

Terbilang Perl-way

oleh : wirawanBrotoyuwono

Terinspirasi oleh fungsi Terbilang karya seorang "MahaGuru" dalam site beliau ¹, saya menyusunnya ulang dengan logika yang (mungkin) berbeda. Ya menurut cara dan gaya 'seenak dan senyaman-nya' saya. It's my perl-way.

Ide awal yang terbayang adalah sederet teks dalam lembar kuitansi. Tercantum kolom 'terbilang' yang merupakan penerjemahan penyebutan angka-angka yang ada sebelumnya. Semacam penegasan tentang nilai angka dengan menjadikannya sebagai bentuk yang (-seharusnya-) lebih mudah dimengerti.

Format dan Pattern

Dalam lembar kuitansi, penulisan deretan nilai angka mengikuti sebuah pattern tertentu.

1
23
456
7 . 890
89 . 012
1 . 234 . 567
210 . 987 . 654 . 321
...
dst

Saya pandang pattern ini sebagai format default sebuah nilai angka. Saya asumsikan,

- Penandaan dengan karakter "." diberikan bila karakter angka lebih dari 3 buah karakter.

¹ Drs. Yohanes Suyanto, <http://yanto.staff.ugm.ac.id>

- Penghitungan karakter dilakukan dari sebelah kanan.(-seperti aturan penulisan huruf arab-).
- Secara umum tiap 'split-ing' angka dengan limiter karakter " ." tersusun dari nilai, **"ratusan --- puluhan --- satuan"**
- Pembilangan tiap kelompok "ratusan puluhan satuan" ditentukan oleh posisi karakter " ." -nya dalam suatu deretan angka. Urutan kelompok ini, (- dalam kenaikan perkalian 1000 -)
"trilyun --- miliar --- juta --- ribu ---"

misal ,

```
89123456789
== 89 . 123 . 456 . 789
== 89 miliar 123 juta 456 ribu 789
```

Saya anggap pertimbangan kondisi-kondisi di atas sudah cukup untuk menyusun sebuah fungsi 'terbilang'. Lantas saya merancang kerangka besar kegiatan ke arah hasil aplikasi 'terbilang'.

Aliran Fungsi dalam Skrip

Berikut langkah-langkahnya,

- Pernyataan interpreter ke arah Perl. Ya, sesuai 'tradisi'.

```
#!/usr/bin/perl
```

- Pembentukan array berisi kesesuaian pembilangan menurut index variabel dalam array-nya.

```
@bil=qw(
nol
satu
dua
tiga
empat
lima
enam
tujuh
delapan
sembilan
);
```

- Pengelompokan nilai satuan angka menurut posisinya.

```
$s=substr($angka,-1,1); $s="$bil[$s]";
$p=substr($angka,-2,1); $p="$bil[$p] puluh";
$r=substr($angka,-3,1); $r="$bil[$r] ratus";

$sb=substr($angka,-4,1); $sb="$bil[$sb] ";
$pb=substr($angka,-5,1); $pb="$bil[$pb] puluh";
```

```
$rb=substr($angka,-6,1); $rb="$bil[$rb] ratus";
```

dst.

- Penyingkapan dengan fungsi bersyarat mengenai panjang variabel angka.

```
if (length($angka) == 1){
    print substr($angka,-1,1);
    print "\n\n";
    $tba="$s";
    $tb="$tba";
};
#####
if (length($angka) == 2){
    print substr($angka,-2,1);
    print substr($angka,-1,1);
    print "\n\n";
    $tba="$p $s";
    $tb="$tba";
};
```

dst.

- Editing output dengan melakukan substitusi dalam proses pattern matching sebagai upaya mencapai 'kewajaran' pembilangan output.

```
$tb=~ s/satu puluh nol/sepuluh/g;
$tba=~ s/satu puluh satu/sebelas/g;
$tba=~ s/satu puluh dua/dua belas/g;
...
$tba=~ s/satu ratus/seratus/g;
$tba=~ s/nol puluh//g;
$tba=~ s/nol ratus//g;
$tba=~ s/nol//g;
$tba=~ s/ +/ /g;
$tba=~ s/juta ribu/juta/;
$tba=~ s/miliar juta/miliar/;
```

Perl-CGI

Untuk menggunakan skrip sebagai sebuah 'common gateway interface', saya perlu menambahkan pernyataan pemakaian modul CGI.

```
use CGI qw(:all);
```

```
print <<MULAI;
Content-type:text/html
```

```
<html><body>
```

```
MULAI
```

```
*** isi skrip fungsi ***
```

```
print <<AKHIR;  
</body></html>  
AKHIR
```

Lalu menghapus baris,

```
system("clear");  
print "\n\n\n\tAngka = ";  
$angka=<STDIN>;  
chop($angka);
```

Begitulah '**Terbilang**'. Begitulah '**Perl-way**'.
Dan akhirnya, begitulah gaya saya dengan logika yang seadanya..

minomartani, maret, 2007

Referensi

Fungsi Terbilang, <http://yanto.staff.ugm.ac.id>
Terbilang, PerlSourceCode – (terlampir)
Terbilang,(rev.1), PerlSourceCode – (terlampir)

(- Lampiran -)

OpenSourceCode – Perl
Terbilang(rev.1)
oleh : wirawanBrotoyuwono, 7 maret,2007

(- Atas komentar/saran dari seorang "Beliau", saya meringkas sourcecode aplikasi.
Tidak ada perubahan logika, hanya dilakukan penyederhanaan penulisan pada fungsi
berkait panjang karakter, **length(\$angka)**.-)

```
#!/usr/bin/perl

system("clear");
print "\n\n\n\tAngka (max. 12 digit) = ";
$angka=<STDIN>;
chop($angka);
@bil=qw(
nol
satu
dua
tiga
empat
lima
enam
tujuh
delapan
sembilan
);

$s=substr($angka,-1,1); $s="$bil[$s]";
$p=substr($angka,-2,1); $p="$bil[$p] puluh";
$r=substr($angka,-3,1); $r="$bil[$r] ratus";

$sb=substr($angka,-4,1); $sb="$bil[$sb] ";
$pb=substr($angka,-5,1); $pb="$bil[$pb] puluh";
$rb=substr($angka,-6,1); $rb="$bil[$rb] ratus";

$sc=substr($angka,-7,1); $sc="$bil[$sc] ";
$pc=substr($angka,-8,1); $pc="$bil[$pc] puluh";
```

```

$rc=substr($angka,-9,1); $rc="$bil[$rc] ratus";

$sd=substr($angka,-10,1); $sd="$bil[$sd] ";
$pd=substr($angka,-11,1); $pd="$bil[$pd] puluh";
$rd=substr($angka,-12,1); $rd="$bil[$rd] ratus";

print "\t                                = ";
#####AA
#####dd
if (length($angka) <= 12){
    print substr($angka,-12,1);
    print substr($angka,-11,1);
    print substr($angka,-10,1);
    print " . ";
    print substr($angka,-9,1);
    print substr($angka,-8,1);
    print substr($angka,-7,1);
    print " . ";
    print substr($angka,-6,1);
    print substr($angka,-5,1);
    print substr($angka,-4,1);
    print " . ";
    print substr($angka,-3,1);
    print substr($angka,-2,1);
    print substr($angka,-1,1);
    print "\n\n";
    $tba="$r $p $s";
    $tbb="$rb $pb $sb";
    $tbc="$rc $pc $sc";
    $tbd="$rd $pd $sd";
    $tb="$tbd miliar $tbc juta  $tbb ribu  $tba";

    if (length($angka) <4){
        $tb=~ s/$tbd ribu//;
    };
    if (length($angka) <7){
        $tb=~ s/$tbd juta//;
    };
    if (length($angka) <10){
        $tb=~ s/$tbd miliar//;
    };

};

#####
$tb=~ s/satu puluh nol/sepuluh/g;
$tb=~ s/satu puluh satu/sebelas/g;
$tb=~ s/satu puluh dua/dua belas/g;
$tb=~ s/satu puluh tiga/tiga belas/g;
$tb=~ s/satu puluh empat/empat belas/g;
$tb=~ s/satu puluh lima/lima belas/g;
$tb=~ s/satu puluh enam/lima belas/g;
$tb=~ s/satu puluh tujuh/tujuh belas/g;
$tb=~ s/satu puluh delapan/delapan belas/g;
$tb=~ s/satu puluh sembilan/sembilan belas/g;

```

```
$tb=~ s/satu ratus/seratus/g;
$tb=~ s/nol puluh//g;
$tb=~ s/nol ratus//g;
$tb=~ s/nol//g;
$tb=~ s/ +/ /g;
$tb=~ s/juta ribu/juta/;
$tb=~ s/miliar juta/miliar/;
#####
$tb=~ s/miliar/miliar\n\t/;
$tb=~ s/juta/juta\n\t/;
$tb=~ s/ribu/ribu\n\t/;
print "\n\t\"$tb\"\n\n\n";

print "\t=====\n".
      "\t[Enter] -- lagi      [Ctrl+c] -- keluar";
print "\n\n\n\n\n\n\t ";
<STDIN>;
system("perl terbilang1.pl");
```

(- Lampiran -)

OpenSourceCode – Perl

Terbilang

1. oleh : wirawanBrotoyuwono, 5 maret,2007

```
#!/usr/bin/perl

system("clear");
print "\n\n\n\tAngka = ";
$angka=<STDIN>;
chop($angka);
@bil=qw(
nol
satu
dua
tiga
empat
lima
enam
tujuh
delapan
sembilan
);

$s=substr($angka,-1,1); $s="$bil[$s]";
$p=substr($angka,-2,1); $p="$bil[$p] puluh";
$r=substr($angka,-3,1); $r="$bil[$r] ratus";

$sb=substr($angka,-4,1); $sb="$bil[$sb] ";
$pb=substr($angka,-5,1); $pb="$bil[$pb] puluh";
$rb=substr($angka,-6,1); $rb="$bil[$rb] ratus";

$sc=substr($angka,-7,1); $sc="$bil[$sc] ";
$pc=substr($angka,-8,1); $pc="$bil[$pc] puluh";
$rc=substr($angka,-9,1); $rc="$bil[$rc] ratus";

$sd=substr($angka,-10,1); $sd="$bil[$sd] ";
$pd=substr($angka,-11,1); $pd="$bil[$pd] puluh";
$rd=substr($angka,-12,1); $rd="$bil[$rd] ratus";
```



```

print "\t      = ";
#####AA
if (length($angka) == 1){
    print substr($angka,-1,1);
    print "\n\n";
    $tba="$s";
    $tb="$tba";
};
#####
if (length($angka) == 2){
    print substr($angka,-2,1);
    print substr($angka,-1,1);
    print "\n\n";
    $tba="$p $s";
    $tb="$tba";
};
#####
if (length($angka) == 3){
    print substr($angka,-3,1);
    print substr($angka,-2,1);
    print substr($angka,-1,1);
    print "\n\n";
    $tba="$r $p $s";
    $tb=$tba;
};
#####BB
if (length($angka) == 4){
    print substr($angka,-4,1);
    print " . ";
    print substr($angka,-3,1);
    print substr($angka,-2,1);
    print substr($angka,-1,1);
    print "\n\n";
    $tba="$r $p $s";
    $tbb="$sb";
    $tb="$sb ribu  $tba";
};
#####BB
if (length($angka) == 5){
    print substr($angka,-5,1);
    print substr($angka,-4,1);
    print " . ";
    print substr($angka,-3,1);
    print substr($angka,-2,1);
    print substr($angka,-1,1);
    print "\n\n";
    $tba="$r $p $s";
    $tbb="$pb $sb";
    $tb="$tbb ribu  $tba";
};
#####BB
if (length($angka) == 6){
    print substr($angka,-6,1);
    print substr($angka,-5,1);
    print substr($angka,-4,1);
    print " . ";

```

```

    print substr($angka,-3,1);
    print substr($angka,-2,1);
    print substr($angka,-1,1);
    print "\n\n";
    $tba="$r $p $s";
    $tbb="$rb $pb $sb";
    $tb="$tbb ribu  $tba";
};
#####cc
if (length($angka) == 7){
    print substr($angka,-7,1);
    print " . ";
    print substr($angka,-6,1);
    print substr($angka,-5,1);
    print substr($angka,-4,1);
    print " . ";
    print substr($angka,-3,1);
    print substr($angka,-2,1);
    print substr($angka,-1,1);
    print "\n\n";
    $tba="$r $p $s";
    $tbb="$rb $pb $sb";
    $tbc="$sc";
    $tb="$tbc juta  $tbb ribu  $tba";
};
#####cc
if (length($angka) == 8){
    print substr($angka,-8,1);
    print substr($angka,-7,1);
    print " . ";
    print substr($angka,-6,1);
    print substr($angka,-5,1);
    print substr($angka,-4,1);
    print " . ";
    print substr($angka,-3,1);
    print substr($angka,-2,1);
    print substr($angka,-1,1);
    print "\n\n";
    $tba="$r $p $s";
    $tbb="$rb $pb $sb";
    $tbc="$pc $sc";
    $tb="$tbc juta  $tbb ribu  $tba";
};
#####cc
if (length($angka) == 9){
    print substr($angka,-9,1);
    print substr($angka,-8,1);
    print substr($angka,-7,1);
    print " . ";
    print substr($angka,-6,1);
    print substr($angka,-5,1);
    print substr($angka,-4,1);
    print " . ";
    print substr($angka,-3,1);
    print substr($angka,-2,1);
    print substr($angka,-1,1);
    print "\n\n";
};

```

```

    $tba="$r $p $s";
    $tbb="$rb $pb $sb";
    $tbc="$rc $pc $sc";
    $tb="$tbc juta    $tbb ribu    $tba";
};

#####dd
if (length($angka) == 10){
    print substr($angka,-10,1);
    print " . ";
    print substr($angka,-9,1);
    print substr($angka,-8,1);
    print substr($angka,-7,1);
    print " . ";
    print substr($angka,-6,1);
    print substr($angka,-5,1);
    print substr($angka,-4,1);
    print " . ";
    print substr($angka,-3,1);
    print substr($angka,-2,1);
    print substr($angka,-1,1);
    print "\n\n";
    $tba="$r $p $s";
    $tbb="$rb $pb $sb";
    $tbc="$rc $pc $sc";
    $tbd="$sd";
    $tb="$tbd miliar $tbc juta    $tbb ribu    $tba";
};

#####dd
if (length($angka) == 11){
    print substr($angka,-11,1);
    print substr($angka,-10,1);
    print " . ";
    print substr($angka,-9,1);
    print substr($angka,-8,1);
    print substr($angka,-7,1);
    print " . ";
    print substr($angka,-6,1);
    print substr($angka,-5,1);
    print substr($angka,-4,1);
    print " . ";
    print substr($angka,-3,1);
    print substr($angka,-2,1);
    print substr($angka,-1,1);
    print "\n\n";
    $tba="$r $p $s";
    $tbb="$rb $pb $sb";
    $tbc="$rc $pc $sc";
    $tbd="$pd $sd";
    $tb="$tbd miliar $tbc juta    $tbb ribu    $tba";
};

#####dd
if (length($angka) == 12){
    print substr($angka,-12,1);
    print substr($angka,-11,1);
    print substr($angka,-10,1);
    print " . ";
    print substr($angka,-9,1);

```

```

print substr($angka,-8,1);
print substr($angka,-7,1);
print " . ";
print substr($angka,-6,1);
print substr($angka,-5,1);
print substr($angka,-4,1);
print " . ";
print substr($angka,-3,1);
print substr($angka,-2,1);
print substr($angka,-1,1);
print "\n\n";
$tba="$r $p $s";
$tb="$rb $pb $sb";
$tb="$rc $pc $sc";
$tb="$rd $pd $sd";
$tb="$tbd miliar $tbc juta $tbb ribu $tba";
};

#####
$tb=~ s/satu puluh nol/sepuluh/g;
$tb=~ s/satu puluh satu/sebelas/g;
$tb=~ s/satu puluh dua/dua belas/g;
$tb=~ s/satu puluh tiga/tiga belas/g;
$tb=~ s/satu puluh empat/empat belas/g;
$tb=~ s/satu puluh lima/lima belas/g;
$tb=~ s/satu puluh enam/lima belas/g;
$tb=~ s/satu puluh tujuh/tujuh belas/g;
$tb=~ s/satu puluh delapan/delapan belas/g;
$tb=~ s/satu puluh sembilan/sembilan belas/g;

$tb=~ s/satu ratus/seratus/g;
$tb=~ s/nol puluh//g;
$tb=~ s/nol ratus//g;
$tb=~ s/nol//g;
$tb=~ s/ +/ /g;
$tb=~ s/juta ribu/juta/;
$tb=~ s/miliar juta/miliar/;
#####
$tb=~ s/miliar/miliar\n\t/;
$tb=~ s/juta/juta\n\t/;
$tb=~ s/ribu/ribu\n\t/;
print "\t"$tb"\n\n\n\n";

print "\t===== \n".
"\t[Enter] -- lagi [Ctrl+c] -- keluar";
print "\n\n\n\n\n\t ";
<STDIN>;
system("perl terbilang.pl");

```