

Laporan Ujian Tengah Semester Mata Kuliah
Prinsip Bahasa Pemrograman
(Praktek)
UTS

Oleh:
Roy Aziz Barera
221524030



Program Studi D4 Teknik Informatika
Jurusan Teknik Komputer dan Informatika
Politeknik Negeri Bandung
2023

No. 1. Perl

Referensi: <https://perldoc.perl.org/>

```
#!/usr/bin/perl

use strict;
use warnings;

my $tag_html = '<div class="content">Visi DIV Teknik Informatika Menjadi Program
Studi unggulanpada jenjang pendidikan diploma IV di bidang Teknik Informatika
sehingga mampu menghasilkan tenaga profesional pengembang perangkat lunak yang
kompeten, memiliki semangat terus berkembang, bermoral, dan berjiwa wirausaha.
</div>';

if ($tag_html =~ /<div class="content">(.*?)<\div>/s) {
    print $1;
} else {
    print "Tag HTML dengan class 'content' tidak ditemukan.";
}
```

```
PS C:\Users\hp\Desktop\Prinsip Pemrograman\uts> perl no1.pl
Visi DIV Teknik Informatika Menjadi Program Studi unggulanpada jenjang pendidikan diploma
IV di bidang Teknik Informatika sehingga mampu menghasilkan tenaga profesional pengembang
perangkat lunak yang kompeten, memiliki semangat terus berkembang, bermoral, dan berjiwa w
irusaha.
PS C:\Users\hp\Desktop\Prinsip Pemrograman\uts>
```

- **#!/usr/bin/perl:** Ini adalah shebang line yang mengindikasikan bahwa ini adalah script Perl.
- **use strict;** dan **use warnings;**: Dua pernyataan ini digunakan untuk mengaktifkan mode ketat dan mode peringatan pada eksekusi program Perl. Mode ketat (strict) memungkinkan Perl untuk memeriksa dan memberikan pesan kesalahan jika ada penggunaan variabel yang tidak didefinisikan sebelumnya, sedangkan mode peringatan (warnings) memberikan peringatan tentang potensi masalah dalam kode.
- **my \$tag_html = ...:** Variabel \$tag_html digunakan untuk menyimpan string HTML yang akan diuji. Isinya adalah tag HTML dengan class "content" yang memiliki teks di dalamnya.
- **if (\$tag_html =~ /<div class="content">(.*?)<\div>/s) { ... }:** Ini adalah pernyataan kondisional yang menguji apakah string \$tag_html cocok dengan pola yang diberikan menggunakan regex.
- **/<div class="content">(.*?)<\div>/s** adalah ekspresi regex yang mencocokkan tag HTML dengan class "content" dan mengekstrak semua teks di dalamnya.
- **(.*?)** adalah bagian yang akan mengekstrak teks dalam tanda kurung.
- **/s** adalah modifikasi regex yang mengizinkan pencocokan melintasi baris (multiline).
- **print \$1;** Jika ekspresi regex cocok, maka \$1 akan berisi teks yang ada di dalam tag HTML "content", dan ini akan dicetak.
- **else { ... }:** Jika tidak ada tag HTML "content" yang ditemukan dalam string, maka program akan mencetak pesan bahwa "Tag HTML dengan class 'content' tidak ditemukan."

No. 2. Perl

Referensi: Tugas Minggu ke 4 check_password

```
#!/usr/bin/perl

use strict;
use warnings;
use Term::ANSIColor;

sub is_valid_password {
    my ($password) = @_ ;

    if ($password =~ /\s/) {
        print colored("Tidak boleh mengandung spasi.\n\n", 'red');
        return 1;
    }
    elsif ($password !~ /^.{15,}$/) {
        print colored("Minimal 15 karakter.\n\n", 'red');
        return 2;
    }
    elsif ($password !~ /[A-Z]/) {
        print colored("Harus mengandung setidaknya satu huruf kapital.\n\n",
'red');
        return 3;
    }
    elsif ($password !~ /[a-z]/) {
        print colored("Harus mengandung setidaknya satu huruf kecil.\n\n",
'red');
        return 4;
    }
    elsif ($password !~ /\d/) {
        print colored("Harus mengandung setidaknya satu angka.\n\n", 'red');
        return 5;
    }
    elsif ($password !~ /\W/) {
        print colored("Harus mengandung setidaknya satu karakter simbolik.\n\n",
'red');
        return 6;
    }
    else {
        print colored("Kata sandi valid.\n\n", 'green');
        return 0;
    }
}

my $valid = 1;
while ($valid != 0) {
    print colored("Masukan kata sandi anda: ", 'cyan');
    my $password = <STDIN>;
    chomp($password);

    $valid = is_valid_password($password);

    if ($valid != 0) {
        print colored("Kata sandi tidak valid.\n\n", 'red');
    }
}
```

```

PS C:\Users\hp\Desktop\Prinsip Pemrograman\uts> perl no2.pl
Masukan kata sandi anda: abc
Minimal 15 karakter.

Kata sandi tidak valid.

Masukan kata sandi anda: abcde1234567890
Harus mengandung setidaknya satu huruf kapital.

Kata sandi tidak valid.

Masukan kata sandi anda: abcde1234567890A
Harus mengandung setidaknya satu karakter simbolik.

Kata sandi tidak valid.

Masukan kata sandi anda: abcde1234567890A@
Kata sandi valid.

PS C:\Users\hp\Desktop\Prinsip Pemrograman\uts> █

```

- **#!/usr/bin/perl:** Ini adalah shebang line yang mengindikasikan bahwa ini adalah script Perl.
- **use strict;** dan **use warnings;**: Dua pernyataan ini digunakan untuk mengaktifkan mode ketat dan mode peringatan pada eksekusi program Perl. Mode ketat (strict) memungkinkan Perl untuk memeriksa dan memberikan pesan kesalahan jika ada penggunaan variabel yang tidak didefinisikan sebelumnya, sedangkan mode peringatan (warnings) memberikan peringatan tentang potensi masalah dalam kode.
- **use Term::ANSIColor;**: Ini adalah perintah untuk mengimpor modul Term::ANSIColor, yang digunakan untuk memberi warna teks yang akan ditampilkan di terminal.
- **sub is_valid_password { ... }:** Ini adalah subrutin (fungsi) yang digunakan untuk memeriksa apakah kata sandi (password) yang diberikan memenuhi kriteria validasi tertentu. Program melakukan serangkaian pemeriksaan dan memberikan pesan warna yang sesuai dengan hasil validasi. Hasil validasi (angka) dikembalikan oleh subrutin.
- **/(\s)/:** Ini adalah ekspresi regex yang mencocokkan spasi (whitespace). Jika ada spasi dalam kata sandi, program akan menampilkan pesan bahwa kata sandi tidak boleh mengandung spasi.
- **/^{15,}\$/:** Ini adalah ekspresi regex yang mencocokkan kata sandi yang memiliki panjang minimal 15 karakter. Jika kata sandi kurang dari 15 karakter, program akan menampilkan pesan bahwa kata sandi harus memiliki setidaknya 15 karakter.
- **/[A-Z]/:** Ini adalah ekspresi regex yang mencocokkan setidaknya satu huruf kapital (uppercase) dalam kata sandi. Jika tidak ada huruf kapital dalam kata sandi, program akan menampilkan pesan bahwa kata sandi harus mengandung setidaknya satu huruf kapital.

- `/[a-z]/`: Ini adalah ekspresi regex yang mencocokkan setidaknya satu huruf kecil (lowercase) dalam kata sandi. Jika tidak ada huruf kecil dalam kata sandi, program akan menampilkan pesan bahwa kata sandi harus mengandung setidaknya satu huruf kecil.
- `/(\d)/`: Ini adalah ekspresi regex yang mencocokkan setidaknya satu angka dalam kata sandi. Jika tidak ada angka dalam kata sandi, program akan menampilkan pesan bahwa kata sandi harus mengandung setidaknya satu angka.
- `/(\W)/`: Ini adalah ekspresi regex yang mencocokkan setidaknya satu karakter simbolik (non-alphanumeric) dalam kata sandi. Jika tidak ada karakter simbolik dalam kata sandi, program akan menampilkan pesan bahwa kata sandi harus mengandung setidaknya satu karakter simbolik.

Catatan: Environment Haskell & GHCi saya terganggu jadi saya cari alternatif lain untuk menggunakan compiler online.

Referensi: <https://www.haskell.org/documentation/>

No. 3. Haskell

```
calculateTotal :: Int -> Int -> Int
calculateTotal porsi harga
  | porsi <= 4 = harga * porsi
  | porsi > 4  && porsi < 8 = (harga - 800) * porsi
  | porsi > 8 = (harga - 1000) * porsi

main :: IO ()
main = do
  let hargaPerPorsi = 15000
  let totalHarga = calculateTotal 7 hargaPerPorsi
  putStrLn ("Total yang harus dibayar: Rp. " ++ show totalHarga)
```

```
1 calculateTotal :: Int -> Int -> Int
2 calculateTotal porsi harga
3   | porsi <= 4 = harga * porsi
4   | porsi > 4  && porsi < 8 = (harga - 800) * porsi
5   | porsi > 8 = (harga - 1000) * porsi
6
7 main :: IO ()
8 main = do
9   let hargaPerPorsi = 15000
10  let totalHarga = calculateTotal 7 hargaPerPorsi
11  putStrLn ("Total yang harus dibayar: Rp. " ++ show totalHarga)
12
```

STDIN

Input for the program (Optional)

Output:

Total yang harus dibayar: Rp. 99400

- **calculateTotal :: Int -> Int -> Int:** Ini adalah deklarasi tipe untuk fungsi calculateTotal. Fungsi ini menerima dua argumen bertipe Int, yaitu jumlah porsi (porsi) dan harga per porsi (harga), dan mengembalikan hasil berupa Int.

- **Fungsi calculateTotal** memiliki tiga kasus yang digerakkan oleh pola (pattern matching):
 - **| porsi <= 4 = harga * porsi:** Jika jumlah porsi kurang dari atau sama dengan 4, maka total harga adalah jumlah porsi (porsi) dikalikan dengan harga per porsi (harga).
 - **| porsi > 4 && porsi < 8 = (harga - 800) * porsi:** Jika jumlah porsi lebih dari 4 dan kurang dari 8, maka total harga adalah jumlah porsi (porsi) dikalikan dengan harga per porsi (harga) yang sudah dikurangi dengan 800 (potongan harga).
 - **| porsi > 8 = (harga - 1000) * porsi:** Jika jumlah porsi lebih dari 8, maka total harga adalah jumlah porsi (porsi) dikalikan dengan harga per porsi (harga) yang sudah dikurangi dengan 1000 (potongan harga).
- **main :: IO ():** Ini adalah deklarasi tipe untuk fungsi main. Fungsi main adalah titik masuk utama program.
- Dalam fungsi main:
 - `let hargaPerPorsi = 15000` mendefinisikan harga per porsi ayam goreng, yang dalam contoh ini diatur ke 15.000.
 - `let totalHarga = calculateTotal 7 hargaPerPorsi` memanggil fungsi `calculateTotal` dengan argumen 7 (jumlah porsi) dan harga per porsi yang telah didefinisikan sebelumnya, dan hasilnya disimpan dalam variabel `totalHarga`.
 - `putStrLn ("Total yang harus dibayar: Rp. " ++ show totalHarga)` mencetak total harga yang harus dibayar ke layar dengan menggunakan `putStrLn`, termasuk pesan dan nilai dari `totalHarga`.

Note: Alasan saya menggunakan `main :: IO ()` adalah karena ghci saya bermasalah jadi saya hanya bisa mengakses fungsi haskell langsung melalui source code nya, sehingga saya panggil fungsi nya yang memiliki parameter 7 dan 15000.

No. 4. Haskell

```
import Data.Char (isAlpha)

extractKodeWilayah :: String -> String
extractKodeWilayah platNomor = takeWhile isAlpha platNomor

main :: IO ()
main = do
    let platNomor = "D1234HT"
    let kodeWilayah = extractKodeWilayah platNomor
    putStrLn ("Plat Nomor Anda: " ++ platNomor)
    putStrLn ("Kode Wilayah Plat Nomor Anda: " ++ kodeWilayah)
```

```
1 import Data.Char (isAlpha)
2
3 extractKodeWilayah :: String -> String
4 extractKodeWilayah platNomor = takeWhile isAlpha platNomor
5
6 main :: IO ()
7 main = do
8     let platNomor = "D1234HT"
9     let kodeWilayah = extractKodeWilayah platNomor
10    putStrLn ("Plat Nomor Anda: " ++ platNomor)
11    putStrLn ("Kode Wilayah Plat Nomor Anda: " ++ kodeWilayah)
12
```

STDIN

Input for the program (Optional)

Output:

Plat Nomor Anda: D1234HT

Kode Wilayah Plat Nomor Anda: D

- **import Data.Char (isAlpha):** Ini adalah pernyataan untuk mengimpor modul Data.Char dan mengambil hanya fungsi isAlpha dari modul tersebut. isAlpha digunakan untuk memeriksa apakah karakter adalah huruf alfabet (baik huruf besar maupun kecil).
- **extractKodeWilayah :: String -> String:** Ini adalah deklarasi tipe untuk fungsi extractKodeWilayah. Fungsi ini menerima satu argumen bertipe String, yaitu platNomor, dan mengembalikan hasil berupa String.
- Dalam fungsi extractKodeWilayah:

- **takeWhile isAlpha platNomor:** menggunakan takeWhile untuk mengambil karakter dari awal platNomor selama karakter tersebut adalah huruf alfabet. Ini berarti kode wilayah akan terdiri dari karakter huruf alfabet di awal plat nomor.
- **main :: IO ():** Ini adalah deklarasi tipe untuk fungsi main. Fungsi main adalah titik masuk utama program.
- Dalam fungsi main:
 - **let platNomor = "D1234HT"** mendefinisikan platNomor sebagai string dengan nilai "D1234HT". Ini adalah plat nomor contoh yang akan diuji.
 - **let kodeWilayah = extractKodeWilayah platNomor** memanggil fungsi extractKodeWilayah dengan argumen platNomor dan hasilnya disimpan dalam variabel kodeWilayah.
 - **putStrLn ("Plat Nomor Anda: " ++ platNomor)** mencetak plat nomor ke layar.
 - **putStrLn ("Kode Wilayah Plat Nomor Anda: " ++ kodeWilayah)** mencetak kode wilayah yang telah diekstrak dari plat nomor.