	<b>FAKULTAS TEKNIK</b> <b>UNIVERSITAS NEGERI YOGYAKARTA</b>			
	<b>LAB SHEET BAHASA PEMROGRAMAN</b>			
	Semester 3	Perulangan <i>for</i>		4 X 50 menit
	No. LST/EKA/EKA255/05	Revisi : 00	Tgl : 8 Sept 2014	Hal 1 dari 5

1. **Kompetensi**

- a. Menjelaskan proses pengulangan menggunakan pernyataan *for*
- b. Menjelaskan tentang variasi pernyataan *for*
- c. Menjelaskan tentang pernyataan *for* dengan menentukan jumlah langkah

2. **Sub Kompetensi**

- a. Memahami struktur dan bentuk pernyataan perulangan *for*
- b. Memahami penggunaan variabel dalam struktur perulangan *for*

3. **Dasar Teori**

Mengulang suatu proses merupakan tindakan yang banyak dijumpai dalam pemrograman. Pada semua bahasa pemrograman, pengulangan proses ditangani dengan suatu mekanisme yang disebut *loop*. Dengan menggunakan *loop*, suatu proses yang berulang misalnya menampilkan tulisan yang sama seratus kali pada layar dapat diimplementasikan dengan kode program yang pendek.

Pada pemrograman proses perulangan dapat dibagi menjadi 2 bagian utama yaitu:

- a. Perulangan yang sudah di ketahui jumlah perulangannya sebelum perulangan tersebut di lakukan.
- b. Perulangan yang belum di ketahui jumlah perulangannya sebelum perulangan tersebut di lakukan. Dalam hal ini dapat dibagi menjadi dua bagian yaitu:
  - 1) kondisi perulangan diperiksa diawal perulangan.
  - 2) kondisi perulangan diperiksa diakhir perulangan.

Untuk kasus 1 seharusnya menggunakan perulangan dengan pernyataan *for* dan akan dipelajari pada praktikum ini. Sedangkan pada kasus 2a dan 2b akan dibahas pada praktikum 10.

Bentuk pernyataan *for* :

```
for (ungkapan1; ungkapan2; ungkapan3)
    pernyataan;
```

Keterangan:

Ungkapan1 : digunakan untuk memberikan inisialisasi terhadap variabel pengendali loop.

Ungkapan2 : dipakai sebagai kondisi untuk keluar dari loop.

Ungkapan3 : dipakai sebagai pengatur kenaikan nilai variabel pengendali loop.

Ketiga ungkapan dalam *for* tersebut harus dipisahkan dengan tanda titik koma (;). Dalam hal ini pernyataan bisa berupa pernyataan tunggal maupun jamak. Jika pernyataannya berbentuk jamak, maka pernyataan-pernyataan tersebut harus

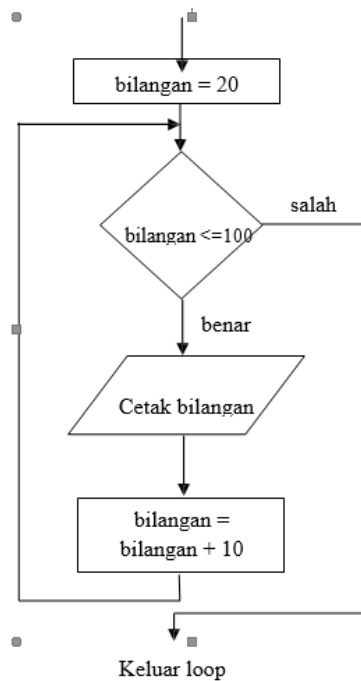
Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
---------------	--	------------------



diletakkan di antara kurung kurawal buka ({} dan kurung kurawal tutup (}), sehingga formatnya menjadi :

```
for (ungkapan1; ungkapan2; ungkapan3)
{
    pernyataan;
    pernyataan;
    .
    .
    .
}
```

Pada Gambar 1 ditunjukkan flowchart looping untuk menulis bilangan 20, 30, ..., 100



Gambar 1. Diagram alir *for* untuk menulis 20,30,...,100

```
for(bilangan = 20; bilangan <= 100; bilangan += 10)
    printf("%d\n", bilangan);
```

Pada program di atas, kenaikan terhadap variabel pengendali *loop* sebesar 10 (positif), yang dinyatakan dengan ungkapan

```
bilangan += 10
```

Dibuat oleh :

Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta

Diperiksa oleh :



Pada contoh yang melibatkan pernyataan *for* di atas, kenaikan variabel pengendali *loop* berupa nilai positif. Sebenarnya kenaikan terhadap variabel pengendali *loop* bisa diatur bernilai negatif.

```
for (bilangan = 60; bilangan >= 10; bilangan -= 10)
    printf("%d\n", bilangan);
```

Kadang-kadang dijumpai adanya pernyataan *for* yang tidak mengandung bagian ungkapan yang lengkap (beberapa ungkapan dikosongkan). Dengan cara ini, pernyataan

```
for (bilangan = 20; bilangan <= 100; bilangan += 10)
    printf(, bilangan);
```

Dapat Ditulis menjadi

```
bilangan = 20;          /* inisialisasi di luar for */
for ( ; bilangan <= 100; )
{
    Printf ("%d\n", bilangan);
    Bilangan +=10;
}
```

Ungkapan kosong

Pengosongan ini juga dilakukan pada ungkapan yang biasa dipakai untuk menaikkan nilai variabel pengendali *loop*. Sebagai gantinya, di dalam tubuh *loop* diberikan pernyataan untuk menaikkan nilai variabel pengendali *loop*, yaitu berupa :

```
bilangan += 10;
```


Ungkapan yang tidak dihilangkan berupa `bilangan <=100`. Ungkapan ini tetap disertakan karena dipakai sebagai kondisi untuk keluar dari *loop*.

Sesungguhnya ungkapan yang dipakai sebagai kondisi keluar dari *loop* juga bisa dihilangkan, sehingga bentuknya menjadi

Dibuat oleh :

Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta

Diperiksa oleh :


	<b>FAKULTAS TEKNIK</b> <b>UNIVERSITAS NEGERI YOGYAKARTA</b>			
	<b>LAB SHEET BAHASA PEMROGRAMAN</b>			
	Semester 3	Perulangan <i>for</i>		4 X 50 menit
	No. LST/EKA/EKA255/05	Revisi : 00	Tgl : 8 Sept 2014	Hal 4 dari 5

for ( ; ; )  
 pernyataan

Suatu pertanyaan mungkin timbul “Lalu bagaimana caranya kalau ingin keluar dari loop pada bentuk di atas?”. Caranya adalah dengan menggunakan pernyataan yang dirancang khusus untuk keluar dari loop. Mengenai hal ini akan dibahas pada praktikum selanjutnya

4. **Alat/Instrument/Aparatus/Bahan**
  - a. Komputer/PC/notebook
  - b. Software Pemrograman Bahasa C Dev-CPP
  
5. **Keselamatan Kerja**
  - a. Gunakan Komputer/PC/Notebook sesuai dengan peruntukannya
  - b. Periksa sistem kelistrikan / Sumber daya
  
6. **Langkah Kerja**
  - a. Siapkan peralatan berupa PC/Komputer yang sudah terinstall software Dev-CPP
  - b. Siapkan software Dev-CPP sebelum memprogram
  - c. Buatlah algoritma dari program
  - d. Buatlah flowchart dari program
  - e. Menulis kode program dalam software pemrograman Dev-CPP
  - f. Simpan kode program
  - g. Compile dan Jalankan program
  
7. **Bahan Diskusi**
  - a. Gunakan loop *for* untuk membuat program sebagai berikut:  
 input : n  
 output : 1 3 4 5 ... m ( m = bilangan ganjil ke n)
  - b. Hitunglah bilangan triangular dari masukan pengguna, yang dibaca dari keyboard dengan menggunakan *scanf()*. Bilangan triangular adalah penjumlahan dari bilangan masukan dengan seluruh bilangan sebelumnya.  
 Contoh Input : 7  
 Output : 7 + 6 + 5 + 4 + 3 + 2 + 1 = 28
  - c. Gunakan loop *for* dengan kenaikan variabel negatif untuk menampilkan seluruh karakter dari Z sampai dengan A dalam baris-baris yang terpisah.
  - d. Gunakan loop *for* untuk membuat program sebagai berikut:  
 input : n

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
---------------	--	------------------

	<b>FAKULTAS TEKNIK</b> <b>UNIVERSITAS NEGERI YOGYAKARTA</b>			
	<b>LAB SHEET BAHASA PEMROGRAMAN</b>			
	Semester 3	Perulangan <i>for</i>		4 X 50 menit
	No. LST/EKA/EKA255/05	Revisi : 00	Tgl : 8 Sept 2014	Hal 5 dari 5

output : 1 -2 3 -4 5 -6 7 -8 ... n

- e. Gunakan loop *for* untuk mendefinisikan apakah sebuah bilangan adalah bilangan prima atau bukan

input : 27

output : Bilangan adalah bilangan prima

- f. Dengan menggunakan looping dan *switch case* atau *else if* buatlah program dalam C untuk menghitung Indeks Prestasi Semester seorang mahasiswa, dimana yang diinputkan adalah nilai huruf dari 5 mata kuliah yang diikutinya dan jumlah jam mata kuliah tsb.

Dimana konversi nilai huruf ke angka untuk menghitung IPS adalah sebagai berikut:

A -> 4, B->3, C->2, D->1, E->0 dan rumus IPS = jumlah (nilai \* jam)/jumlah jam keseluruhan

Contoh : Input :

Nilai Mata Kuliah 1 : A    jumlah jam : 2

Nilai Mata Kuliah 2 : C    jumlah jam : 2

Nilai Mata Kuliah 3 : B    jumlah jam : 3

Nilai Mata Kuliah 4 : A    jumlah jam : 3

Nilai Mata Kuliah 5 : C    jumlah jam : 3

Output:

Indeks Prestasi Semester : 3

Output di atas didapatkan dari :  $(4*2 + 2*2 + 3*3 + 4*3 + 2*3)/(2+2+3+3+3) = 39/13$

## 8. LAPORAN RESMI

- a. Untuk setiap listing program dari percobaan-percobaan di atas, ambil *capture* outputnya.
- b. Tuliskan kesimpulan dari percobaan yang telah anda lakukan

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
---------------	--	------------------