

Nama : Linda Damayanti

NIM : 191524017

D4 Teknik Informatika

6. Password Validation

Create the regular expression in PERL to validate password text that does the following:

- No white-space character
- Minimum length of 10 characters
- Make sure at least :
 - One uppercase letter
 - One lowercase letter
 - One symbolic letter (non alpha)
 - One numeric letter

Source Code :

```
1 use strict;
2 print "Input your password : ";
3 my $password = <STDIN>;
4
5 if ($password =~ m/^(?!.*[ ])(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#\$%\^&\-_]*).-{10,}$/) {
6     print "valid\n";
7 } else {
8     print "invalid\n";
9 }
```

Output :

```
E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 6.pl
Input your password : 123lin$lin
valid

E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 6.pl
Input your password : 123lin_lin
invalid

E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 6.pl
Input your password : 123linlin
invalid

E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 6.pl
Input your password : linlin&
invalid

E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 6.pl
Input your password : 123456
invalid

E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 6.pl
Input your password : 123lin_
invalid

E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 6.pl
Input your password : 123lin_linda
valid
```

Analisis :

Pada if statement memakai regex

- `^(?!.*[])`, agar tidak ada white space.
- `(?=.*\d)`, untuk numerik letter

Nama : Linda Damayanti

NIM : 191524017

D4 Teknik Informatika

- `(?=.*[a-z])(?=.*[A-Z])`, untuk kondisi uppercase dan lowercase.
- `(?=.*[!@#\$%\^&_\-*])`, untuk pengecekan memiliki karakter selain alfabet dan angka.
- `{10,}`, untuk minimal 10 karakter.

Saat diinputkan, array akan diperiksa oleh if statement yang di dalamnya telah diberi peraturan-peraturan yang di buat dalam bentuk regex. Jika memenuhi syarat akan menampilkan "valid", jika sebaliknya akan menampilkan "invalid".

7. Domain Web Validation

Create the regular expression to validate URL (domain), consist of (.com, .id, .net), for example :

- `www.google.com` → valid
- `google.com` → invalid
- `www.google.my` → invalid
- `yahoo.com` → invalid
- `www.yahoo.com` → valid
- `www.abc123.id` → valid
- `www.abc123.sg` → invalid
- `www.detik.net` → valid

Source Code:

```
1 use strict;
2 print "Input domain : ";
3 my $dom = <STDIN>;
4
5 if ($dom =~ m/^(.+[a-z0-9])\.(.+[a-z0-9])\.(com|id|net)$/ ) {
6     print "valid\n";
7 } else {
8     print "invalid\n";
9 }
```

Output:

```
E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 7.pl
Input domain : www.google.com
valid

E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 7.pl
Input domain : www.google.my
invalid

E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 7.pl
Input domain : google.com
invalid

E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 7.pl
Input domain : www.kompas123.com
valid

E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 7.pl
Input domain : www.polban.id
valid

E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 7.pl
Input domain : www.polban.sg
invalid

E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 7.pl
Input domain : www.polman.net
valid
```

Nama : Linda Damayanti
NIM : 191524017
D4 Teknik Informatika

Analisis:

Pada if statement memakai regex :

- `(.[a-z0-9])` , untuk huruf a-z dan angka 0-9.
- `(com|id|net|)` , untuk akhiran .com, .id dan .net

Pemisah “\.” Pada regex menandakan jika inputan akan di periksa pada sub sub bagian sebelum dan sesudah titik tersebut sesuai dengan fungsi regexnya.

8. Serial Number Matching

Create the regular expression to validate serial number below :

22-Ab627-0360XY → valid

50-Yz6AA-076cUg → valid

- Format : XX-XXXXX-XXXXXX
- Must be exactly 15 characters
- Must only contains number for first 2 digit
- Allowed to use alpha numerics for the rest of digit
- Allowed to use uppercase or lowercase letters

Source Code:

```
1  use strict;
2  print "Input your serial number (example : 22-Ab627-0360XY) :";
3
4  my $serialnum = <STDIN>;
5
6  if ($serialnum =~ m/^\d{2}-([\w]{5})-([\w]{6})$/) {
7      print "valid\n";
8  } else {
9      print "invalid\n";
10 }
```

Output:

```
E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 8.pl
Input your serial number (example : 22-Ab627-0360XY) :30-Ld305-2010GH
valid

E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 8.pl
Input your serial number (example : 22-Ab627-0360XY) :11-234li-2344li
invalid
```

Analisis:

Pada if statement memakai regex :

- `(\d{2})`, untuk hanya angka berjumlah 2 karakter.
- `([\w]{5})`, untuk alphanumeric berjumlah 5 karakter.
- `([\w]{6})`, untuk alphanumeric berjumlah 6 karakter.

Dalam fungsi nya pun terdapat pemisah berupa “-” dimana nanti pada inputan akan di periksa jika terdapat “-” maka sebagai section baru (pemeriksaan ke regex berikutnya).

9. Sorting a List

Nama : Linda Damayanti

NIM : 191524017

D4 Teknik Informatika

Create the Subroutine to Sorting an arrays , with criteria:

Letter appear first before number

Uppercase appear first before lowercase

Ex:

Input("Alamat","aku","Alamat04","Handphone","03","02","z9");

Output: Alamat Alamat04 Handphone aku z9 02 03

Source code:

```
1  use strict;
2
3  my @arr = ("Alamat", "aku", "Alamat04", "Handphone", "03", "02", "z9");
4  my @result = sorting(@arr);
5  print "@result";
6
7  sub sorting {
8      my @arr;
9      my @numb;
10     foreach my $word (@_) {
11         if ($word =~ m/^[0-9]/){
12             push(@numb,$word);
13         }
14         else{
15             push(@arr,$word);
16         }
17     }
18     @arr=sort(@arr);
19     @numb=sort(@numb);
20     push(@arr,@numb);
21     return @arr;
22 }
```

Output:

```
E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 9.pl
Alamat Alamat04 Handphone aku z9 02 03
```

Analisis:

Pada sorting dibuat dua variabel. Dimana `@arr` untuk array yang depannya berupa huruf dan `@numb` untuk yang depannya berupa angka. Pada sort di perl sudah default mengurutkan huruf kapital lalu ke non kapital, tapi angkanya di depan (dimana berlawanan dengan keinginan), maka agar angkanya ke belakang dengan cara memisahkan `@arr` kedepan dan `@numb` ke belakang. Lalu, di sorting dimana uppercase terlebih dahulu lalu lowercase, lalu di susul dengan yang ada angkanya pada

```
@arr=sort(@arr);
```

```
@numb=sort(@numb);
```

```
Push(@arr, @numb);
```

10. Add Two Arrays

Create the Subroutine to add elements of two arrays and print the result

input

```
@a = (1, 2, 3);
```

```
@b = (4, 5, 6);
```

Output (5, 7, 9)

Source code:

Nama : Linda Damayanti

NIM : 191524017

D4 Teknik Informatika

```
1  use strict;
2
3  my @a = (1, 2, 3);
4  my @b = (4, 5, 6);
5
6  print (join(" ", add2Array(@a,@b)), "\n");
7
8  sub add2Array{
9      my $length = @_;
10
11      my @a = @_[0 .. ($length/2) - 1];
12      my @b = @_[length/2 .. $length - 1];
13
14      my @result;
15
16      for (my $i=0; $i<$length/2; $i++){
17          push(@result, @a[$i]+@b[$i]);
18      }
19
20      return @result;
21
22 }
```

Output:

```
E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 10.pl
5 7 9
```

Analisis:

Karena arraynya sama maka caranya ialah panjang array di bagi 2 lalu dikurangi 1, karena arraynya jumlah 2 arraynya pasti genap. Yang pertama, dari 0 sampai panjang array dibagi 2 lalu di kurang satu (karena dari 0 harus dikurangi 1). Yang kedua, dari panjang array di bagi 2 sampai panjang array nya di kurang 1. Pada *@result* di tambahkan array *@a* index ke *i* dengan array *@b* index ke *i*. Lalu dipush , direturn hasilnya diprint.

11. Returned Value From SubRoutine

Create the Subroutine to calculate the average of elements in arrays and print the result

input:

```
$a = calc_avg(11, 20, 2, 8, 3);
```

```
$a = calc_average(1,2,3);
```

Source code :

Nama : Linda Damayanti

NIM : 191524017

D4 Teknik Informatika

```
1  use strict;
2
3  my @a = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
4
5  print (calc_average(@a), "\n");
6
7  sub calc_average{
8      my $lenght = @_;
9
10     my $sum = 0;
11
12     foreach my $num (@_){
13         $sum += $num;
14     }
15
16     return $sum / $lenght;
17 }
```

Output :

```
E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 11.pl
5.5
```

Analisis:

Value pada `@a` di tambahkan lalu di bagi dengan jumlah nya (`@lenght`)

12. Max – Min

Create two subroutines `min()` and `max()` which accept an array as input and calculate the minimum and maximum numeric value of their arguments respectively

Source code:

```
1  use strict;
2
3  my @a = (30, 17, 77);
4
5  print("Minimum : ", min(@a), "\n");
6  print("Maximum : ", max(@a), "\n");
7
8  sub max{
9      my $result = @_[0];
10
11     foreach my $num (@_){
12         if ($num > $result){
13             $result = $num;
14         }
15     }
16
17     return $result;
18 }
19
20 sub min{
21     my $result = @_[0];
22
23     foreach my $num (@_){
24         if ($num < $result){
25             $result = $num;
26         }
27     }
28
29     return $result;
30 }
```

Output:

Nama : Linda Damayanti

NIM : 191524017

D4 Teknik Informatika

```
E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 12.pl
Minimum : 17
Maximum : 77
```

Analisis:

Perbandingan value dari subroutine pada max dan min. Ketika num > dari result maka num akan di jadikan result dan nanti akan di return menjadi maksimum, begitupula dengan minimum, dimana ketika num < dari result maka num akan dijadikan result dan nanti akan di return menjadi minimum.

13. Word And Character Count

Create the Subroutine to calculate the total of Word in line and count the character and print the result

Input = JTK Polban adalah sekolah vokasi Output Word = 5 Character = 28.

Source code:

```
1  use strict;
2
3  print "Input the sentence : ";
4  my $str = <STDIN>;
5
6  countInsentence($str);
7
8  sub countInsentence{
9      my @arr = split(' ', @_);
10
11      my $words = @arr;
12
13      my $char = 0;
14      foreach my $word (@arr){
15          $char += length($word);
16      }
17
18      print ("Word count : ", $words, "\n", "Char count: ", $char);
19  }
```

Output:

```
E:\Semester 3\Prinsip Pemrograman\Praktek\Perl\regex>perl 13.pl
Input the sentence : alhamdulillah bisa rebahan bentar
Word count : 4
Char count: 30
```

Analisis:

Lakukan split dari index ke 0. Lalu, dimasukkan ke words, lalu array di masukkan ke variabel baru word, ketika array dimasukkan ke word di hitung panjangnya berapa nanti masuk ke dalam char.