   function setYellow(x){
6      document.getElementById(x).style.background="yellow";
7   }
8   function setWhite(x){
9      document.getElementById(x).style.background="white";
10  }
11  </script>
12  </head>
13  <body>
    Nama <input type="text" onfocus="setYellow(this.id)"
           onblur="setWhite(this.id)" id="nama" />
    <br />
    NIM <input type="text" onfocus="setYellow(this.id)"
```

```
        onblur="setWhite(this.id)" id="nim" />
</body>
</html>
```

Kalo kita jalankan program di atas, maka akan hasilnya akan seperti ini:



Ketika field nama dikenai fokus, maka background field nama jadi berwarna kuning.



Giliran field NIM yang dikenai fokus, otomatis field Nama kehilangan fokus atau kena event blur, maka field nama akan kembali berwarna putih karena lepas fokus / blur, dan field NIM menjadi kuning.

onSubmit

Event onsubmit dijalankan ketika suatu form disubmit. Biasanya event handler ini digunakan untuk validasi form sebelum data dikirimkan ke server.

Program: onsubmit.html

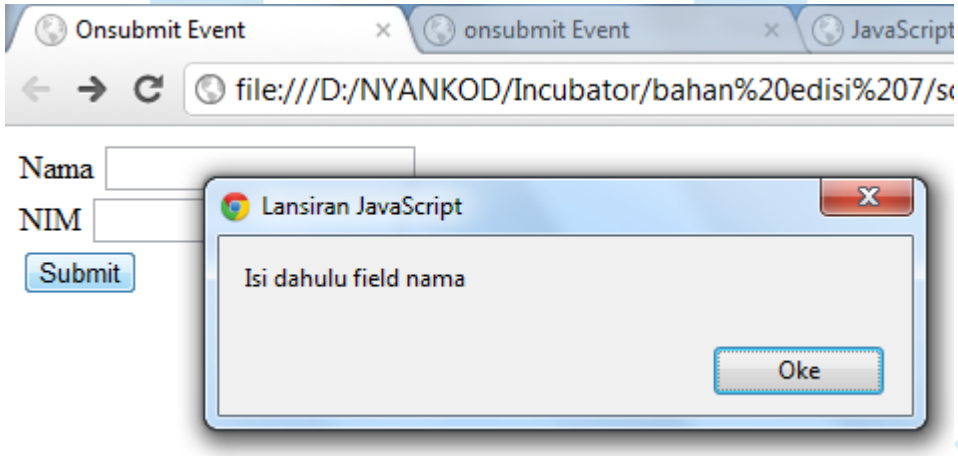
```
1   <html>
2   <head>
3   <title>Onsubmit Event</title>
4   <script type="text/javascript">
5   function validasi(){
6     if(document.getElementById("nama").value == ""){
7       alert("Isi dahulu field nama");
8       document.getElementById("nama").focus();
9       return false;
10    }
11    else if(document.getElementById("nim").value == ""){
12      alert("Isi dahulu field NIM");
13      document.getElementById("nim").focus();
14      return false;
15    } else {
16      alert("Validasi Sukses!");
17      return true;
18    }
19  }
20  </script>
21  </head>
22  <body>
23  <form action="onsubmit.html" method="post"
24    onsubmit="return validasi()">
25  Nama <input type="text" name="nama" id="nama" />
26  <br />
```

Event onsubmit hanya dapat diterapkan pada elemen <form> pada HTML, dan object form pada JavaScript.

```
27    NIM <input type="text" name="nim" id="nim" />
28    <br />
29    <input type="submit" value="Submit" />
30    </form>
31    </body>
32    </html>
```

Kalo kita jalankan program di atas, maka akan muncul form. Kalo kita coba mencoba mensubmit form tanpa mengisi terlebih dahulu field didalamnya, maka berturut-turut akan muncul pesan popup untuk mengisikan terlebih dahulu field nama, kemudian NIM.

Ketika form disubmit, maka form akan menjalankan event handler onsubmit yang bertugas memanggil fungsi validasi. Di fungsi validasi, setiap field dicek apakah sudah ada isinya atau belum. Kalo belum ada isinya, maka program akan memunculkan pesan popup untuk mengisi field tersebut, kemudian memindahkan fokus ke field yang belum diisi.

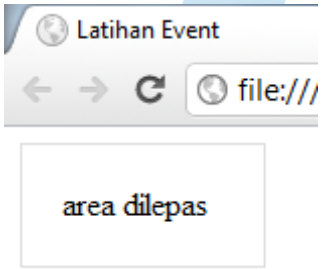


Latihan

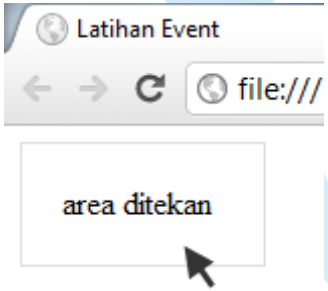
#hariinisayasudah belajar dasar event di javascript, paham kegunaan event handler, dan cara pake event handler! #yeaah #bersyukur

Oke, saya cukupkan pembahasan kali ini tentang event. Konsep event sangat mudah dan bisa diulik sediri. Kamu tinggal buka saja w3schools.com bagian JS Event atau langsung buka referensi tentang semua event yang tersedia di javascript untuk mencoba mempelajarinya satu per satu.

Terakhir, supaya lebih melekat dan ada hasil dari belajarnya, yuk kita latihan *#lagilagilatihan*. He he.. Tugasnya kali ini adalah membuat program seperti ilustrasi berikut:



Awalnya terdapat kotak dengan tulisan "area dilepas".



Ketika area kotak ditekan maka teks dalam kotak akan berubah menjadi "area ditekan", dan menjadi tulisan "area dilepas" lagi setelah mouse dilepas kembali.

Selamat Berkoding Ria dan interaktifkan programmu!! ;D

[Python]

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

Mengulang-ulang di Python? Seperti Apa Yah?

Hola Amigos, Berjumpa lagi dengan Python di Nyankod. Kali ini, di dalam artikel ini. Saya akan mencoba membahas bagaimana membuat sebuah pengulangan atau istilah gaulnya **"Looping"** di Python. *It's Simple and Obvious* untuk membuat sebuah looping di Python.Disini saya sajikan dua buah pengulangan yang standard di Python yaitu **For** dan **While**.J

Let's PyNyankod.

Isi Artikel :

b) Range, generator angka di Python
c) Mencoba pengulangan dengan For
d) Mencoba pengulangan dengan While
e) Referensi
f) Latihan

Range, Generator Angka di Python

Range, yang berarti rentang, jangkauan, kawasan, jarak, atau kiraan dalam Bahasa Indonesia adalah sebuah fungsi built-in yang ada di Python. Untuk apa sih fungsi **range** ini ? Sesuai dengan arti dari kata range itu sendiri. Fungsi ini digunakan

untuk membuat sebuah deret angka. Lebih jelasnya coba perhatikan kode berikut ini. Buka terminal atau cmd Nyankoder dan buka python shell Nyankoder.

```
root@ridwan-laptop:/home/ridwan# python

Python 2.6.5 (r265:79063, Apr 16 2010, 13:09:56)

[GCC 4.4.3] on linux2

Type "help", "copyright", "credits" or "license" for more
information.

>>> range(1, 10)

[1, 2, 3, 4, 5, 6, 7, 8, 9]

>>>
```

Lihat bagian yang ditebali. Disana saya panggil fungsi range dengan dua buah parameter yaitu **1** dan **10**. Parameter pertama adalah angka pertama dari deret yang akan kita hasilkan. Sedangkan parameter kedua adalah angka akhir dari deret yang akan kita hasilkan. Nah, dari contoh diatas secara harfiah : **range(1, 10)** artinya **kita minta sebuah deret yang isinya dimulai dari angka 1 sampai angka kurang dari 10**. Secara default sih deret tersebut dihasilkan dengan kenaikan atau *incremental* satu. Lalu bagaimana kalau Nyankodist ingin menghasilkan deret dengan kenaikan atau *incremental* bukan 1 ? misalnya Nyankodist ingin menghasilkan deret dengan kenaikan 2, 3, 4, 5 atau angka yang Nyankodist inginkan ? Coba deh perhatikan kode berikut ini.

```
>>> range(1, 10, 2)

[1, 3, 5, 7, 9]
```

```
>>> range(1, 10, 3)
[1, 4, 7]
>>> range(1, 10, 4)
[1, 5, 9]
>>> range(1, 10, 5)
[1, 6]
>>> range(1, 10, 11)
[1]
>>> range(1, 10, 0)
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
ValueError: range() step argument must not be zero
>>>
```

Apa yang bisa Nyankodist cermati dari kode diatas ? Yah, betul. Kita tinggal menambahkan parameter ketiga di fungsi *range*. Parameter ketiga dari fungsi *range* adalah angka yang ditentukan oleh programmer untuk menentukan kenaikan atau *incremental* dari deret yang akan kita buat. Nah, dari contoh diatas secara harfiah : **range(1, 10, n)** artinya **kita minta sebuah deret yang isinya dimulai dari angka 1 sampai angka kurang dari 10 dengan kenaikan angka n**. Jika n bernilai kurang dari 10 dan lebih dari 0 maka deret yang dihasilkan akan lebih dari sama dengan 2. Jika n bernilai lebih dari sama dengan 10 maka akan dihasilkan deret

yang hanya berisi angka 1. Jika n bernilai 0 maka akan muncul error seperti pemanggilan **range(1, 10, 0)** di potongan kode diatas. Cukup jelas kan ? Lalu untuk apa fungsi range ini selain menghasilkan deret angka ? Bahasan range selanjutnya ada di topik berikutnya. *Let's Move Nyankodist.*

Mencoba Pengulangan Dengan For

Pengulangan dengan menggunakan **for** bisa dilakukan dengan me-looping deret angka yang dihasilkan dari range dan melooping sebuah list.Tapi untuk bahasan kali ini kita hanya akan bahas membuat looping dengan menggunakan fungsi **range**. Berikut adalah contoh yang bisa Nyankoder coba.

```
>>> for i in range(1,10):
...         print "Baris ke - %d dalam looping python" % (i)
...
Baris ke - 1 dalam looping python
Baris ke - 2 dalam looping python
Baris ke - 3 dalam looping python
Baris ke - 4 dalam looping python
Baris ke - 5 dalam looping python
Baris ke - 6 dalam looping python
```

```
Baris ke - 7 dalam looping python

Baris ke - 8 dalam looping python

Baris ke - 9 dalam looping python

>>>
```

note : jangan lupa indentasi sebanyak 4 spasi untuk membuat sebuah blok statement di Python.Lebih jelasnya Nyankodist coba baca Python Manual.

Dari contoh diatas kita membuat sebuah looping pernyataan "Baris ke – n dalam looping python" dengan rentang 1 hingga kurang dari 10.*Simple* kan ? Lanjut yuk ke bahasan selanjutnya.

Mencoba Pengulangan Dengan While

Lah kalau menggunakan **for** kan hanya bisa membuat looping yang terbatas. Kalau ingin membuat *looping forever* atau *looping* yang tidak terpatok dengan rentang gimana yah ? Misalnya *looping* ditentukan dengan Boolean **True** dan **False.** Python menangani hal tersebut dengan *looping* ***while***. *Looping* ***while*** mempunyai kerangka seperti berikut :

inisialisasi varibel
while expression *sama dengan* ***true****:*
    eksekusi statement yang ada
    di dalam while

Jadi tidak seperti **for** yang mengandalkan deret angka**.** Dalam while kita mengandalkan operasi Boolean untuk menjalankan while. While berjalan jika hasil

ekspresi yang kita tentukan bernilai True.Jika False maka eksekusi while akan dihentikan oleh Python. Berikut adalah salah satu contoh kode looping yang menggunakan **while**.

```
>>> i = 1
>>> while i < 10:

...          print "Baris ke - %d dalam looping python." % (i)

...          i += 1

...

Baris ke - 1 dalam looping python.

Baris ke - 2 dalam looping python.

Baris ke - 3 dalam looping python.

Baris ke - 4 dalam looping python.

Baris ke - 5 dalam looping python.

Baris ke - 6 dalam looping python.

Baris ke - 7 dalam looping python.

Baris ke - 8 dalam looping python.

Baris ke - 9 dalam looping python.

>>>
```

Dari contoh diatas kita awali variabel i dengan angka 1. Kemudian kita masuk ke dalam **while i < 10** di baris ini variabel i dicek apakah i kurang dari 10 atau tidak jika kurang dari 10 maka ekspresi bernilai True dan statement di di dalam while akan dieksekusi. Statement tersebut antara lain adalah baris untuk mencetak **"Baris ke – n dalam looping python."** dan penjumlahan **i += 1**. Nah dalam baris **i += 1** inilah *looping* ***while*** akan berjalan karena angka semula ditambah ditambah satu dan angka i yang baru akan diperiksa lagi dalam **while** untuk proses looping berikutnya. Hal ini dikerjakan terus sampai nilai i lebih dari sama dengan 10. Rame kan ? Rame donk kita kan udah sampai looping untuk melangkah ke dunia programming lainnya bersama Python. Hehe

Referensi :

4. Python Tutorial, http://www.java2s.com
5. Python Documentation, http://www.python.org
6. Python Cookbook, O'Reilly Publisher
7. Python Programming Fundamental, Springer Publisher

Latihan

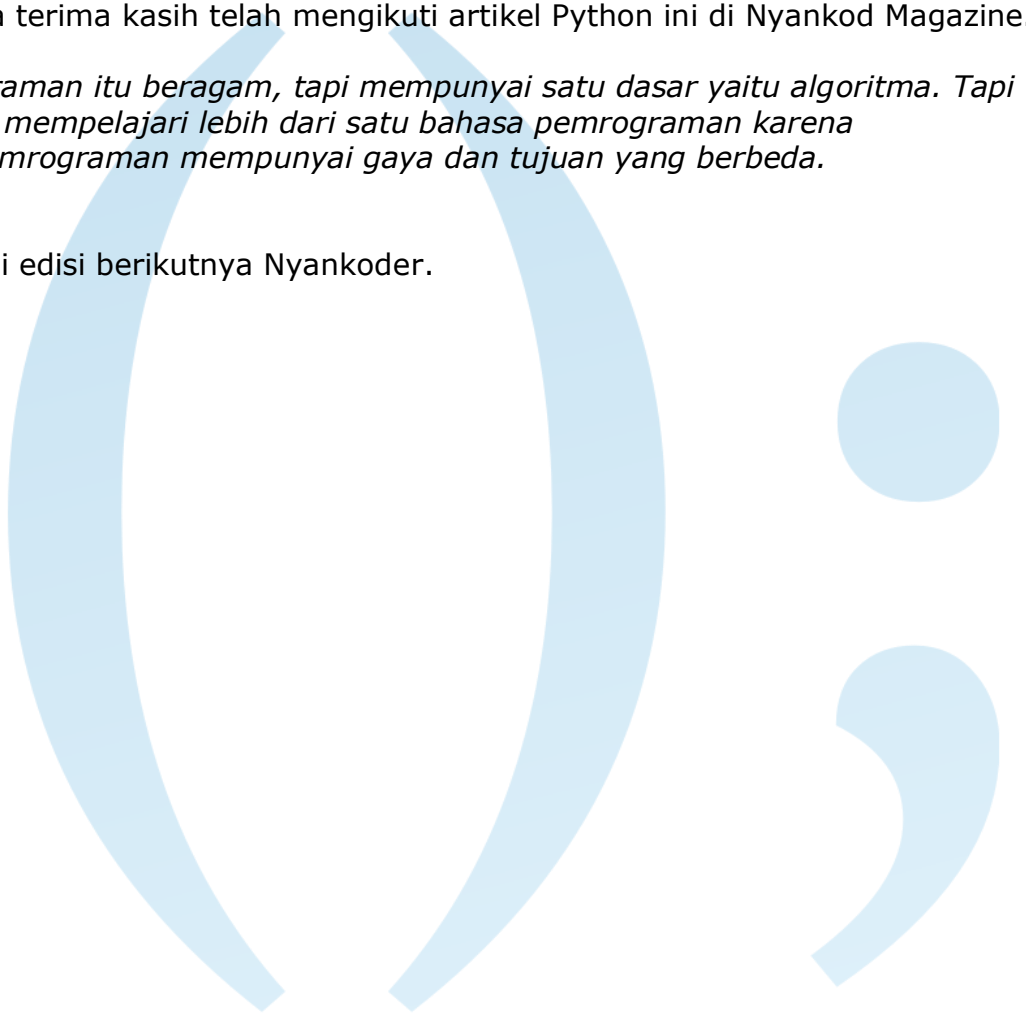
Buatlah sebuah pengulangan yang menampilkan deret angka genap dan ganjil. Nanti file python nya kirim ke ridwanbejo@gmail.com atau ridwanbejo@nyankod.com yah dengan

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 setiapbahasa pemrograman mempunyai gaya dan tujuan yang berbeda.

Sampai jumpa di edisi berikutnya Nyankoder.



[ActionScript]

Tarom Apriyanto
@tarompey
tarompey@gmail.com

Penggunaan GotoAndStop,

Asalamualaikum Wr. Wb. Salam super buat kita semua (udah kayak pak Mario Teguh aja), Kembali dan 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 (komputer maksudnya) atau Lapi (Laptop maksudnya) masing-masing maupun minjem temennya??. (Nyankoder: Tadi nanya apaan ya, sampe lupa??). tentunya baik semua lah dan Nyankodist doain yang baik-baik lah buat Nyankoder semuanya.

Gimana materi kemarin ga susah kan??, gampang dan juga bahan latihan pun gampang banget kan buatnya?? Nah kalau ada yang masih merasa susah, jangan malu-malu untuk menyakan langsung di email saya Ok.

Nah Sob, kembali lagi kita bertemu dan tentunya dengan materi yang baru pula. Di inget-inget dari materi-materi sebelumnya ternyata saya baru inget kalau materi yang akan kita pelajari ini belum diberikan sebelumnya padahal ini adalah materi dasar ActionScript di Flash yang sering dipakai dalam pembuatan-pembuatan aplikasi atau animasi-animasi tertentu, sobat tau itu apa?? Yaa bener, sesuai dengan judul yang tertera di atas yaitu *Penggunaan GotoAndStop*. Haadeeeuh,..maaf ya sobat saya hilaf itu, soalnya banyak kesibukan yang kuliah lah, main bola bareng temen-temen lah, main catur lah dan yang paling sibuk lagi main game susah kalau nglewatin kegiatan yang terakhir ini, suwer dah (Itu bukan kesibukan mas, itu tanda kalau kurang kerjaan…hahahaha).

Oke sob, sebelum kita mulai menelusuri materi hari ini, alangkah baiknya sobat Nyankod atau Nyankoder siapkan posisi yang enak biar materi yang disampaikan dapat dicerna dengan mudah (hahaha kayak ahli kesehatan aja). Minimalnya siapkan secangkir teh hangat untuk menemani penelusuran kita, yaaa nggak?? Hehehee. Oke Check this out..

Menggunakan GotoAndStop

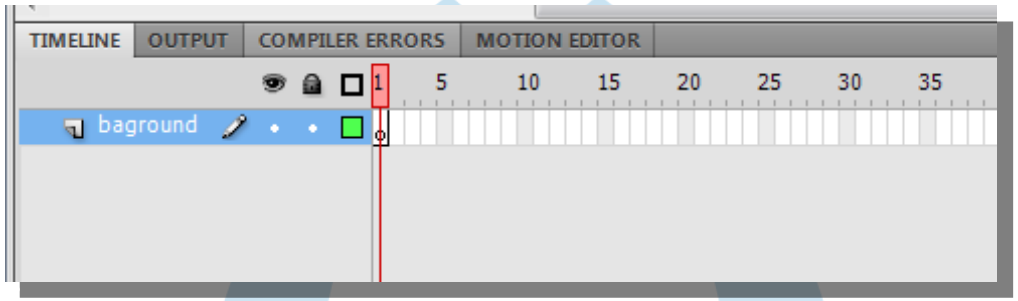
Ada yang tau GotoAndStop artinya apa??? (pergi ked an berhenti) yaa itu artinya, terus maksudnya apaaah??. Oke di jelasin, GotoAndStop maksudnya menuju ke sebuah keyframe (frame yang telah terisi animasi) dan animasi pada keyframe tersebut langsung berhenti kalau sudah selesai berjalan (engga diulang-ulang).

Gitu maksudnya, nah seperti di materi sebelumnya saya katakan mulai sekarang kita lebih banyak praktiknya ketimbang penjelasan. Oke kita check langsung aja bagaimana contoh penggunaan ActionScript *GotoAndStop* ini. **Check this out...**

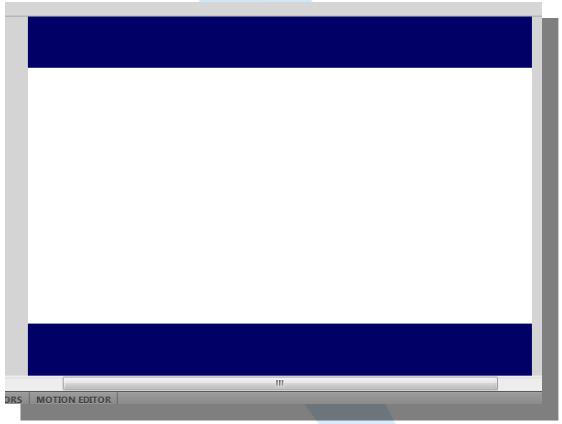
Langkah-langkah penggunaan GotoAndStop, **Like this yoo** :

1. Buat lembar kerja baru Adobe Flash, pilih ActionScript 2.0, **look at the picture**:

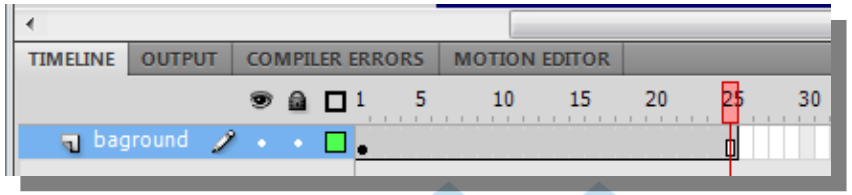
2. Klik 2x pada layer pertama, kita beri nama "**baground**", like this:



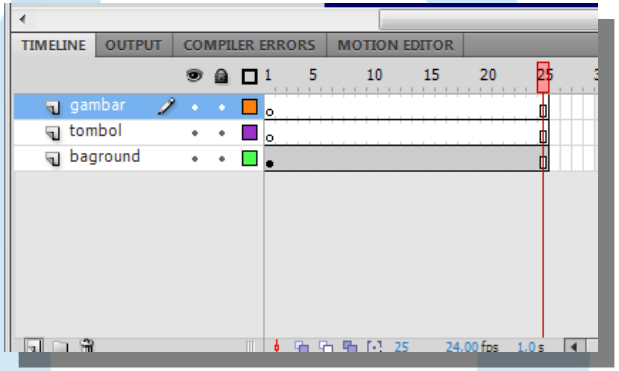
3. Buat objek kotak berwarna biru pada stage (lembar kerja) dengan menggunakan tombol **Rectangle Tool**  , Like this :



4. Klik kanan pada frame 25 kemudian pilih **Insert Frame** untuk memperpanjang framenyaa, like this :

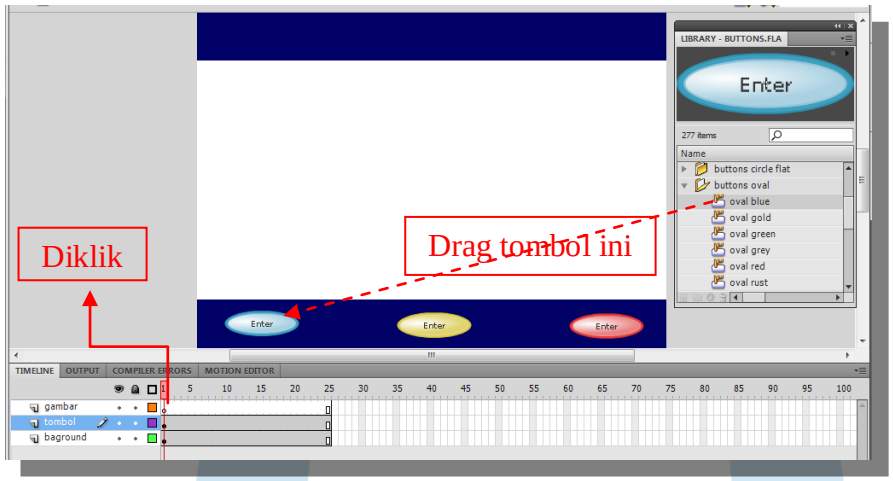


5. Tambahin 2 layer baru, masing-masing kasih nama "tombol" dan "gambar", tentunya dengan menekan tombol **New Layer** untuk menghasilkan layer baru tersebut, Like this :



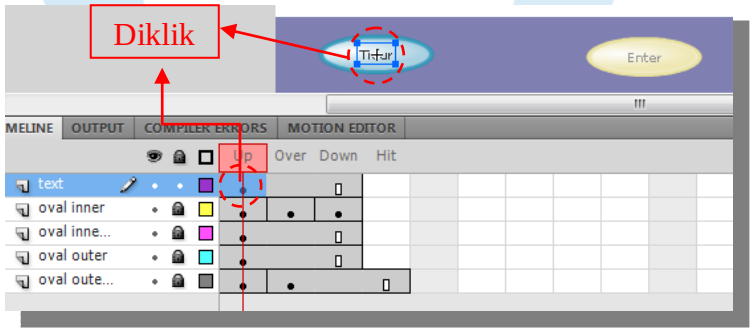
6. Terus klik menu **Window – Common Libraries – buttons** untuk nampilin panel **Library-buttons.**

7. klik frame 1 layer tombol, drag 3 tombol (button) dari panel **Library-buttons** tersebut ke stage, atur like this yo :



8. Next, klik 2x pada tombol paling kiri (tombol yang tadi dah dimasukin ke stage), buat di edit tulisannya.

9. Klik keyframe bagian **Up** di layer text, ganti tulisan "Enter" menjadi "Tidur" dengan menekan tombol **Text tool** T, Like this yoo :

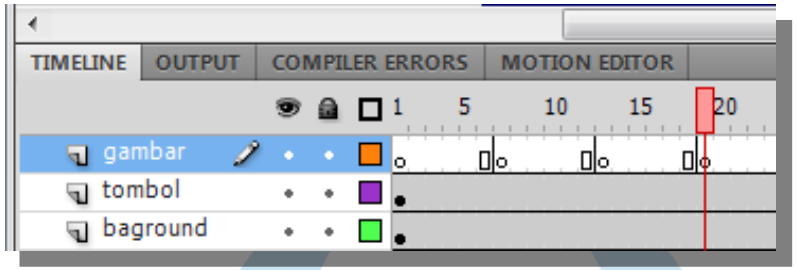


10. Terus tekan **Ctrl+E**, balik ke lembar kerja.

11. Pake cara yang sama ubah nama semua tombol menjadi **like this** :

12. Kemudian di frame 1 layer yang sama (layer **tombol**) tulis menggunakan **Teks tool** "Jangan Disebarkan" (atau apa saja,..suka-suka sobat, saya kasih tulisan ini agar jangan disebarkan contoh gambar yang saya pakai dalam source Code yang diberikan), letakan di posisi kotak biru bagian atas. **Like this** :

13. Sekarang ke layer **gambar**, klik kanan pada frame 7 kemudian pilih **Insert Keyframe**, untuk menambahkan keyframe baru. Kemudian dengan cara yang sama lakukan pada frame 13, dan 19. **Like this** :

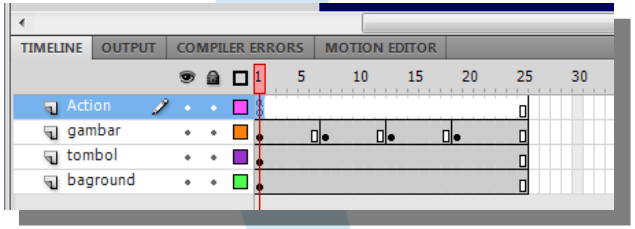


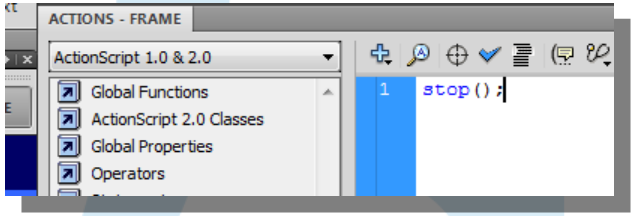
14. Klik frame 1 di layer **gambar**, terus tuliskan teks "Ini Contoh GotoAndStop" **like this** :

15. Kemudian klik frame 7 dan masukan (drag) sebuah gambar yang sebelumnya telah anda import ke panel Library, masih inget kan cara import gambar ke panel library, inget dong (klo ga inget silakan liat di materi-materi sebelumnya pasti ada), **like this** :

16. Kemudian klik frame 13 dan masukan gambar yang berbeda sesuai selera, lakukan juga pada frame 19 masih di layer yang sama yaitu layer **gambar**, **like this** :

17. Sekarang kita mulai memasukan ActionScript, buat layer baru dengan nama "action", klik frame 1 dilayer **action** kemudian tekan **F9** (membuka panel Action(AS)) dan tuliskan script stop ();. Ini berarti apabila animasi dijalankan (**Ctrl+Enter)** maka akan berhenti pada frame 1 pada setiap layer **like this**:





18. Nah untuk AS GotoAndStopnya kita masukan ke dalam setiap tombol pada layer **tombol**, kita mulai dari tombol pertama yaitu tombol "Tidur". Klik tombol tersebut lalu tekan **F9** (untuk membuka panel Action). Kemudian tuliskan ActionScrip sebagai berikut :

```
On (release) {
        gotoAndStop(7);
}
```

Maksud dari perintah tersebut adalah jika animasi dijalankan dengan **Ctrl+Enter** dan ditekan tombol "Tidur" maka animasi tersebut pindah menuju frame 7 dan kemudian berhenti.

19. Dengan cara yang sama lakukan pada tombol "Janji" dan "Watter" pada tanda kurung ("()") setelah script gotoAndStop masukan no. setiap frame yang telah kita masukan gambar tadi. Like this:

    Tombol "Janji" :

```
On (release) {
        gotoAndStop(13);
}
```

    Tombol "Watter" :

```
On (release) {
        gotoAndStop(19);
}
```

20. Tekan **Ctrl+Enter** untuk menjalankan animasinya. **Selesai**.

PR

Nah untuk PR kali ini, silahkan sobat Nyankod buat sebuah aplikasi menampilkan menu makanan dan masakan atau gallery photo (pilih salah satu) dengan menggunakan gotoAndStop seperti yang sudah kita praktekan tadi, tentunya dengan kreasi dan imajinasi sobat masing-masing. Gampang kan?? Gampang donk…

SILAKAN DICOBA DI KOMPI ATAU LAPI MASING-MASING,....CAYOOO!!

Penutupan

Referensi book:
Kupas Tuntas Adobe Flash CS5
Madcoms & Penerbit Andi

Nah segitu aja dulu materi dan praktiknya untuk edisi 7 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 malu-malu 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-8 sobat Nyankod.

[PHP]

Ahmad Oriza Sahputra
@oriza_sahputra
ahmadoriza@gmail.com
http://orizasahputra.blogspot.com

$artikel->setJudul('Objek');

Dalam edisi ini penulis mau bayar utang janji di nyankodMagz edisi ke-2, soal pembahasan tipe data. Itu.. itu.. yang bahas objek, masih inget kah??. Waktu itu penulis belum bisa nyampein materi tipe data objek sebab nyankoders belum cukup umur #ngacolagi.. hehe. Maskudnya belum cukup materinya. Berhubung nyankoders udah pada jago sekarang, materi : tipe data + operator + struktur kontrol + array + fungsi.. udah diluar kepala, sekarang saatnya kita pelajarin objek di PHP, buat nambah ilmu… :)

Sebelumnya penulis mohon izin buat bercakap-cakap dulu soal teori Pemrograman Berorientasi Objek, soalnya tipe data objek konsepnya dari situ. Pemrograman Berorientasi Objek (PBO) merupakan metode pemrograman yang berorientasikan kepada objek, semua data dan fungsi dibungkus dalam suatu kelas kelas atau objek.

Knapa harus belajar PBO om?? Gak pake objek juga bisa bikin program PHP, terus ada acara bungkus-bungkusan segala, bikin ribet aja .. eet dah emangnya mu buat nasi bungkus??

Emang sih mas bro.. (kok jadi mas bro?? xixi) gak pake PBO ma tipe data objeknya juga kita bisa bikin program PHP, sekedar sharing last project penulis juga dikerjain tanpa implementasi PBO 100%.. alias prosedural. Kembali lagi mas bro, itu semua tergantung kebutuhan, ada kelebihan dan kekurangannya. PHP emang bukan murni bahasa pemrograman yang mengharuskan developernya full pake konsep PBO, beda ma Java dan laen laen. Sebagai developer PHP yang bertanggung jawab, cerdas dan keren :) kita diharuskan ngerti konsep PBO, banyak proyek swasta yang mewajibkan developernya pake framework pihak ketiga, atau mungkin bikin

framework sendiri dalam pengerjaan aplikasi, yang notabene pake konsep PBO. Pustaka-pustaka yang beredar juga pake konsep PBO lho.. contoh.. pustaka generator berkas PDF dan Excel. Okeh.. sekalian aja nih penulis kasih tau kelebihan pake konsep PBO :

1. Membuat program lebih fleksibel
2. Mudah untuk mengubah program
3. Cocok diimplementasikan untuk program berskala besar
4. Program lebih mudah dikembangkan dan dirawat

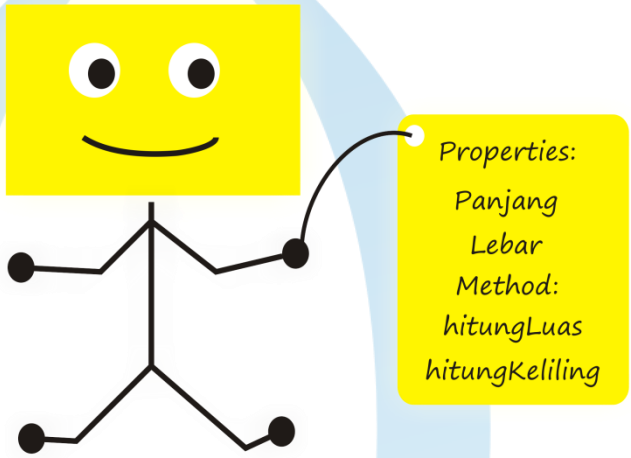
Sip.. Gimana nyankoders?? Mau pingin lanjut belajar PBO PHP gak?? *Maoooo*. yaud'ddah.. mari kita kemon!!!

Konsep Dasar PBO nih!!

Nyankoders harus paham dulu konsep dasar PBO, diantaranya **Encapsulation**, **Inheritance**, dan **Polymorphisme**. Kalo biasanya di pemrograman prosedural kita bikin data/variabel sama fungsi terpisah, sekarang di PBO kita satuin mas bro!! jadilah suatu kelas, itulah Encapsulation, dan objek adalah hasil representasinya. Data/variabel di dalem kelas dinamain **properties**, sedangkan fungsi di dalem kelas disebut **method**. Inheritance adalah metode pewarisan, dengan kata lain kelas yang mewarisi suatu kelas, semua sifat (properties dan method) kelas parent akan turun ke kelas child. Yang terakhir Polymorphisme merujuk pada sebuah bahasa pemrograman yang punya kemampuan buat proses objek secara berbeda tergantung pada tipe data atau kelasnya.

Gimana??? Ngerti kan mas bro?? *kitorang belum mangarti om!!* Aaaadooh.. beta su jelaskan kamorang belum mangarti.. kumaha??!!.

Yaud'ddah. Penulis kasih contoh.. kenal ma persegi panjang gak?? *Kenal…* nah, itu panjang, lebar, dll yang jadi atribut dia itulah properties, sedangkan yang jadi method contohnya fungsi perhitungan luas sama kelilingnya. Cekidot gambar ini deh :



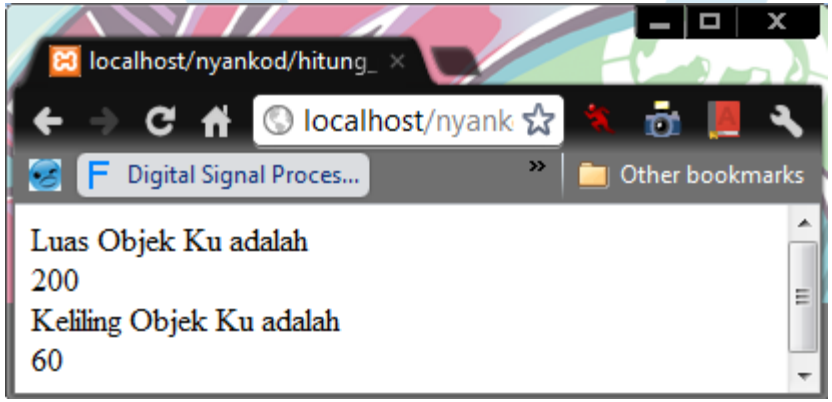
Yup.. sekarang kita coba bikin kelas sama objek perdana di PHP, kita bikin program persegi panjang, cekidot :

Program: hitung_persegi_panjang.php

```
1  <?
2  //bikin kelas sederhana, kelas persegi panjang
3  class Persegi_Panjang{
4     //ini propertiesnya mas bro
5     var $panjang = 10;
6     var $lebar = 20;
7
8     //ini methodnya mas bro
9     function tampil_luas(){
```

```
10          $luas = $this->panjang * $this->lebar;
11          echo $luas;
12       }
13
14       function tampil_keliling(){
15          $kel = (2 * $this->panjang) + (2 * $this->lebar);
16          echo $kel;
17       }
18    }
19    //sekarang kita buat objek baru dari kelas yg tadi dibikin
20    $objek_ku = new Persegi_Panjang();
21    //tampilkan hasil perhitungan dengan memanggil method
22    echo "Luas Objek Ku adalah <br>";
23    $objek_ku->tampil_luas();
24    echo "<br>";
25    echo "Keliling Objek Ku adalah <br>";
26    $objek_ku->tampil_keliling();
27    ?>
```

Output:



Gimana???. Pasti sekarang tambah banyak yang gak ngertinya ya??? xixixi ;D. Becanda mas bro. Barusan itu kita buat kelas persegi panjang sederhana yang kita udah tentuin nilai atributnya (properties), yaitu panjang dan lebarnya luk et baris 5 dan 6. Terus kita bikin dua method yang berfungsi menghitung sekaligus menampilkan luas dan keliling persegi panjang tadi. Method pertama di baris 9-12 untuk luas, lanjut di baris 14-17 untuk keliling. Dalem operasi di method, nyankoders liat this->panjang sama $this->lebar gak??, kalo liat berarti matanya sehat sehat aja. Lho???.. bukan itu. Fungsi kata $this untuk nunjuk variabel yg ada di dalem kelas, sedangkan kata $this->panjang, maskudnya nunjuk var $panjang, kalo pake $this nama var nya gak pake tanda (‘$’) lagi.

Coba perhatiin baris 20, ada ekspresi $objek_ku = new Persegi_Panjang(); . Disitulah kita bikin objek baru yang merupakan instansiasi kelas Persegi_Panjang. Kita bisa operasikan objek dengan method yang terbungkus di dalem kelas Persegi_panjang, contohnya di baris 23 dan 26.

Om.. kok jadi ribet banget si?? mau bikin program persegi panjang kaya gitu aja jadi panjang banget source codenya, padahal pake cara biasa cuma butuh 6 baris, sambil merem lagi!! Haduh.. nanya mulu nih. Sabar mas bro.. namanya juga belajar. Kita harus paham dulu konsep sama implementasi dasarnya, jadi pakenya ya program kaya gitu, biar mudah dimengerti. Kalo bikin kelas yang advance ntar puyeng lagi ;D (penulisnya yang puyeng kali.. hehe). Yaud'ddah.. lanjut nyoo..

Setter dan Getter

Nah.. sekarang mari kita pahamin konsep setter dan getter. Ini juga menu wajib yang harus diicip-icip nyankoders. Kalo tadi kita buat kelas yang statis sekarang kita buat dinamis. Pertama setter, setter adalah fungsi/method yang bermanfaat untuk memanipulasi nilai properties, that is it. Kedua getter, getter adalah keadaan dimana anggota tubuh kita terasa bergetar-getar.. sering terjadi di wilayah saku celana, saat

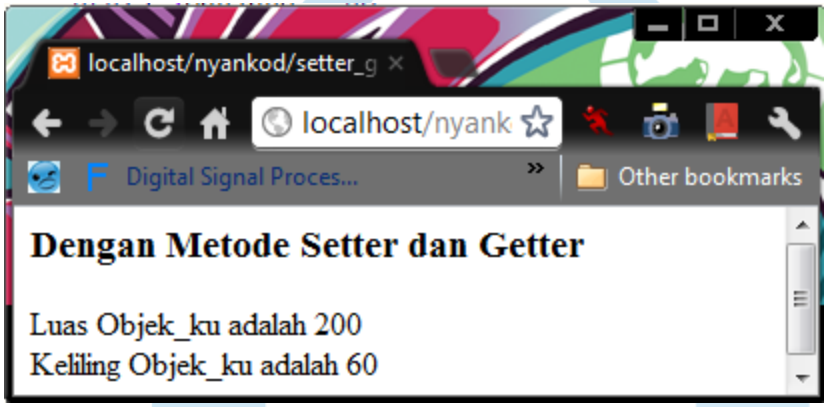
mengantongi hp dengan profil vibrate. xixi.. ;D. Halaah.. bukan itu, getter adalah fungsi/method yang bermanfaat untuk mengembalikan nilai hasil operasi. Cobain :

Program: setter_getter.php

```
1   <?
2   class Persegi_Panjang{
3      var $panjang;
4      var $lebar;
5
6      //ini method setter panjang
7      function setPanjang($p){
8         $this->panjang = $p;
9      }
10     //ini method setter lebar
11     function setLebar($l){
12        $this->lebar = $l;
13     }
14     //ini method getter luas
15     function getLuas(){
16        $luas= $this->panjang * $this->lebar;
17        return $luas;
18     }
19     //ini method getter keliling
20     function getKel(){
21        $kel = (2 * $this->panjang) +  (2 * $this->lebar);
22        return $kel;
23     }
24  }
25  $objek_ku = new Persegi_Panjang();
26  $objek_ku->setPanjang(20);
27  $objek_ku->setLebar(10);
28  echo "<h3>Dengan Metode Setter dan Getter</h3>";
29  echo "Luas Objek_ku adalah ".$objek_ku->getLuas()."<br>";
30  echo "Keliling Objek_ku adalah ".$objek_ku->getKel()."<br>";
```

```
31  ?>
32
```

Output:



Sip.. Keliatan begete kan perbedaan cara penggunaan kelas dibanding program sebelumnya. Dalam kelas diatas kita membuat properties tanpa ditentukan nilainya. Dalam membuat method setter kita dibantu dengan satu argumen/parameter fungsi yang menyimpan nilai untuk kemudian dijadikan nilai properties, luk et baris 8 dan 12. Sedangkan dalam membuat method getter, kita bikin fungsi seperti biasa dan mengoperasikan nilai properties sesuai rumus, yang membedakannya adalah menambahkan **return** untuk mengembalikan nilai hasil operasinya. Pemanggilan setter ada di baris 26 dan 27, sedangkan getter ada di baris 29 dan 30.

Sebelumnya mohon maaf nih nyankoders, artikel edisi ini dikit banget. Inheritance, Polymorphisme, Konstruktor, Destruktor dan materi lain belum sempet disampein. Tapi di edisi selanjutnya bakal penulis sampein lebih lengkap soal PBO di PHP. Maapin kalo ada kesalahan dalam penyampaiannya, maaf juga kalo materinya kurang lengkap.

Referensi :

- Utomo, Eko Priyo. 2008. 125 Tips Menguasai Bahasa Pemrograman PHP. Yrama Widya: Bandung
- Peranginangin, Kasiman. Aplikasi Web dengan PHP dan MySQL. Andi: Yogyakarta
- Komunitas Open Source Indonesia. 2008. Bahasa Pemrograman Open Source. Detiknas: Jakarta
- http://www.wikipedia.com

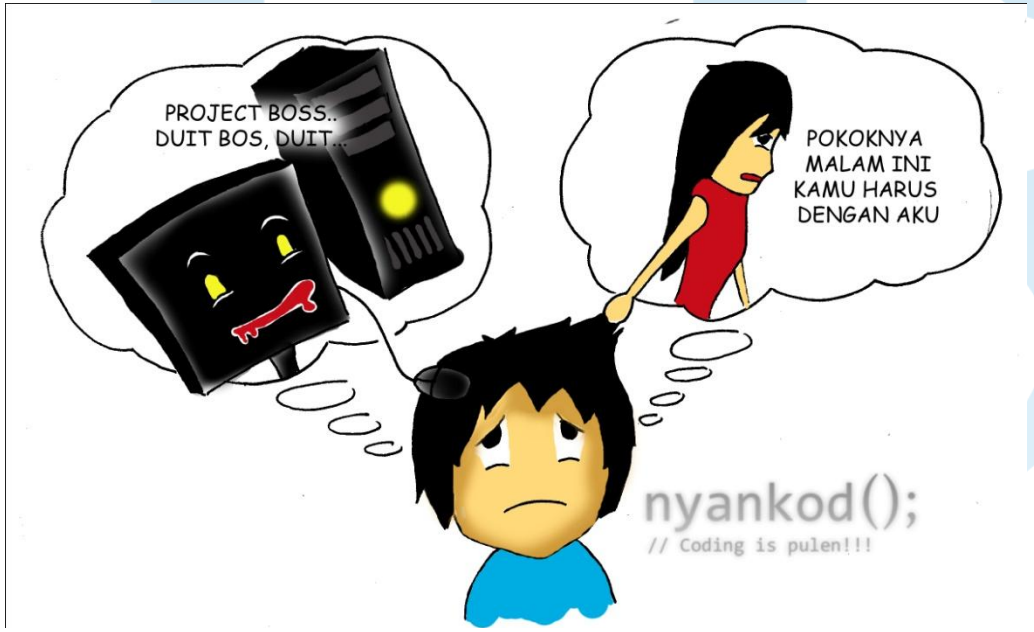
Saat-saat Dilema Programer

Yudha Pangesti D. Syailendra
@duysyailend
schaden47@ymail.com

Programer juga manusia cong, akan ada saatnya dimana programer juga merasakan dilema yang mendalam. Nih, Nyankod punya ilustrasi saat-saat dimana programer galau, ketika harus memilih sesuatu yang menentukan hidup dan mati.

Project atau Malem Mingguan

Kamu lagi ngerjain project dan udah mendekati deadline, tapi malam ini adalah malam minggu. Pilih project pacar marah-marah, pilih malam mingguan, project nggak kelar-kelar. Dah mepet nih jou... dilema tingkat nasional.



Skripsi atau Project

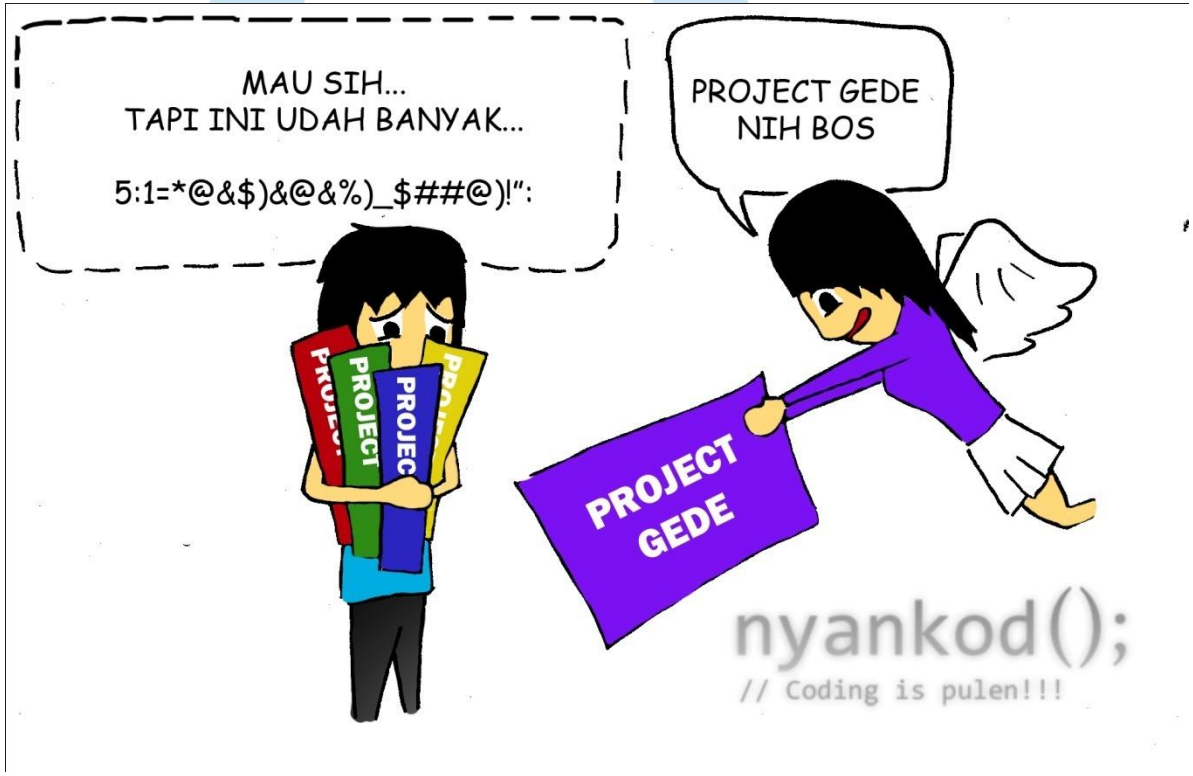
Skripsi adalah tantangan terakhir kuliah, itu sangat penting. Tapi makan juga penting. Jadi harus pilih mana nih?? Project atau skripsi..... Apalagi kalo projectnya gede, ini bakalan jadi dilema yang luar biasa, seperti dilema yang dihadapi spiderman.

Cewek Baru VS Laptop Baru??

Kalau kamu disuruh milih, kamu bakal milih mana nih, antara pacar baru atau laptop baru?? #jawabJujur

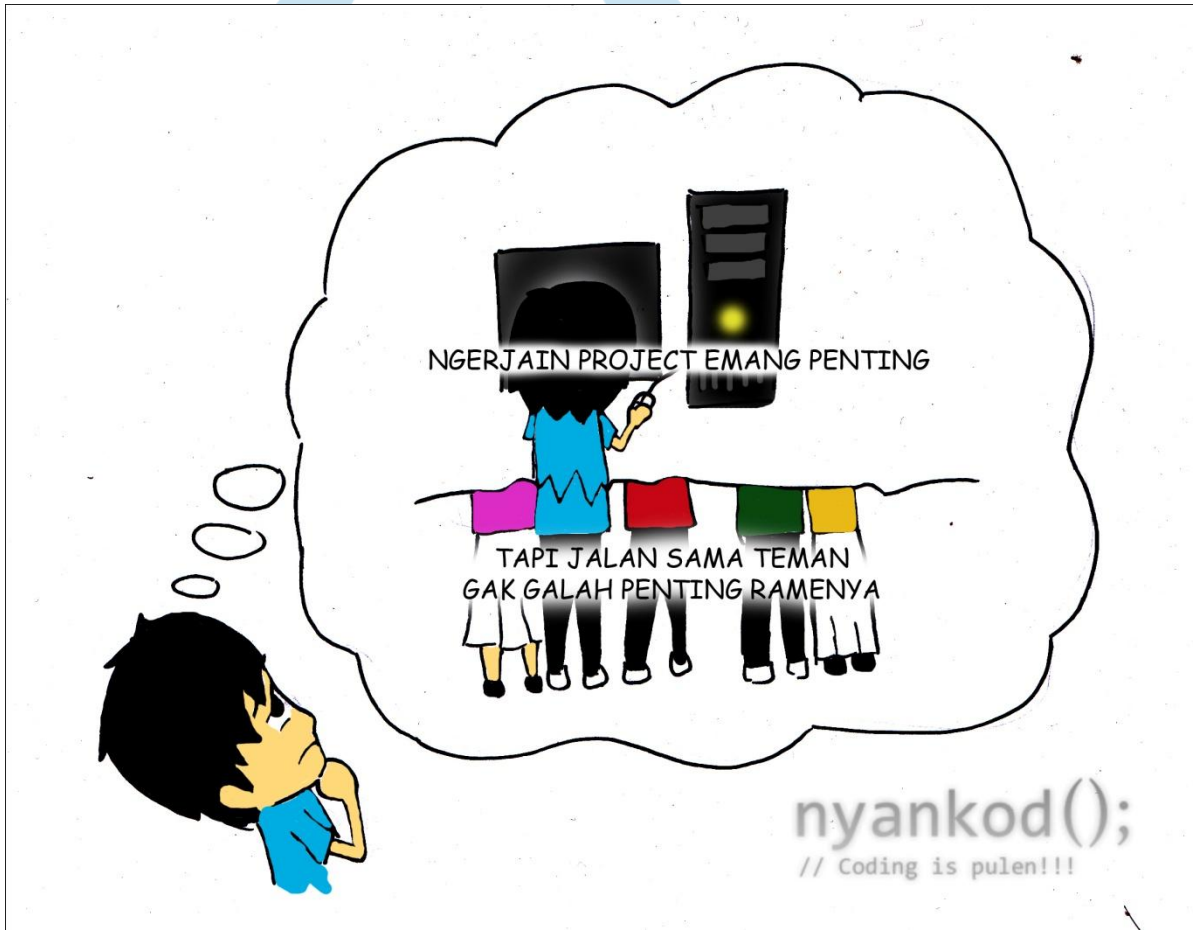
Kebanyakan Project

Sebagai manusia biasa, kadang-kadang super hero juga akhirnya melakukan kerja sambilan dengan menerima beberapa project sekaligus. Dan disaat ngerjain project-project itu, kadang-kadang ada tawaran project baru, dan nggak jarang juga project baru itu ternyata lebih gede dari project-project sebelumnya. Ditolak sayang, nggak ditolak pasti rusuh. -___-



Diajakin Maen Sama Temen

Pahlawan super juga kadang-kadang butuh refreshing nih. Tiap hari kerja di depan laptop, kadang-kadang bikin stress juga, tapi masalahnya adalah deadline. Deadline oh deadline.... gimana dong????



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

Edisi ke-8 akan terbit: Rabu, 13 Juni 2012