

Nyankod Magz

Love it and Earn It

Type Data

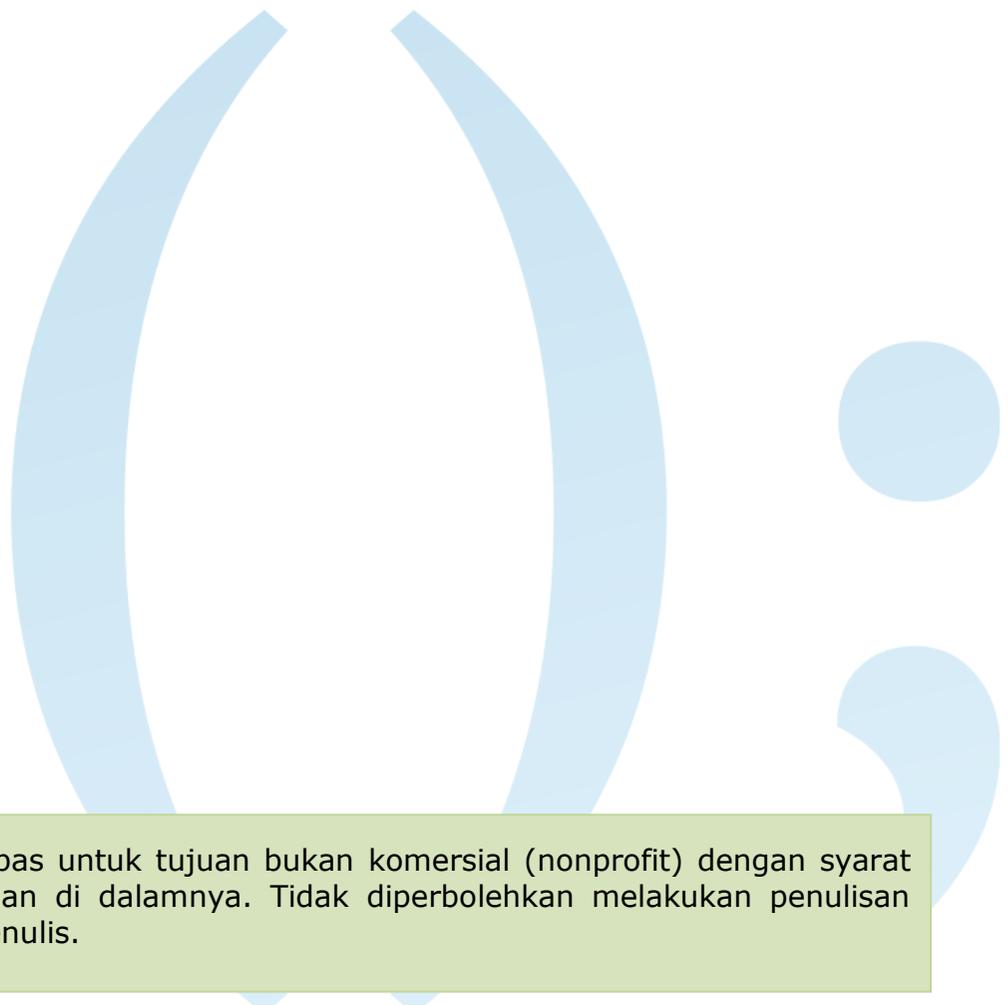


Ruby
Perl
ActionScript
Bash
JavaScript
PHP
Python
C



Nyankod Magz Edisi 2 Tipe Data

Love it and Earn It



Nyankod Magz ini dapat Anda sebarkan secara bebas untuk tujuan bukan komersial (nonprofit) dengan syarat tidak mengurangi dan menambahkan isi kandungan di dalamnya. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapat izin terlebih dahulu dari penulis.

Sapa Nyankod

Ada ungkapan yang menarik, "*Mempertahankan itu jauh lebih sulit dari pada membangun*". Sobat Nyakoders, percayakah Anda dengan ungkapan itu? Bila Anda percaya berarti kita sepemikiran dalam hal itu. Karena inilah yang saat ini sedang kami rasakan dan alami. Baru 2 minggu yang lalu kami memulai untuk menerbitkan Nyankod Magz edisi pertama, dan kini kami diuji untuk bisa mempertahankan komitmen kami pada edisi kedua bahkan edisi-edisi selanjutnya. Sungguh bukan perkara mudah ternyata, tapi syukurlah sampai saat ini komitmen kami untuk ini masih bisa terjaga. So, edisi kedua ini kami bisa terbitkan kembali. Doakan kami Sobat Nyankoders, semoga ini akan terus berlanjut, agar kedepannya semakin banyak lahir programmer-programer di Indonesia.

Sobat Nyankoders, gimana dengan edisi pertama kami? Sudahkah Anda membacanya? Gimana tanggapannya? Atau buat yang belum sempat baca, silahkan baca terlebih dahulu. Anda bisa mendownloadnya secara gratis kok di homepage (<http://nyankod.blogspot.com>) kami. Buat yang udah baca, gimana nih tanggapannya? Semoga bermanfaat ya... dan bila ada beberapa kekurangan di Nyankod Magz di edisi manapun, jangan ragu-ragu ya untuk memberi tahu kami ke email kami (nyankod@gmail.com). Juga bila Nyankoders sekalian punya ide, saran ataupun masukan, juga jangan ragu-ragu untuk memberitahu kami. Kami akan berterima kasih sekali intuk itu.

Di edisi kedua ini kami mencoba untuk membahas lebih dalam dari bahasa pemrograman yang di edisi pertama telah dikenalkan. Tema utamanya adalah tentang tipe data. Sebenarnya kami memberikan kebebasan kepada para Nyankodist untuk mengalirkan pembahasan, jadi mungkin ke depannya ada kemungkinan pembahasan antara bahasa pemrograman yang satu dengan yang lainnya akan ada perbedaan. Itu tergantung bagaimana penulisnya masing-masing.

Oh ya Sobat Nyankoders, pernah nggak terpikirkan oleh Anda hal ini, suatu saat di tahun 2025, saat itu Indonesia adalah negara yang benar-benar maju, terutama dalam bidang Teknologi Informasi. Laptop dan komputer benar-benar telah ada di setiap rumah warga. Anak-anak telah dikenalkan pemrograman sejak kecil, bahkan lomba membuat program telah diadakan oleh pemerintah untuk anak SD. Perusahaan perangkat komputer telah menjamur dimana-mana baik itu perangkat keras maupun perangkat lunak. Berdasarkan survei, pekerjaan terbanyak masyarakat waktu itu adalah menjadi programmer. Software-software dari Indonesia telah digunakan hampir diseluruh penjuru dunia. Bahkan Operating System yang digunakan oleh masyarakat Indonesia saat itu benar-benar murni buatan Indonesia, dan itu juga digunakan oleh berbagai masyarakat di penjuru dunia, karena lebih user friendly dan lebih powerful dibanding produk Microsoft. Jangankan software-software semisal pemutar musik dan video, software-software games dengan tingkat 3D yang sangat detail pun telah menjamur, dan itu asli buatan Indonesia. Masyarakat saat itu tidak lagi dipusingkan dengan bagaimana cara melamar kerja yang baik dan bagaimana cara supaya pemerintah menurunkan harga beras dan minyak, bahkan tidak dipusingkan dengan mengapa Jakarta selalu banjir saat musim hujan, tapi masyarakat saat itu dipusingkan karena mencari ide untuk berbagai penemuan-penemuan baru, yang akan berguna untuk masyarakat dan negara. Semua rakyat Indonesia sibuk dan berlomba-lomba untuk berkarya. Indonesia tidak lagi menjadi negara korupsi, tetapi menjadi negara tempat para programmer dan hacker. Karena budaya pemrograman telah benar-benar menjadi atmosfer di Indonesia saat itu. Hahahaha.... Keren kan Indonesia saat itu???

Bagaimana sudahkah melamunkan hal itu?? Kami selalu memimpikan hal itu bisa terjadi. Semoga saat-saat seperti itu bisa kita rasakan. Kita hanya bisa berusaha dan berdoa supaya itu bisa terwujud.

Oke semoga ini bisa bermanfaat buat Nyankoders sekalian. Mohon maaf bila masih banyak kekurangan. Jangan lupa ya buat berlangganan Nyankod Magz, gratis kok, caranya bisa dibaca di bagian akhir Nyankod Magz ini.

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:
nyankod@gmail.com.

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

Di dalam sini, ada..

	Perl - [Tipe Data – Skalar]	Halaman 9
DASAR-DASAR PHP – IDENTIFIER, VARIABEL, DAN TIPE DATA		Halaman 20
Hello #{RubyWorld} – Bagian Number dan Operator		Halaman 31
	ActionScript {Variabel}	Halaman 39
	Tipe Data di Bahasa C	Halaman 47
	Bash Programming - Basic	Halaman 51
	JavaScript Object	Halaman 58

[Perl]



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



[Tipe Data – Skalar]

Hi Nyankoders, gimana nih dengan ritual kenalannya sama Perl?? Berjalan lancarkah?? Udah sejauh mana sekarang hubungannya?

Oke, anggap aja udah lancar ya kenalannya. Buat yang baru gabung, silahkan baca artikel "Kenalan dengan Perl" di edisi sebelumnya (Edisi 1: Nyankod Say "Hello World").

Well, setelah kenalan, sekarang kita akan coba supaya bisa lebih dekat lagi dengan Perl, supaya kita bisa tau lebih dalam tentang Perl dan bisa menjiwai dengan sepenuh hati hubungan ini (ahahahay), istilahnya mungkin PDKT-an.

Sekedar review pembahasan, sebelumnya kita udah bahas tentang asal usul Perl, mulai dari latar belakang orang tuanya (maksudnya orang yang pertama kali membuatnya), fungsi dan kegunaannya, mottonya yang aneh, berbagai perlengkapan perang buat menghadapinya, cara mengeksekusi kode program, bahkan sampai pada lelucon nama PERL dan mitos hacker. Lumayan seru juga kan? Sebuah bahasa pemrograman yang unik.

So, apa yang bakal kita bahas sekarang?

Oke, sila tengok judul artikelnya! (Malaysiaan ceritanya)

Tipe Data – Skalar?

Ya bener banget, kita bakal coba bahas tentang Tipe Data – Skalar.

Maksudnya?? (garuk-garuk kepala)

Aizz, ya maksudnya kita bakal bahas tentang Tipe Data, terutama Tipe Data Skalar...

Ouh... kirain.

Oh ya Penulis, bapak Kamu guru Matematika ya?

Lho, kok tau??

Karena kamu telah membingungkan hatiku?? Hahahahay.....

Gubrag!!!! Kamu bisa aja deh... Hatiku meleleh nih!! Tapi ada yang aneh, hati emangnya bisa bingung ya?? Ah biarina aja... hehehehe.

Baiklah, mari kita mulai aja pembahasanya.

🌸~ Cemungud yuah kakak....!!!

Aizzz.... Yawdah... Kita mulai ya...

Esensi dari sebuah pemrograman adalah komputasi dan otomatisasi. Kita ingin komputer kita melakukan sesuatu setelah kita menginput data, kemudian komputer melakukan proses dan menampilkan hasilnya di layar. Amat jarang kita menulis sebuah program yang hasilnya jelas-jelas kita sudah mengetahui keluarannya. Dan jauh lebih jarang lagi kita menulis program yang hasilnya tidak melakukan apa-apa. So, jika kita ingin menulis program yang tidak hanya sekedar menuliskan kata "Hello", kita harus mengetahui berbagai karakteristik dan pola di dalam pemrograman tersebut. Nah, salah satu hal yang sangat penting buat kita ketahui adalah tipe data.

Perl memiliki 3 jenis tipe data, yaitu **skalar**, **array** dan **hash**. Skalar menyimpan nilai tunggal, array menyimpan kumpulan skalar dengan urutan tertentu, sedangkan hash menyimpan kumpulan skalar berpasangan. Pembahasan tentang array dan hash insya Allah akan saya coba bahas di edisi-edisi selanjutnya, buat sekarang kita fokus dulu di skalar.

Skalar merupakan tipe data yang sangat fundamental di Perl, karena merupakan basic dari unit data. Masih inget tentang skalar dalam matematika atau fisika? Ya kurang lebih seperti itulah. Intinya adalah tipe data skalar merupakan data dengan nilai tunggal dan tidak dapat berubah. Maksudnya tidak dapat berubah adalah 5 adalah 5, 5 bukanlah 4 atau 6. Sampai kapan pun 5 adalah 5. Skalar dalam Perl terdiri dari dua jenis yaitu Numbers dan Strings.

Numbers

Number = angka. Tipe data angka dalam Perl terbagi menjadi dua yaitu integer dan float. Integer merupakan bilangan bulat, seperti 2, 100, -6, 0 dan 143. Sedangkan float adalah bilangan real, seperti 1.5, 45.9, -43.8 dan 1.1.

Dalam penulisan angka di Indonesia, biasanya kita menuliskan angka dengan format ini 1.234.000,56. Titik merupakan pemisah ribuan, sedangkan koma untuk pecahan desimalnya. Dalam Perl formatnya agak sedikit berbeda. Untuk mempermudah kita bisa menggunakan underscore (`_`) sebagai pembatas angka ribuan. Sebenarnya underscore ini bisa saja diabaikan, artinya Nyankoders boleh saja tidak menggunakannya (ex: 10000000), namun tentu akan lebih sulit dalam beberapa kondisi. Adapun untuk pecahan desimal Perl menggunakan titik (`.`).

Untuk lebih jelasnya cobalah tuliskan program ini:

Program 2: skalar_numbers.pl

```
1  #!/usr/bin/perl
2  # nama program : skalar_numbers.pl
3
4  print 1, "\n";           # integer
5  print 1000, "\n";       # integer ribuan
6  print 123456, "\n";     # integer dengan banyak angka
7  print 1_000, "\n";      # dengan pembatas underscore
8  print 1_234_567, "\n";  # jutaan dengan pembatas
9  print 34.67, "\n";      # float
10 print 1_234_567.89, "\n"; # float dengan pembatas
```

Cobalah eksekusi program tersebut. Maka keluarannya kurang lebih akan seperti berikut:

```
$ perl skalar_numbers.pl
1
1000
123456
1000
1234567
34.67
1234567.89
$
```

Selain bisa menuliskan angka dengan format bilangan desimal (bilangan 10-an, yaitu 0, 1, 2, 3, 4, 5, 6, 7, 8, 9), dalam Perl juga kita bisa menuliskan dengan format lain seperti Oktal (bilangan 8-an, yaitu 0, 1, 2, 3, 4, 5, 6, 7), Heksadesimal (bilangan 16-an, yaitu 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) dan Biner (bilangan 2-an, yaitu 0 dan 1).

Untuk menuliskan bilangan oktal, maka harus diawali dengan 0 (ex: 0327). Untuk bilangan heksa, maka harus diawali dengan 0x (ex: 0xfa). Dan untuk bilangan biner harus diawali dengan 0b (ex: 0b1010001). Untuk lebih jelasnya silahkan tulis program berikut ini:

Program 3: format_bilangan.pl

```
1  #!/usr/bin/perl
2  # nama program : format_bilangan.pl
3
4  print 255,          "\n"; # bilangan desimal
5  print 0377,        "\n"; # bilangan oktal
6  print 0xFF,        "\n"; # bilangan heksadesimal
7  print 0b11111111, "\n"; # bilangan biner
```

Jalankan program tersebut seperti biasa.

```
$ perl format_bilangan.pl
255
255
255
255
$
```

Yups, semuanya akan menunjukkan angka 255. Bilangan desimal dari bilangan oktal 377 adalah 255. Bilangan desimal dari heksadesimal FF adalah 255. Dan bilangan desimal dari biner 11111111 adalah 255. Bagi Nyankoders yang masih kebingungan tentang konversi dari bilangan oktal, heksadesimal dan biner ke bilangan desimal silahkan baca-baca dulu artikel tentang ini di tempat lain. Hehehe... karena terlalu panjang bila harus membahas ini juga.

Strings

Satu jenis data lainnya dalam skalar adalah string. Sulit mendefinisikan string secara harfiah, intinya string adalah sebuah karakter, tulisan, huruf, angka dan lain sebagainya, yang cara penulisannya harus diapit oleh tanda petik, baik itu petik tunggal atau ketik ganda. Di edisi pertama kita telah membuat program sederhana buat menampilkan kalimat "Hello World, Nyankoder lagi belajar Perl nih!!!". Nah itu merupakan salah satu contoh string.

Dalam penulisan string terdapat beberapa perbedaan antara tanda kutip ganda dengan tanda kutip tunggal. Biar lebih mudah dicerna, perbedaan antara keduanya saya akan coba jelaskan dengan contoh. Silahkan tulis kode program berikut:

Program 4: kutip_ganda_vs_tunggal.pl

```
1  #!/usr/bin/perl
2  # nama program : kutip_ganda_vs_tunggal.pl
3
4  $saya = "nyankoder";
5
6  print "aku adalah $saya \n"; # kutip ganda
7  print `aku adalah $saya \n`; # kutip tunggal
```

Silahkan jalankan program tersebut.

```
$ perl kutip_ganda_vs_tunggal.pl
aku adalah nyankoder
aku adalah $saya \n
$
```

Oh ya, sebelumnya \$saya disebut variabel, yaitu sebuah tempat untuk menyimpan nilai. Untuk sementara diterima aja dulu ya, buat lebih jelas tentang variabel nanti akan saya jelaskan setelah ini.

Perbedaan yang tampak bila program tersebut dijalankan adalah, pada kutip ganda nilai variabel dapat terbaca, dan \n bisa bekerja dengan baik. \n merupakan spesial karakter untuk pindah baris. Sedangkan pada kutip tunggal, variabel \$saya ditulis apa adanya, dan tidak dikenal sebagai sebuah variabel, demikian juga karakter \n. tidak dianggap sebagai karakter untuk pindah baris (hanya ditulis apa adanya).

Dalam beberapa keadaan tanda petik ganda lebih enak digunakan, namun dalam keadaan lainnya berlaku kebalikannya. Seiring berjalannya waktu saya jamin, Nyankoders sendiri akan bisa menentukan kapan saatnya menggunakan petik ganda atau petik tunggal.

Bro, di atas kita lihat kan ada karakter \n yang berfungsi buat pindah baris. Nah sebenarnya ada beberapa karakter lainnya yang sejenisnya. Berikut ane tampilin deh:

Escape Sequence	Arti
\t	Tab
\n	Pindah baris
\r	Kembali ke awal baris
\b	Menghapus satu karakter sebelumnya
\a	Alarm

Nyankoders bisa bereksperimen sendiri dengan karakter-karakter di atas. Oh ya, mereka berlaku hanya di petik ganda saja. Sedangkan bila di petik tunggal mereka hanya dibaca sebagai karakter biasa.

Oke, saatnya mencoba yang satu ini bray....

Program 5: backslash_petik.pl

```
1  #!/usr/bin/perl
2  # nama program : backslash_petik.pl
3
4  print "\\n";
5  print "C:\WINDOWS\Media\ \n";
6  print "C:\\WINDOWS\\Media\\ \n";
7  print `C:\WINDOWS\Media\ `, "\n";
```

```
$ perl backslash_petik.pl
\nC:WINDOWSMedia
C:\WINDOWS\Media\
C:\WINDOWS\Media\
$
```

Jadi gini, karakter backslash (\) akan membekukan fungsi karakter-karakter khusus. Sehingga karakter khusus yang ada di depan tanda backslash akan ditampilkan apa adanya.

Variabel

Sebelumnya udah sempat saya bahas sedikit, variabel adalah sebuah tempat untuk penyimpanan data skalar (sebenarnya bisa juga buat nyimpen data dengan tipe lainnya). Nilai variabel bisa kita rubah kapan pun kita mau.

Di beberapa bahasa pemrograman, biasanya untuk menciptakan sebuah variabel kita terlebih dahulu harus mendeklarasikan terlebih dahulu tipe data variabel tersebut. Namun untuk Perl, itu tidak berlaku. Kita tidak perlu mendeklarasikannya terlebih dahulu, ya karena memang skalar hanya ada dua, yaitu numbers dan strings. Bahkan di Perl, tipe data bisa otomatis dikonvert ke tipe data yang lainnya bila memang diperlukan.

Oke dari pada bingung-bingung, mendingan kita langsung pelajari dari contoh aja deh. So, silahkan coba program berikut:

Program 6: mulai_variabel.pl

```
1  #!/usr/bin/perl
2  # nama program : mulai_variabel.pl
3
4  $x = 'nyankod';
5  $y = 2_012;
6  $z = $x;
7  $x = 'nyankoder';
8
9  print $x, "\n";
10 print $y, "\n";
11 print $z, "\n";
12 print 'Saya adalah seorang ', $x, "\n";
13 print "Selamat tahun baru $y \n";
```

Bila dijalankan, maka tampilannya adalah seperti berikut

```
$ perl mulai_variabel.pl
nyankoder
2012
nyankod
Saya adalah seorang nyankoder
```

Oke penjelasannya gini cuy. Variabel \$x pertama kita beri nilai 'nyankod', \$y diisi dengan 2_012 (maksudnya 2012, udah dibahas sebelumnya), \$z kita isi dengan \$x. Maksudnya adalah variabel \$z diisi dengan nilai dari variabel \$x. Atau dengan kata lain \$z diisi dengan 'nyankod'. Selanjutnya variabel \$x kita ubah nilainya dengan 'nyankoder'. Jadi, ketika kita tampilkan di layar dengan perintah print untuk variabel \$x, maka yang akan tampil di layar bukan nyankod, tetapi nyankoder. Karena nilai \$x telah kita rubah sebelumnya. Selanjutnya print \$y tentu akan menghasilkan 2012. Dan \$z akan berisi nyankod. Nah coba perhatikan baris ke 12 dan 13. Kita pun bisa menuliskan langsung string pada print dan kita juga bisa langsung menampilkan variabel di layar. Karena pada baris 12 kita menggunakan petik tunggal, dan petik tunggal tidak bisa membaca variabel, jadi kita memisahkan string dengan koma (,). Untuk baris 13, karena kita menggunakan petik ganda, jadi kita bisa menyimpan variabel di dalam string.

Mungkin pembahasan edisi kedua kita sampe di sini dulu. Di edisi selanjutnya saya punya pembahasan yang jauh lebih menarik dari ini, silahkan tunggu 2 minggu lagi. Semoga bermanfaat, tetap semangat buat belajar Nyankoders...

Coding is pulen bro!!!

[PHP]



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



DASAR-DASAR PHP – IDENTIFIER, VARIABEL, DAN TIPE DATA

Pengertian Identifier

Dalam artikel pada edisi kedua, penulis ingin menyampaikan beberapa pengetahuan tentang dasar-dasar PHP. Dimulai dari pengertian Identifier sampai kepada contoh penerapan tipe data dalam pemrograman PHP. Penulis harap nyankoders memahami artikel pada edisi ini, sebab ini merupakan pondasi untuk membuat program-program dalam PHP :D. Okeh.. langsung saja ke bahasan pertama yaitu identifier. Identifier adalah suatu nama yang dibuat programmer untuk variabel, fungsi, atau kelas (class). Adapun aturan penamaan identifier dalam bahasa PHP sebagai berikut :

- Posisi awalan harus huruf atau underscore (_)
- Posisi berikutnya bisa huruf atau underscore lagi, bukan spasi
- Sifat identifiernya Case Sensitive artinya huruf capital dan huruf kecil dibedakan, atau memiliki arti yang berbeda
- Dilarang menggunakan tanda baca

Contoh : `$box width, @update()` <- salah
 `$box_width, update()` <- benar

Oya nyankoders.. di dalam PHP khusus untuk penamaan variabel harus dimulai dengan tanda \$, unik ya. Ini dia salah satu perbedaan syntax bahasa PHP dengan bahasa pemrograman lain. Sip.. pastinya nyankoders sudah paham gambarannya kan??.. kalo begitu mari kita pahami variabel lebih jauh..

Definisi Variabel dan Jenisnya

Variabel merupakan suatu lokasi dalam memori komputer yang dapat menyimpan suatu nilai. Variabel bersifat dinamis, dapat berubah kapan saja. Yup.. yang kita tulis dalam source code sebenarnya akan diolah dalam memori. Nah.. PHP sendiri merupakan suatu interpreter. Beliau tidak menghasilkan execution file karena pada umumnya dipakai untuk menciptakan aplikasi web based. Ketika file php di-request oleh browser client lewat url baik dalam lingkup LAN atau Internet, interpreter PHP di dalam server kemudian memproses source code yang dipanggil, mengubahnya kedalam bahasa rakitan, supaya dimengerti oleh mesin komputer server, setelah itu mengirimkan kembali hasilnya ke layar monitor client. Kembali lagi ke variabel.. :D, berikut ini jenis variabel yang lazimnya digunakan dalam PHP:

- Variabel Global: variabel yang bersifat global dan bisa dipakai di seluruh bagian script. Baik itu function atau prosedur. Biasanya variabel global dideklarasikan di awal. Contoh:

```
1 <html>
2 <head><title>Belajar PHP</title></head>
3 <body>
4   <h3>Variabel Global</h3>
5   <?php
6     $a = 12; //variabel global
7
8     function tampil(){
9       global $a; //mengakses var global
```

```
10     echo $a;
11     }
12     echo tampil();
13     ?>
14 </body>
15 </html>
```

- **Variabel Lokal:** variabel yang dideklarasikan dan diinisialisasikan di dalam suatu fungsi saja, sehingga hanya dapat digunakan di dalam fungsi tersebut. Contoh:

```
1 <html>
2 <head><title>Belajar PHP</title></head>
3 <body>
4 <h3>Variabel Lokal</h3>
5 <?php
6     $a = 12; //variabel global
7
8     function tampil(){
9         $b = 20; //variabel lokal
10        echo $b;
11    }
12    echo tampil();
13 ?>
14 </body>
15 </html>
```

Penggunaan Variabel dengan Macam-macam Tipe Data

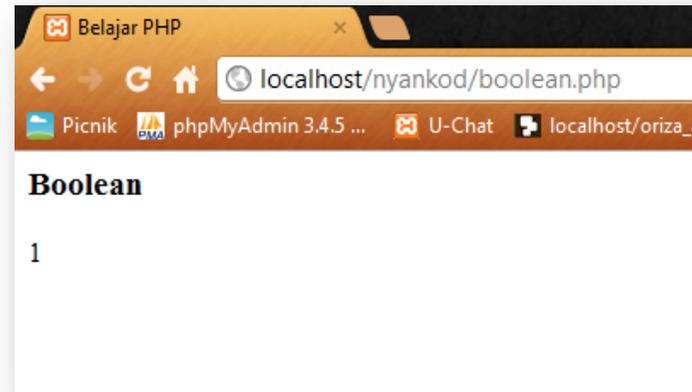
Pemrograman PHP mempunyai banyak sekali tipe data. Masing-masing tipe data dapat digunakan sesuai dengan kebutuhan program yang akan kita buat. Nah.. yang menarik dalam PHP kita tidak selalu harus mendeklarasikan penggunaan variabel dan tipe datanya diawal source code program seperti pada umumnya. Bahkan tanpa kita deklarasikan dengan catatan kita langsung memberikan nilai awal (inisialisasi) di tengah source code program sama sekali dibolehkan, dan hal demikian tidak akan menyebabkan galat. Berikut ini penjelasan masing-masing tipe data beserta contohnya (semua source code edisi kedua bisa di download di <http://www.mediafire.com/?e22k22khcs97100>):

1. Boolean

Yaitu tipe data yang hanya memiliki nilai TRUE atau FALSE yang bersifat case sensitive, tipe data ini digunakan untuk menyatakan suatu nilai kebenaran.

```
1 <html>
2 <head><title>Belajar PHP</title></head>
3 <body>
4   <h3>Boolean</h3>
5   <?php
6     //inisialisasi
7     $x = TRUE;
8     $y = FALSE;
9     //operasi
10    echo $x = $x."<br>";
11    echo $y = $y;
12    /*output program hanya akan menghasilkan nilai 1, sebab
    jika benar ketemu benar jawabannya adalah benar, jika salah
    ketemu salah jawabannya salah, dan tidak menghasilkan apapun*/
13    ?>
14 </body>
15 </html>
```

Output:



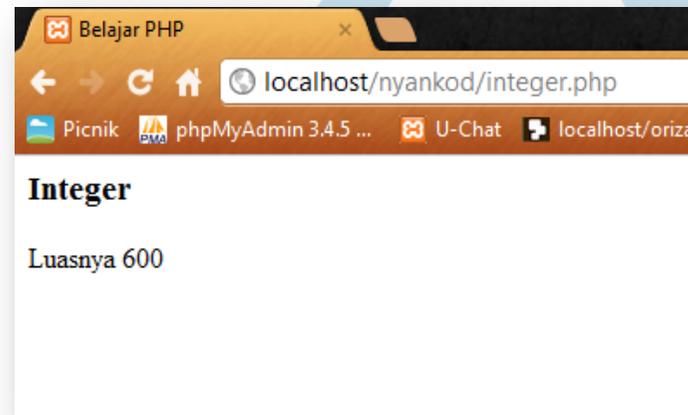
2. Integer

Yaitu tipe data yang dapat menyimpan nilai bilangan bulat. Bilangan yang dapat disimpan mencapai angka 2 milyar. Tipe data yang satu ini sering sekali digunakan dalam program yang berhubungan dengan perhitungan angka.

```
1 <html>
2 <head><title>Belajar PHP</title></head>
3 <body>
4   <h3>Integer</h3>
5   <?php
6     //inisialisasi
7     $panjang = 20;
8     $lebar = 30;
9     $luas = $panjang * $lebar;
10    //operasi
11    echo "Luasnya ".$luas;
12    /*output program menampilkan suatu hasil perhitungan var
integer panjang dan lebar*/
```

```
13     ?>
14     </body>
15     </html>
```

Output:



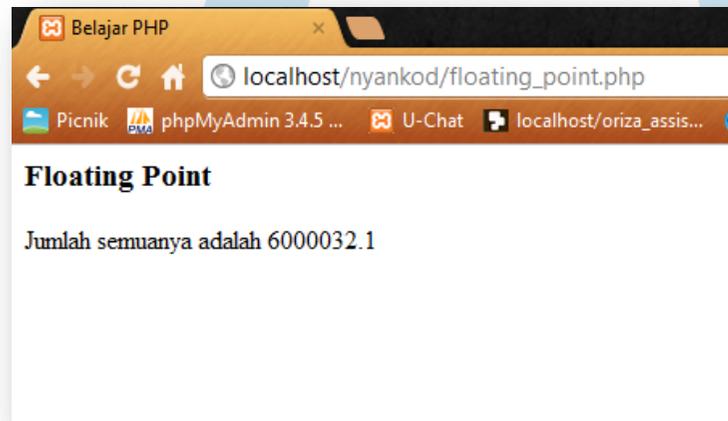
3. Floating-Point

Yaitu tipe data yang digunakan untuk menyimpan nilai bilangan real, float, ataupun double. Tipe data ini dapat dimanfaatkan untuk menyimpan angka yang bersifat pecahan sampai kepada nilai yang mempunyai banyak digit.

```
1     <html>
2     <head><title>Belajar PHP</title></head>
3     <body>
4         <h3>Floating Point</h3>
5         <?php
6             //inisialisasi
7             $a = 2;
8             $b = 30.10; //float
9             $c = 6000000;
```

```
10     $hasil = $a + $b + $c;
11     //operasi
12     echo "Jumlah semuanya adalah ".$hasil;
13     /*output program menampilkan suatu hasil perhitungan var
floating point dengan perbedaan inisialisasi masing-masing tipe
data*/
14     ?>
15 </body>
16 </html>
```

Output:



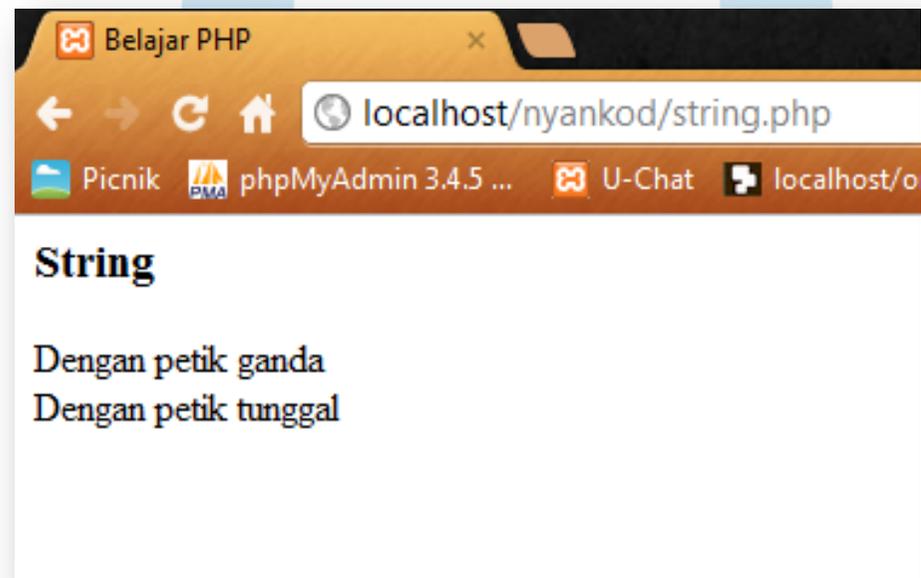
4. String

Yaitu tipe data yang menyatakan suatu karakter atau sekumpulan karakter. Penggunaan string dalam PHP dapat dinyatakan dengan petik tunggal ataupun petik ganda.

```
1 <html>
2 <head><title>Belajar PHP</title></head>
```

```
3 <body>
4   <h3>String</h3>
5   <?php
6     //inisialisasi
7     $kata = "Dengan petik ganda";
8     $kata2 = 'Dengan petik tunggal';
9     //operasi
10    echo $kata."<br>".$kata2;
11    /*output program menampilkan suatu nilai string dengan 2
12   cara*/
13   ?>
14 </body>
</html>
```

Output:

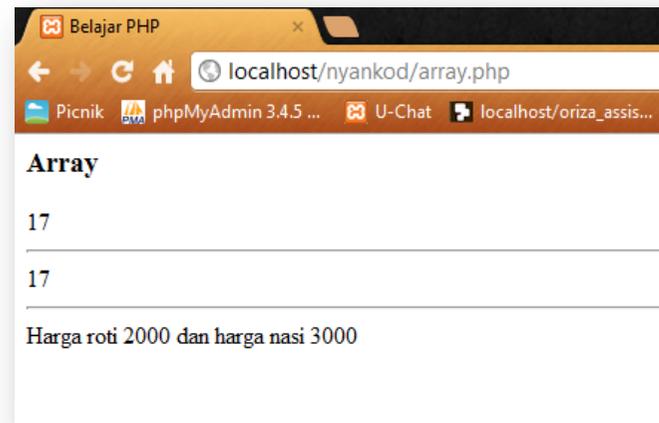


5. Array

Yaitu tipe data yang terdiri dari rangkaian variabel yang bertipe sama. Array bisa kita panggil dengan larik. Rangkaian elemen dalam larik bisa akses dengan index. Index array di dalam PHP dimulai dari nol (0).

```
1 <html>
2 <head><title>Belajar PHP</title></head>
3 <body>
4   <h3>Array</h3>
5   <?php
6     //array index angka, cara 1
7     $nilai[0] = 6; //membuat var array dan mengisinya dengan
tipe integer
8     $nilai[1] = 9;
9     $nilai[2] = 8;
10    echo "Jumlah semua nilainya : ".$nilai[0] + $nilai[1] +
$nilai[2]."<hr>";
11    //array index angka cara 2
12    $nilai_b = array ( 0=>6,1=>9,2=>8);
13    echo "Jumlah semua nilainya : ".$nilai_b[0] + $nilai_b[1]
+ $nilai_b[2]."<hr>";
14    //array index string
15    $harga["roti"] = 2000;
16    $harga["nasi"] = 3000;
17    echo "Harga roti ".$harga["roti"]." dan harga nasi
".$harga["nasi"];
18    /*source code diatas mengajarkan kita untuk memahami cara
menggunakan dan mengakses array dengan index berupa integer dan
string*/
19    ?>
20 </body>
21 </html>
```

Output:



6. Object

Yaitu tipe data yang lebih kompleks karena memiliki struktur tersendiri. Tidak seperti tipe data yang lain. Tipe data objek terdiri dari atribut (banyak jenis variabel yang digunakan) dan metode (fungsi/prosedur). Penggunaan objek ini lebih diutamakan untuk membuat program berorientasi objek. Objek sendiri merupakan instansiasi dari kelas. Lho.. jadi banyak kosakata baru nih :D.. hehe, supaya gak bingung kita lewat aja contoh penggunaan objek. Penulis janji akan membahas tentang objek dalam artikel di edisi berikutnya yaitu dalam pembahasan pemrograman berorientasi objek (Object Oriented Programming, OOP).

Keep Learning and Be The Best..

[Ruby]



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

Hello #{RubyWorld} – Bagian Number dan Operator

Hallo Hallo Hallo semuanya.. hha kembali lagi dengan stasium Ruby2012FM, Love it And Earn It :D

Kamu akan dirasuki oleh makhluk halus yang bernama ilmu **Ruby** :D, namun sebelum itu kita review dulu nih atau me-load dulu ingatan kita tentang ilmu **Ruby** sebelumnya di **RubyDojo** edisi 1, Apa aja sih yang sudah kita pelajari? Please answer the question:

- 1) Siapakah **Matz**?
- 2) Gimana Cara install **Ruby** di Komputer kamu ???
- 3) Cara Compile **Ruby** gimana? Trus irb itu apa?
- 4) Kenapa **kutip dua** dan **kutip satu** berbeda?
- 5) Apa aja sih kelebihan **Ruby** dibanding bahasa pemograman lain? Jabarkan donk!

Nah loh, bisa jawab ga pertanyaan di atas? Klo ga bisa berarti kamu belum belajar dengan betul atau mungkin belum tau edisi yang pertama, atau mungkin kamu hanya gaya-gayaan download majalah pertama n disimpan di komputer kamu sebagai koleksi aja tanpa kamu baca, atau mungkin kamu download untuk di cetak n dijual,,,ato mungkin X_X haduh parah.... hahaha



Kalau sudah mantap dan kerasukan ilmu Ruby pertama, kamu sudah siap menerima latihan untuk mengupgrade kemampuan kamu ke level 2 XD

Let's Rolling (^o^)

Sebelum dirasuki ilmu baru, lebih baik pahami dulu pertanyaan saya di atas ya, cari tau jawabannya di nyankod's Mags edisi 1 (Edisi 1: Nyankod Say "Hello World") :P download di <http://nyankod.blogspot.com>.

Level Up

Sekarang kita akan meng-upgrade tingkat kerubian kamu ke level 2, sebelum memulai siapkan dirimu dan siapkan sesajen berikut :

- 1) Buka Folder **RubyDojo**.
- 2) Buka Editor teks kesayangan kamu.
- 3) Buka **Ruby** editor interaktif (irb) di terminal | cmd.

Kamu sudah siap untuk berlatih di **RubyDojo** sekarang, mari kita bermain dan berlatih. Di **Ruby**, ada dua tipe number yaitu bilangan bulat dan bilangan pecahan. Bilangan bulat disebut dengan **integer**, bilangan cacah disebut **float**. Contoh untuk integer seperti angka 012345 dan seterusnya, tanpa ada koma dan pecahan. Untuk float angkanya sama dengan integer namun dengan koma dan pecahan setelahnya, contoh 1.5, 2.8924, 3.14628369121 dan sebagainya :D..

Sekarang silahkan kamu coba di irb dengan perintah berikut :

```
irb(main):001:0> puts 10+10, 10-10, 10*10, 10/10
20
0
100
1
=> nil
```

catatan : koma disini sebagai separator eksekusi proses.

Perhatikan contoh dari bilangan bulat (integer) diatas, masing masing penghitungan dengan operator standar yang biasanya ada di kalkulator tukang sayur itu *ups . Ada operator tambah (+), kurang (-), kali (*), bagi (/). sekarang silahkan coba contoh bilangan pecahan (float):

```
irb(main):002:0> puts 10.5+10.5, 10.5-10.5,  
10.5*10.5, 10.5/10.5  
21.0  
0.0  
110.25  
Nyangkod's edisi 21.0  
=> nil
```

Di bilangan pecahan (float) terdapat hasil bulat, namun karena yang di eksekusi adalah angka float maka hasilnya pun akan float dengan ditandai satu angka 0 dibelakang koma.

Kemudian, bagaimana ceritanya kalau yang kita hitung adalah angka pecahan dan bulat? Apa yang terjadi??? o_O

```
irb(main):003:0> puts 14.245+10  
24.245  
=> nil
```

Ternyata, hasilnya adalah angka float. Inilah yang terjadi di Ruby angka yang dihasilkan dapat otomatis meng-casting(mengubah tipe dari suatu variabel) sendiri seperti yang dijelaskan di level 1 kemarin. Ini juga berlaku terhadap variable apapun yang digunakan di Ruby, bersifat flexible dan dapat berubah sesuai kebutuhan.

Nah sekarang lihat tabel dibawah ini, dapet dari web <http://rubylearning.org> sih, dia dapat dari buku/e-book (courtesy: Dave Thomas' -

Programming Ruby), trus saya ubah dikit supaya mudah dimengerti :P diurut dari tingkat keutamaan teratas sampai terbawah.

Method	Operator	Deskripsi
★	[] []=	kumpulan dari, kumpulan dari sama dengan
★	**	Eksponensial
★	! ~ + -	Not, komplemen/pelengkap, penambahan dan pengurangan
★	* / %	Perkalian, bagi, dan modulus/ sisa
★	+ -	Tambah dan kurang
★	>> <<	Geser kanan dan kiri
★	&	"and"
★	^	XOR (eksklusif OR) dan OR
★	<= < > >=	Operator perbandingan
★	<=> == === != =~ !~	Kesamaan dan operator yang berpola kesesuaian (!~ dan != bukan operator)
	&&	Logical "and"
		Logical "or"
	Range/luas (inclusive dan exclusive)
	?:	if-then-else
	= %= /= -= += = &= >>=	Assignment / penugasan gabungan
	<<= *= &&= = **=	
	define?	Cek jika symbol telah ditentukan
	not	Negasi logic
	or and	Logical Composition
	if unless while until	Expression modifiers
	begin/end	Block expression

Kerjain nih:

1. Buat file Ruby di folder RubyDojo yang isinya penjumlahan pengurangan perkalian pembagian dengan angka bulat dan pecahan. Kemudian eksekusi.
2. Eksekusi angka 5 lebih besar dari 10, menggunakan operator yang ada di tabel kemudian menghasilkan keluaran false.
3. Cari sisa bagi menggunakan % (modulus) dari 10 dengan 4, 5 dengan 1, 3 dengan 4, dan 3 dengan 5.

Pusing ga tuh liatnya? Itu operator-operator yang nantinya kamu gunakan di **Ruby**, jadi jangan sampai lupa ya... lebih bagus lagi kalau kamu cetak lalu tempel di dinding kamar, sebelum tidur kamu hapal deh ahaha... Ada catatan bagus nih dari tabel di atas, inget ya *hapalnyabacaberulang-ulang.

- 1) Operator untuk menaikkan (++) dan menurunkan (--) nilai dari suatu variable atau method tidak berlaku di **Ruby**, untuk menangani itu bisa menggunakan +=1 atau -=1 karena sifatnya sama yaitu menaikkan satu angka.

- 2) Sama seperti ilmu matematika yang kita pelajari sewaktu SD, kalau yang di dalam kurung itu yang di itung lebih dulu atau istilahnya di eksekusi terlebih dahulu. Contoh : $(2+4)*4$.
- 3) Ada tanda bintang di bagian modul di tabel di atas artinya bahwa operator tersebut bukan hanya operator sederhana, namun juga memanggil method.

Wowh, sudah cukup banyak nih...rehat dulu ah sebentar...sekarang kamu latihan dulu sedikit supaya lebih fasih sebelum lanjut ke bagian sulit di level 2 ini :D biar Hafal dan paham.... kan ada istilah "Bisa karena terbiasa".

HERO state :: Extra Fokus

Di tahap ini, level 2 akan butuh perhatian khusus lagi nih... siap-siap ya :D

Tadi sudah kita pelajari mengenai aritmatika penghitungan dasar dengan menggunakan $+$ $-$ $*$ $/$ $\%$. sekarang kita akan berlatih untuk lebih memahami tentang penggunaan $+=$ $-=$ $*=$ $/=$ $\%=$, apa sih sebenarnya operator itu? Kenapa bisa dikatakan sebagai pengganti dari $++$ yang biasanya digunakan untuk menaikkan nilai variable sebanyak 1 angka. Nah sekarang fokuskan pikiran kamu, tarik nafas dalam-dalam...santai tapi fokus oke..

```
irb(main):004:0> varPagi = 10
=> 10
```

di atas kita buat variable varPagi diisi dengan nilai integer 10. Selanjutnya naikan satu nilai varPagi dengan cara:

```
irb(main):005:0> varPagi = varPagi + 1
=> 11
```

penjelasannya dibawah ini nih...



Sifat dari += di samping dimiliki juga oleh -= *= /= %= .

Nilai yang ditambahkan tidak hanya selalu angka, namun bisa juga dengan variabel atau method lainnya yang memiliki nilai integer atau float.

Ini sangat efektif digunakan dalam perulangan dan sebagainya (akan kita bahas berabad abad lagi :D)

di si HIJAU ada varPagi + 1 yang menghasilkan nilai 11, kemudian nilai tersebut dimasukan ke nilai varPagi KUNING. Dan hasil akhirnya nilai varPagi yang tadinya 10 menjadi 11.

Sekarang, kamu coba asimilasikan...ups bahasanya ribet ya, maksudnya anggap bahwa varPagi di atas adalah orang(variabel maksudnya) yang sama. Jika orang tersebut sudah punya 10 Hape misal (*juraganHape) terus dia membeli 1 hape lagi, maka berapa hape yang dia punya?? kamu bisa jawab!!!

Pasti 11 kan... hahaha. Sekarang perhatikan di bawah ini.

```
irb(main):006:0> varPagi = 10
=> 10
irb(main):007:0> varPagi += 1
=> 11
```

perintah pertama, kamu memberikan nilai si varPagi menjadi 10.lalu di perintah kedua, kamu membuat varPagi bertambah 1 nilai menjadi 11.

Artinya tanda += disini menambahkan nilai yang dimiliki variabel sebelum tanda += atau disini si varPagi, dengan angka setelah tanda += yaitu 1.

Jadi, sama fungsinya dengan varPagi = varPagi+1 , namun diefektifkan menjadi varPagi += 1.

Akhirnya, setelah lama-lama berlatih di **RubyDojo**, kamu sudah lolos naik ke level 2 ini. Daritadi kamu menghitung terus menerus kayak belajar matematika kelas SD aja tentang aritmatika dasar...hahaha. Ok kita simpulkan apa saja yang sudah kita kuasai di level 2 ini:

1. Kamu sudah mengetahui tipe angka di **Ruby** yaitu **integer** dan **float**.
2. Kamu sudah tau apa aja operator yang ada di **Ruby**, ada di tabel itu loh.
3. Kamu sudah tau sekarang bagaimana untuk mengatasi penggunaan (++) atau (-) yang biasanya ada di bahasa pemograman lain di **Ruby**.
4. Kamu sudah mengerti dan jago menggunakan += -= *= /= %= .

Lebih baik dilatih lagi ya tentang penguasaan aritmatika dasar ini, karena akan sering digunakan nantinya....**MORE FIRE**

Salam akhir saya harap tulisan ini dapat bermanfaat bagi kamu dan lebih tertarik lagi untuk PDKT dengan Ruby, mohon maaf apabila ada kesalahan kata atau kurang berkenan.

Apabila ada pertanyaan dan sebagainya bisa menghu-bungi saya di kontak: <http://about.me/MuhammadSinggihZA>.

Salam, hirokakaoshi

[ActionScript]



Tarom Apriyanto
@tarompey
tarompey@gmail.com



Variabel

Hallo apa kabar sobat nyankod, ketemu lagi ni heeeehhhh.....

sebelum masuk ke materi edisi ke dua, alangkah baiknya saya menanyakan kabar buat sobat nyankod yang masih setia mengunduh dan belajar materi dari kami para nyakodist, Apa kabarnya nih sobat nyankod???....(baik donk...!!). kirim kritik dan sarannya yaaa, mengenai bagaimana isi dari materi edisi perdana kami (kurang atau malah ada kesalahan) di WEB atau melalui email (nyankod@gmail.com) bahkan lebih lagi email pribadi para nyakodist untuk materi yang bakal sobat-sobat nyankod dalam, khususnya Materi dari saya yaitu ActionScript.

Oia sobat nyakod, sekali lagi saya paparkan ActionScript ini hanya dapat berjalan apabila sobat nyankod memiliki software peng-compilennya, bisa menggunakan Flash player dan juga software pendukungnya seperti Adobe Flash CS4 yang saya pakai karena ini adalah bahasa pemrograman berorientasi objek.

Mengenal “Pengenal”

Pengenal (identifier) adalah suatu nama yang digunakan dalam program untuk menyatakan variabel, fungsi bla bla bla (dan lain-lain maksudnya). Aturan umum yang berlaku dalam membuat pengenal pada Action Script:

1. Harus diawali dengan Alphabet (A,a,B,b dst) atau Underscore (_), (ga boleh angka duluan)
2. Tidak boleh mengandung spasi atau titik (.)
3. Tidak boleh menggunakan keyword dari ActionScript seperti : while, do, if, else, return, delete, this, with, true, false dan bla bla bla lagi (heeheheheee).
4. Huruf besar dan kecil harus di perhatikan, karena sangat berpengaruh (case sensitive).

Mengenal Variabel dan Tipe Data

Mengenai variabel, saya dapatkan analogi yang pas agar penjelasan variabel ini lebih mudah baru nanti dikasih tahu pengertiannya. (Dapat dari mana Analoginya? Jawabanya: dari bukulah kan sama-sama masih belajar....hehehehehe) begini ni analoginya:

"kalau kita ingin menyimpan air, tentu kita perlu tempatnya. Begitu juga di dalam ActionScript, ketika kita perlu menyimpan data/informasi maka kita perlu variabel untuk menyimpannya".

Dilanjutkan

"kalau kita ingin memilih tempat menyimpan air, maka pasti akan kita pilih-pilih air jenis apa yang mau kita simpan. Kalau kita punya air siroup maka akan lebih cocok jika disimpan dalam gelas. Jika kita punya air teh, maka akan lebih cocok kita simpan di cangkir. Jika kita punya air untuk mencuci, maka akan lebih cocok jika kita simpan di ember".

Bayangin coba kalau kita nyimpen air siroup di ember besar, kan ga sesuai tempatnya bray. (^__^)*****[di bayangin]**.

Pada ActionScript pun seperti demikian, jika kita ingin menyimpan data angka, maka simpan dalam tipe data Number, jika ingin menyimpan data teks/tulisan maka disimpan dalam tipe data String bla bla bla (dst). Dan seperti analogi di atas, jika suatu data disimpan pada variable yang tidak cocok tipe datanya, maka akan muncul kesalahan.

Jadi begitu lah menurut buku yang saya baca. Untuk pengertiannya, Variabel adalah suatu nama yang menyatakan tempat dalam memori komputer yang digunakan untuk menyimpan suatu nilai dan nilainya dapat diubah sewaktu-waktu ketika program sedang dieksekusi (misalnya : dulunya suatu variabel bernilai 27 sekarang berisi atau di ganti dengan 29, begitulah kira-kira). Biar lebih jelas lihat tabel_1 di bawah ini:

Tipe Data	Keterangan	Contoh deklarasi dan assignment
Number	Angka	<code>var index:Number; Index = 3;</code>
String	Teks atau Tulisan	<code>var kode_masuk:String; kode_masuk="nyankod";</code>
Boolean	Nilai yang berisi benar (true) atau salah (false)	<code>var jalankan:Boolean; Jalankan = true;</code>
Object	Tipe data objek	<code>var suaraku = new Sound (); suaraku.attachSound("ketemu.wav");</code>

Tipe Data Array

Jika kita hubungkan dengan kebutuhan penyimpanan data dalam ActionScript, maka sering kali kita menjumpai bahwa kita memiliki data yang

bertipe sejenis pada konteks yang sama. Misalnya kita memiliki data 3 buah soal. Ketiga soal tadi kita simpan dalam variable-variabel bertipe string.

Contoh:

```
1 | var tanya1:String = "Siapakah pendiri Nyankod magazine?";
2 | var tanya2:String = "Siapakah nama-nama Anggota Nyankod
   | magazine?";
3 | var tanya2:String = "Siapa yang mengisi materi ActionScript
   | ini?";
4 | //narsis dikit hehehehehe
```

Dengan menggunakan script di atas, maka kita harus mempunyai 3 variabel untuk menyimpan tiga pertanyaan. Sebenarnya semua soal tersebut bisa kita simpan dalam satu variable tetapi memiliki banyak variasi pertanyaan. Tipe data variable yang memiliki sifat tersebut adalah array.

Array adalah sebuah variable yang dapat menampung beberapa data dengan tipe sejenis. Seperti kereta api, array memiliki nama dan memiliki index ("nomor kereta"). Sehingga 3 soal di atas bisa kita deklarasikan dengan cara:

```
1 | var arrayTanya = new Array();
2 | arrayTanya = ['Siapakah pendiri Nyankod magazine?',
   | 'Siapakah nama-nama Anggota Nyankod magazine?',
   | 'Siapa yang mengisi materi ActionScript ini?'];
```

Untuk mendapatkan pertanyaan yang pertama, kita hanya menuliskan:

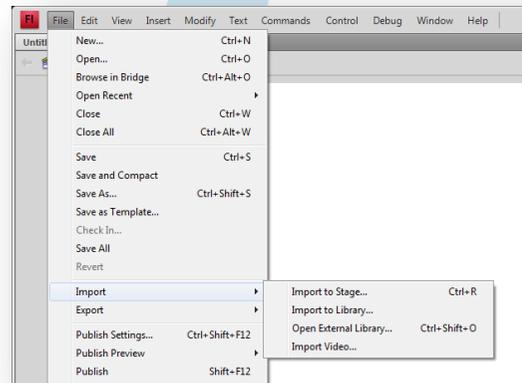
```
arrayTanya[0];
```

NB: Array dalam ActionScript dimulai dari index 0 dst.

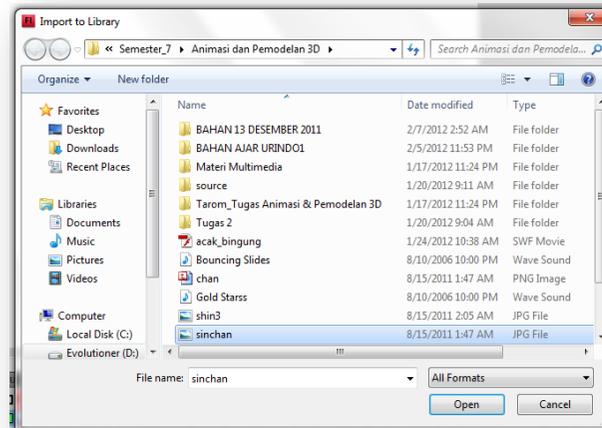
Instance Name

Jika variable kita beri nama, maka symbol harus diberi nama agar bisa dipanggil. Nama tersebut adalah Instance Name. Sebagai contoh:

1. Pastikan sobat dah masuk ke Adobe flash dan pilih buat ActionScript 2.0
2. Kemudian klik menu File>>Import to Library (buat masukin gambar atau file lain sperti lagu, movie dll yang bakal kita gunakan nantinya)



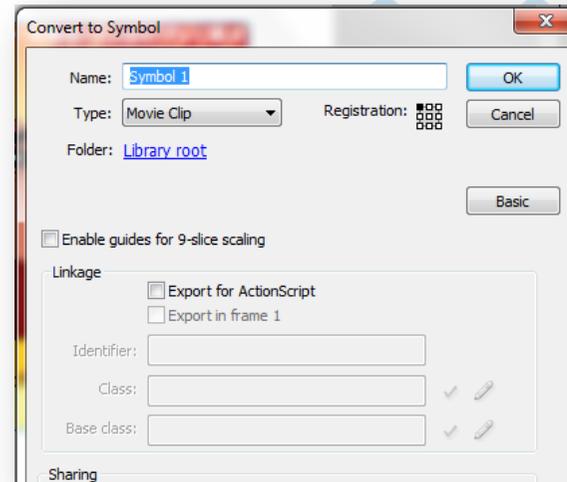
3. Cari filenya terletak di mana, pilih **Open**



4. Buka tab library yang ada di samping kanan lembar kerja atau windows ceklis panel library jika sudah di ceklis berarti sudah di sebelah kanan lembar kerja bareng sama Properties. Klik tab librarynya dan drag gambar yang sobat masukin tadi.



5. Kemudian klik kanan pada gambar dan pilih Covert to Symbol. Beri nama dan pada type pilih Movie clip lalu OK.



6. Klik pada gambar, kemudian klik tab Properties yang ada di sebelah kanan lembar kerja lalu ketikkan nama di "Instance Name" sebagai nama variabel untuk kita panggil dan di jalankan pada panel Action.

Gimana mudah kan memberikan variabel pada animasi (Movie clip) yang sudah kita buat sebelumnya. Nah, untuk sekarang sobat Nyankod harus sabar mengenai bagaimana pemanggilan variabel pada animasi ini bisa diterapkan karena tema pembelajaran ini masih mengenal type data dan variabel sobat. So ikutin terus ya pembelajaran Nyankod ActionScript ini hingga nanti kita bisa membuat program aplikasi yang menggunakan Animasi.

OK sobat sampai di sini dulu materinya, kata pepatah "*kena getah buah duku, maka udah dulu*" (...@_@...???!***8, ngawor pepatah dari mana itu....**##\$)

[C]



Ade Kurniawan
@adekurniawan
noadekur@yahoo.com



Tipe Data di Bahasa C

Setelah berkenalan dengan bahasa C di pembahasan awal, langkah selanjutnya adalah mengenal tipe data yang digunakan dalam bahasa C. Di dalam bahasa pemrograman komputer, data yang digunakan umumnya dibedakan menjadi nilai data numerik dan nilai data karakter. Nilai numerik dapat dibedakan lagi menjadi nilai numerik integer(bulat) dan nilai numerik pecahan. Nilai numerik pecahan dapat dibedakan lagi menjadi nilai numerik pecahan ketepatan tunggal dan nilai numerik pecahan ketepatan ganda. Bahasa C menyediakan lima macam tipe data dasar, yaitu tipe data integer(bilangan bulat dideklarasikan dengan int), floating point(nilai numerik pecahan ketepatan tunggal dideklarasikan dengan float), double-precision(nilai numerik pecahan ketepatan ganda dideklarasikan dengan double), karakter(dideklarasikan dengan char), kosong(dideklarasikan dengan void).

No	Tipe Data	Ukuran	Jangkauan	Format	Keterangan
1	char	1 byte	-128 s/d 127	%c	Karakter / string
2	int	2 byte	-32768 s/d 32767	%d, %i	Integer / bilangan bulat
3	float	4 byte	-3.4E-38 s/d 3.4E+38	%f	Float / bilangan pecahan
4	double 8 byte	8 byte	-1.7E-308 s/d 1.7+308	%lf	Pecahan presisi ganda
5	void	0 byte			Tidak bertipe

Tipe Data Karakter

Variabel tipe karakter dideklarasikan dengan tipe char atau signed char dan dimaksudkan untuk menampung nilai sebuah huruf atau karakter. Contoh penulisannya seperti ini:

```
1 | # include <stdio.h>
2 | main()
3 | {
4 |     char karakter;
5 |     karakter = 'a';
6 |     printf("karakter yang disimpan = %c", karakter);
7 | }
```

Tipe Data Integer

Variabel numerik integer sering menimbulkan masalah, seperti dalam hal penampilan nilainya yang tidak sesuai dengan yang seharusnya, hal ini disebabkan karena variabel numerik integer hanya menampung nilai bulat saja dan lebar tempat yang disediakan terbatas, terutama untuk nilai integer 16 bit. Berikut karakteristik nilai integer:

1. Nilai numerik pecahan yang disimpan di tempat nilai integer akan dibulatkan ke bawah(dibuang nilai pecahannya).
2. Nilai numerik negatif disimpan di memori dengan cara komplemen 2(two,s complement).

Contoh penulisan:

```
1 | # include <stdio.h>
2 | main()
3 | {
```

```
4 | int x,  
5 | unsigned int y;  
6 | x = 3276789;  
7 | y = -3455;  
8 | printf("x = %d \n", x);  
9 | printf("y = %u \n", y);  
10 | }
```

Tipe Data Float dan Double Precision

Variabel numerik float dan double precision digunakan untuk menampung nilai pecahan ketepatan tunggal dan ganda.

Contoh penulisan:

```
1 | # include <stdio.h>  
2 | main()  
3 | {  
4 |     float x = 1.234;  
5 |     double y = 1.897698;  
6 |     printf("x = %f \n", x);  
7 |     printf("y = %le \n", y);  
8 | }
```

Nb: untuk menampilkan nilai pecahan ketepatan tunggal dengan notasi saintifik dapat digunakan dengan kode "%e", sedangkan untuk nilai pecahan ketepatan ganda dapat digunakan kode format "%le".

[Bash]



Taufik Sulaeman
@taufiksu
taufiksu@gmail.com
<http://npaperbox.com/user/profile/1>



Bash Programming - Basic

Sedikit demi sedikit lama-lama menjadi bukit, sebuah peribahasa yang dapat digunakan oleh kita dalam proses belajar. Mari kita mulai dengan dasar memprogram bash.

Output

Untuk menampilkan output dengan BASH kita dapat membuat program dengan cara-cara berikut:

1. Masuk ke terminal/console

```
taufik@ubuntu:~$
```

2. Cek kita direktori kita berada dimana, dengan cara:

```
taufik@ubuntu:~$ pwd  
/home/taufik
```

3. Mari kita membuat folder untuk menaruh file-file yang akan digunakan untuk berlatih

```
taufik@ubuntu:~$ mkdir -p latihanbash
taufik@ubuntu:~$ ls
Application      Download  latihanbash
taufik@ubuntu:~$ cd latihanbash
taufik@ubuntu:~/latihanbash$
```

4. Gunakan perintah touch untuk membuat file baru.

```
taufik@ubuntu:~/latihanbash$ touch output.sh
```

5. Mari kita mulai membuat dengan menggunakan editor nano

```
taufik@ubuntu:~/latihanbash$ nano output.sh
```

6. Isi program

```
1 | #!/bin/bash
2 | #File: output.sh
3 | #Output dengan bash
4 | #Output string, perhatikan petik satu dan petik dua
5 | echo "Mangkod bilang: "
6 | echo -n "'Mangkod suka belajar koding dari yang termudah'"
7 | echo 'Mangkod seneng bisa ngoding bash :-D''
```

7. Untuk menutup editor nano tekan Ctrl+x pada saat itu akan ada pertanyaan untuk menyimpan perubahan pilih ya dengan ketik y kemudian tekan enter. Cara menjalankannya dengan

```
taufik@ubuntu:~/latihanbash$ sh output.sh
```

8. Selesai.

Data

Seperti makan nasi tanpa sambal, gak pedes rasanya. Nah begitu juga memprogram tanpa data (Gak jelas lanjut aja ya). Data adalah suatu nilai yang bisa diolah oleh komputer. Asalnya mungkin dari masukan pemakai, dan nantinya akan ditampilkan ke pemakai. Dalam program, data bisa berupa konstanta, yaitu data yang tetap. Sementara itu Variabel adalah simbol yang menyimpan suatu data, dan bisa berubah. Ayo kita mulai bikin programnya:

Variabel Program

1. Ulangi proses membuat file yang sudah dijelaskan sebelumnya. Kemudian mari kita edit dengan mengisi program berikut:

```
1 | #!/bin/bash
2 | #File : data.sh
3 | #Membuat variabel konstanta
4 | A="Nengkod"
5 | echo "Mangkod suka "$A
6 | #Rubah isi konstanta
7 | A="Nyankod"
8 | echo "Semua suka "$A
```

2. Save dan coba jalankan program tersebut.

Environment Variabel

Salah satu keistimeaan BASH adalah, environment variabel. Anda bisa melihat semua variabel tersebut dengan perintah set. Environment variabel merupakan variabel yang di set pada sebuah lingkungan utama sistem.

Environment variable tidak hanya seperti variabel global pada sebuah program saja, tetapi dapat digunakan berbagai program lainnya.

1. Mengecek environment variabel

```
taufik@ubuntu:~/latihanbash$ sh set | less
```

2. Mendefinisikan environment variable baru

```
taufik@ubuntu:~/latihanbash$ export ARTIS="Mangkod"
```

3. Memanggil environment variabel dari program

```
1 |#!/bin/bash
2 |echo "Hallo "$ARTIS
```

Input

"The only limiting factor of the Linux operating system is its user." –Linus

Input merupakan masukan dari user ke program yang dapat dimasukkan ke dalam variabel tertentu dan dapat dioperasikan kembali dalam program. Perintah yang digunakan untuk menerima input adalah `read`.

1. Isi program

```
1 |#!/bin/bash
2 |echo "Ohaio, Siapa namamu ?"
3 |read  NAME
4 |echo "Ohaio gozaimaz " $NAME ", saya Mangkod"
```

2. Simpan dan jalankan program tersebut.

Operasi

Operasi artinya mengolah data, sehingga didapat data dengan nilai baru. Masalahnya ada berbagai jenis tipe data, tipe data yang paling umum adalah huruf dan angka. Kita harus hati-hati membedakan jenis tipe data, karena kita harus menentukan variabel penyimpanannya, dan operasi yang dapat dilakukan. Pada pemrograman BASH, satu variabel dapat diisi jenis data apa saja.

Operasi String

1. Isi program

```
1 |#!/bin/bash
2 |# Variabel String
3 |VAR1="Nyankod"
4 |VAR2="Mangkod"
  |VAR3="Nengkod"
  |
  |# Penggabungan string
  |S1="$VAR1 $VAR2 $VAR3, pake kutip dua !"
  |S2='$VAR1 $VAR2 $VAR3, pake kutip satu !'
  |
  |# Nama variabel tergabung tanpa spasi
  |S3="$VAR1-Bash !"
  |S4="$VAR2Bash !"
  |S5=' ${VAR3}Bash !'
  |echo $S1
  |echo $S2
```

```
echo $S3  
echo $S4
```

2. Simpan dan jalankan

Operasi Aritmatika

1. Isi Program

```
1 #!/bin/bash  
2 # belajar berhitung  
3 echo "1+1=" 1+1  
4 echo "2+1=" ${2+1}  
echo "1*3=" ${1*3}  
echo "1/4=" ${1/4}
```

2. Save dan jalankan

```
taufik@ubuntu:~/latihanbash$ sudo -s  
root@ubuntu:~/latihanbash$ chmod 777 aritmatika.sh  
root@ubuntu:~/latihanbash$ ./aritmatika.sh
```

Operasi Logika

Untuk operasi logika mari kita pelajari di edisi berikutnya. ;)

[JavaScript]



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

JavaScript Object

Ketemu lagi dengan pembahasan javascript, bahasa pemrograman yang sangat fleksibel. Sesuai kesepakatan para nyankodist, di edisi kedua ini kita akan membahas tentang tipe data.

Tapi, tunggu, kenapa JavaScript ini judulnya Javascript Object??

Perlu kita ketahui, bahwasanya JavaScript memang bahasa pemrograman berbasis objek. Maksudnya, di JavaScript ini kita diperbolehkan untuk mendefinisikan sendiri sebuah objek dan membuat sendiri tipe variabel.

Maksudnya??

Begini.. Lain dengan banyak bahasa pemrograman lain yang mengharuskan seorang programmer mendefinisikan terlebih dahulu tipe data ketika mendeklarasikan variabel. Contohnya seperti yang Kamu lihat di pembahasan tentang ActionScript dan C. JavaScript juga tidak mengharuskan programmer untuk mendeklarasikan setiap variabel yang akan di gunakan di awal program. Variabel yang kita perlukan dapat dibuat dimanapun dan kapanpun cukup dengan memanggil kata kunci `var` diikuti nama variabel.

```
1 <script type="text/javascript">
2   var teks="Hai, Nyankoders!";
3   document.write(teks.length);
4 </script>
```



Properties adalah nilai tertentu yang dimiliki oleh sebuah objek.

Pada contoh di atas, kita mendeklarasikan variabel bernama `teks` dan mengisinya dengan string "Hai, Nyankoders!". Disini kita tidak perlu menentukan

tipe data secara spesifik. JavaScript akan menyesuaikan tipe data dengan data yang diinputkan. Nah, pada saat ini, kita sebenarnya sudah menggunakan *object* pada JavaScript. Salah satu buktinya adalah baris ketiga program di atas. Variabel teks yang sudah kita buat ternyata punya properties bernama `length` yang mengembalikan nilai berupa jumlah karakter yang terdapat pada variabel `teks`. Pada saat program di atas dijalankan, maka program akan mencetak keluaran:

12

Selain memiliki properties, sebuah objek juga memiliki method. Method serupa dengan fungsi, tetapi berada di dalam tubuh objek itu sendiri. Contohnya seperti berikut:



Method adalah sebuah **aksi** atau **fungsi** yang terdapat di dalam sebuah objek.

```
1 <script type="text/javascript">
2   var teks="Hai, Nyankoders!";
3   document.write(teks.toUpperCase());
4 </script>
```

Pada baris ketiga program di atas, kita memanggil method `toUpperCase()` dari variabel objek `teks` yang berfungsi untuk mencetak keluaran berupa nilai objek yang telah dirubah menjadi huruf kapital semua. Hasil keluaran program di atas adalah seperti ini:

HAI, NYANKODERS!

So, **dari mana datangnya properties dan method** tersebut?? Kenapa ketika kita mendeklarasikan variabel `teks` sejurus kemudian variabel tersebut punya properties `length` dan method `toUpperCase()` ?

Object

Seperti yang sudah disinggung di atas, JavaScript adalah pemrograman berbasis pada objek. Artinya setiap variabel yang kita buat, hakikatnya adalah sebuah objek.

Lalu, menanggapi pertanyaan di atas, dari mana datangnya properties dan method yang muncul pada sebuah variabel? Jawabannya kira-kira, dari objek itu sendiri. Atau dengan kata lain, sebuah objek ketika ia dibuat, ia telah dilengkapi dengan properties dan method yang sesuai dengan datanya.

Jadi, **JavaScript akan membuat objek sesuai dengan data yang dimasukkan??**

Hmm.. begitulah kira-kira.

Ada beberapa tipe objek standar yang sudah disediakan oleh JavaScript lengkap dengan properties dan methodnya masing-masing. Tipe-tipe objek tersebut diantaranya:

Objek String

Objek string digunakan untuk memanipulasi konten berupa teks. Contoh dari objek string sudah kita lihat di awal pemaparan. Objek string dapat dibuat dengan cara:

```
var txt = new String("string");
```

atau lebih mudahnya langsung saja seperti ini:

```
var txt = "string";
```

Properties dan method yang dimiliki oleh objek string lengkapnya dapat dilihat di http://www.w3schools.com/jsref/jsref_obj_string.asp.

Objek Date

Objek date digunakan untuk bekerja dengan data tanggal dan waktu. Ada 4 cara membuat objek tanggal. Cekidot!

```
1 | var sekarang = new Date()
2 | var d1 = new Date("October 13, 1975 11:13:00")
3 | var d2 = new Date(79,5,24)
4 | var d3 = new Date(79,5,24,11,33,0)
```

Baris pertama untuk mendapatkan data waktu saat program dijalankan. Baris kedua untuk membuat objek tanggal sesuai dengan tanggal yang diinputkan. Baris ketiga dan keempat untuk membuat objek tanggal dengan format `new Date(year, month, day, hours, minutes, seconds, milliseconds)`. Baris ketiga adalah contoh bila kita hanya ingin memasukkan tanggal saja, tanpa memasukkan data waktu.

Properties dan method selengkapnya yang dimiliki objek date bisa dilihat di http://www.w3schools.com/jsref/jsref_obj_date.asp.

Objek Array

Array, seperti yang sudah dijelaskan di pembahasan tentang array pada bahasa pemrograman lain, adalah variabel yang dapat memuat lebih dari satu nilai. Cara membuat objek array adalah seperti ini:

```
1 | // membuat objek array
2 | var buah2an = new Array();
3 |
4 | // mengisi array
5 | buah2an[0] = "Jeruk";
6 | buah2an[1] = "Pepaya";
```

```
7 | buah2an[2] = "Pisang";
```

Selain itu ada juga cara lain yang lebih singkat, berikut contohnya.

```
1 | // array condensed
2 | var buah2an = new Array("Jeruk", "Pepaya", "Pisang");
3 | // array literal
4 | var sayuran = ["Bayam", "Wortel", "Kangkung"];
```

Cara memanggil nilai dalam suatu array adalah dengan memanggil nama variabel diikuti indeks array yang akan diambil nilainya. Contohnya:

```
1 | document.write(buah2an[0]);
```

Baris program di atas akan memunculkan keluaran

```
Jeruk
```

Lebih lengkap tentang properties dan method pada objek array dapat dilihat di http://www.w3schools.com/jsref/jsref_obj_array.asp.

Objek Boolean

Objek ini merepresentasikan dua nilai, 'true' dan 'false'. Cara membuat objek boolean adalah sebagai berikut:

```
1 | var myBoolean = new Boolean();
```

dengan nilai inisial berupa *true* atau *false*. Nilai false juga dapat diinisiasi oleh nilai 0, -0, null, "" (string kosong), undefined dan NaN.

Objek Math

Objek ini memudahkan kita dalam menyelesaikan tugas perhitungan seperti pembulatan, randomisasi, pangkat dan akar pangkat, dan sebagainya. Perbedaannya dengan objek lain, objek Math ini bukan constructor, artinya untuk menggunakan properties dan method math, kita hanya tinggal memanggilnya dari objek Math itu sendiri. Contohnya sebagai berikut.

```
1 | var x = Math.PI; // mengembalikan nilai PI  
   | var y = Math.sqrt(16); // mengembalikan akar kuadrat dari 16
```

Properties dan method yang bisa kita manfaatkan dari objek math ini bisa dicek di http://www.w3schools.com/jsref/jsref_obj_math.asp.

Demikian pembahasan tentang tipe data, dalam hal ini JavaScript object. Sebenarnya masih ada satu objek bawaan lagi dari javascript yang mesti dibahas, yakni RegExp Object, tapi sepertinya pembahasan tentang Regular Expression akan dibahas tersendiri di lain kesempatan, bareng bahasa pemrograman lainnya. So, sampai ketemu di edisi ketiga dan jangan lupa kunjungi tautan resource yang sudah saya cantumkan di tiap-tiap pembahasan objek di atas.

Terima kasih ;D

Cara Berlangganan Nyankod Magz via Email

Bagi Anda yang ingin berlangganan buletin Nyankod Magz via email, caranya gampang, udah gitu gratis lagi bro....

Gini nih caranya:

1. Buka homepage kami di <http://nyankod.blogspot.com>.
2. Pada sidebar sebelah kanan ada kolom Berlangganan Nyankod Magz. Silahkan isi form email dengan alamat email Anda, kemudian klik tombol Submit.

The screenshot shows a portion of the Nyankod Magz website. At the top right, there is a blue banner with the text "learn coding here....". Below this, on the right side, is a search bar with the placeholder text "Search" and a magnifying glass icon. Underneath the search bar is a section titled "Berlangganan Nyankod Magz". The text in this section reads: "Dengan mengisi form di bawah ini, Anda akan mendapatkan email notifikasi setiap edisi terbaru Nyankod Magz." Below the text is a form with an input field labeled "Email address..." and a "Submit" button. At the bottom of this section, there is a "Recommend" button and a social media link for Facebook with the text "Kresna Galuh BestFriend, Toni Y. Haryanto and 15". On the left side of the screenshot, there is a blue banner with the text "Nyankod Magz Edisi ke-2 akan terbit : Senin, 20 Februari 2012". Below this banner is a blog post titled "'Hello World'" with the author "ents". The post content begins with "Hello World... salam coders untuk Anda semua. Bagi Anda yang sedang belajar atau berniat untuk belajar pemrograman, sepertinya apa yang kami sajikan ini mungkin akan bisa membantu. Baiklah, ini adalah edisi pertama Nyankod Magz. Di dalamnya kami telah menyusun berbagai pembelajaran tentang pemrograman. Adapun bahasa pemrograman yang ada

3. Selanjutnya akan keluar pop-up Email Subscription Request. Isi form yang ada di bawah dengan kode yang tampil di layar. Setelah itu klik tombol Complete Subscription Request.



 FeedBurner™

Email Subscription Request

Thank you for your request.

nyankoders@email.com

... will receive a verification message once you submit this form. FeedBurner activates your subscription to "Nyankod.com" once you respond to this verification message.



To help prevent spam, please type the text you see in the box above:

©2004–2012 Google ([Terms of Service](#) • [Privacy Policy](#))

4. Beres deh, nanti setiap ada edisi terbaru dari Nyankod Magz akan langsung terkirim pula ke email Anda. Biasanya sih satu hari setelah penerbitan.

Bagi Anda yang merasa Nyankod Magz bermanfaat, dan Anda ingin memberikan apresiasi, maka silahkan kirimkan Testimoni Anda kepada kami, atau silahkan tulis langsung di halaman Testimoni di web kami.

Edisi ke-3 akan terbit: Senin, 5 Maret 2012



Coding is pulen bro!!!