

**DIKTAT**

**ALGORITMA DAN PEMROGRAMAN II**

**OLEH :**  
**YUSUF RAMADHAN NASUTION, M.KOM**



**PROGRAM STUDI ILMU KOMPUTER**  
**FAKULTAS SAINS DAN TEKNOLOGI**  
**UIN SUMATERA UTARA MEDAN**

**2019**

## **KATA PENGANTAR**

Puji syukur ke hadirat Allah SWT, yang telah memberikan rahmat-Nya sehingga diktat perkuliahan Algoritma dan Pemrograman II ini dapat diselesaikan dengan sebaik-baiknya.

Dikat ini dibuat sebagai pedoman dalam melakukan kegiatan perkuliahan Algoritma dan Pemrograman II yang merupakan matakuliah wajib Prodi Ilmu Komputer Fakultas Sains Dan Teknologi UIN Sumatera Utara Medan. Diktat ini diharapkan dapat membantu mahasiswa/i dalam mempersiapkan dan melaksanakan perkuliahan dengan lebih baik, terarah, dan terencana. Penyusun menyakini bahwa dalam pembuatan Diktat Perkuliahan Algoritma dan Pemrograman II ini masih jauh dari sempurna. Oleh karena itu penyusun mengharapkan kritik dan saran yang membangun guna penyempurnaan diktat perkuliahan ini dimasa yang akan datang.

Akhir kata, penyusun mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung.

Medan, September 2019

Penyusun

**Yusuf Ramadhan Nasution, M.Kom**

## DAFTAR ISI

<b>KATA PENGANTAR .....</b>	<b>ii</b>
<b>DAFTAR ISI.....</b>	<b>iii</b>
<b>DAFTAR GAMBAR.....</b>	<b>v</b>
<b>DAFTAR TABEL .....</b>	<b>vii</b>
<b>MODUL 1 .....</b>	<b>1</b>
<b>MENGENAL BAHASA C .....</b>	<b>1</b>
1. Sejarah .....	1
2. Kelebihan dan Kekurangan Bahasa C .....	1
3. Mengetahui Editor Bahasa C .....	2
4. Penulisan Program Bahasa C .....	4
<b>MODUL 2 .....</b>	<b>7</b>
<b>STRUKTUR BAHASA C.....</b>	<b>7</b>
1. Konstanta .....	7
2. Tipe Data .....	7
3. Variabel .....	8
<b>MODUL 3 .....</b>	<b>11</b>
<b>ARITMATIKA DAN OPERATOR .....</b>	<b>11</b>
1. Operator Penugasan .....	11
2. Operator Aritmatika.....	11
3. Operator Hubungan (Perbandingan).....	11
<b>MODUL 4 .....</b>	<b>14</b>
<b>SCANF .....</b>	<b>14</b>
Percobaan 1 .....	14
<b>MODUL 5 .....</b>	<b>16</b>
<b>GETS() .....</b>	<b>16</b>
1. Pengertian GETS() .....	16

<b>MODUL 6</b> .....	<b>18</b>
<b>GETCHAR(), GETCH(), GETCHE()</b> .....	<b>18</b>
1. GETCHAR().....	18
2. GETCH() .....	18
3. GETCHE().....	18
<b>MODUL 7</b> .....	<b>22</b>
<b>IF</b> .....	<b>22</b>
1. IF.....	22
<b>MODUL 8</b> .....	<b>24</b>
<b>IF ELSE</b> .....	<b>24</b>
1. IF ELSE .....	24
<b>MODUL 9</b> .....	<b>26</b>
<b>ELSE IF</b> .....	<b>26</b>
<b>MODUL 10</b> .....	<b>27</b>
<b>SWITCH...CASE...DEFAULT</b> .....	<b>27</b>
<b>MODUL 11</b> .....	<b>29</b>
<b>WHILE</b> .....	<b>29</b>
<b>MODUL 12</b> .....	<b>30</b>
<b>DO...WHILE</b> .....	<b>30</b>
<b>MODUL 14</b> .....	<b>32</b>
<b>GO TO &amp; LABEL</b> .....	<b>32</b>
<b>DAFTAR PUSTAKA</b> .....	<b>33</b>

## DAFTAR GAMBAR

Gambar 1. Program Bahasa C yang baik.....	5
Gambar 2. Ouput Program Bahasa C yang baik.....	5
Gambar 3. Program Bahasa C yang kurang baik .....	5
Gambar 4. Output Program Bahasa C yang kurang baik.....	6
Gambar 5. Program Penggunaan Konstanta .....	9
Gambar 6. Ouput Program Penggunaan Konstanta .....	9
Gambar 7. Program Tipe Data dan Variabel.....	10
Gambar 8. Output Tipe Data dan Variabel .....	10
Gambar 9. Program Operator Aritmatika.....	12
Gambar 10. Ouput Program Operator Aritmatika.....	12
Gambar 11. Program Operator Perbandingan.....	13
Gambar 12. Output Program Perbandingan .....	13
Gambar 13. Program Penggunaan Scanf .....	15
Gambar 14. Output Program Penggunaan Scanf.....	15
Gambar 15. Program Penggunaan gets() .....	16
Gambar 16. Ouput Program Penggunaan gets() .....	17
Gambar 17. Program Penggunaan gets() .....	19
Gambar 18. Ouput Program Penggunaan gets() .....	19
Gambar 19. Program Penggunaan getche() .....	20
Gambar 20. Ouput Program Penggunaan getch() .....	20
Gambar 21. Program Penggunaan getchar() .....	21
Gambar 22. Output Program Penggunaan getchar() .....	21
Gambar 23. Program Penggunaan if .....	23
Gambar 24. Ouput Program Penggunaan if .....	23
Gambar 25. Program Penggunaan If Else.....	25
Gambar 26. Output Program Penggunaan If Else .....	25
Gambar 27. Program Penggunaan Else If.....	26
Gambar 28. Output Program Penggunaan Else If .....	26
Gambar 29. Program Penggunaan Switch..Case..Default .....	27
Gambar 30. Output Program Penggunaan Switch..Case..Default .....	28
Gambar 31. Program Penggunaan While.....	29

<b>Gambar 32. Output Program Penggunaan While .....</b>	<b>29</b>
<b>Gambar 33. Program Penggunaan Do..While.....</b>	<b>30</b>
<b>Gambar 34. Output Program Penggunaan While .....</b>	<b>30</b>
<b>Gambar 35. Output Program Penggunaan For .....</b>	<b>31</b>
<b>Gambar 36. Output Program Penggunaan For .....</b>	<b>31</b>
<b>Gambar 37. Program Penggunaan GoTo &amp; Label.....</b>	<b>32</b>
<b>Gambar 38. Output Program Penggunaan GoTo &amp; Label.....</b>	<b>32</b>

## DAFTAR TABEL

<b>Tabel 1. Tipe Data.....</b>	<b>8</b>
<b>Tabel 2. Operator Hubungan (Perbandingan).....</b>	<b>12</b>
<b>Tabel 3. Kode Penentu Format.....</b>	<b>14</b>

# MODUL 1

## MENGENAL BAHASA C

### 1. Sejarah

Selamat datang di dunia pemrograman, modul ini akan mempelajari bahasa C. Bahasa C merupakan bahasa komputer yang sangat singkat. Namanya tidak memiliki kepanjangan apapun, bahasa ini diciptakan oleh Dennis Ritchie sekitar tahun 1972. Hingga kini bahasa ini masih populer diberbagai platform; dari Windows sampai linux dan dari PC hingga mainframe.

Bahasa C merupakan perkembangan dari bahasa BCPL yang dikembangkan oleh Martin Richards pada tahun 1967. Selanjutnya bahasa ini memberikan ide kepada Ken Thompson yang kemudian mengembangkan bahasa yang disebut bahasa B pada tahun 1970. Perkembangan selanjutnya dari bahasa B adalah bahasa C oleh Dennis Ritchie sekitar tahun 1970-an di Bell Telephone Laboratories Inc. (sekarang adalah AT&T Bell Laboratories) (R. Munir, 2005 : 5<sup>1</sup>) . Bahasa C pertama kali digunakan di computer Digital Equipment Corporation PDP-11 yang menggunakan system operasi UNIX. Hingga saat ini penggunaan bahasa C telah merata di seluruh dunia. Hampir semua perguruan tinggi di dunia menjadikan bahasa C sebagai salah satu mata kuliah wajib. Selain itu, banyak bahasa pemrograman populer seperti PHP dan Java menggunakan sintaks dasar yang mirip bahasa C. Oleh karena itu, kita juga sangat perlu mempelajarinya.

### 2. Kelebihan dan Kekurangan Bahasa C

#### a. Kelebihan Bahasa C

- Bahasa C tersedia hampir di semua jenis computer.
- Kode bahasa C sifatnya adalah portable dan fleksibel untuk semua jenis computer.
- Bahasa C hanya menyediakan sedikit kata-kata kunci, hanya terdapat 32 kata kunci.
- Proses executable program bahasa C lebih cepat
- Dukungan pustaka yang banyak.
- C adalah bahasa yang terstruktur

---

<sup>1</sup> R. Munir, *Algoritma dan Pemrogram dalam Bahasa Pascal dan C*, Bandung: Informatika Bandung, 2005, Hal 5.



- Bahasa C termasuk bahasa tingkat menengah .

**b. Kelemahan Bahasa C**

- Banyaknya Operator serta fleksibilitas penulisan program kadang-kadang membingungkan pemakai.
- Bagi pemula pada umumnya akan kesulitan menggunakan pointer

**3. Mengenal Editor Bahasa C**

**a. Memulai Bahasa C**

- Buka Editor Bahasa C yang ada, seperti Borland C, Turbo C, dan sebagainya. Semua program yang ada di tutorial ini bisa dicoba Turbo C.

**b. Untuk mengkompilasi Program**

- Pilih menu Compile dengan menekan Alt + C
- Pilih Submenu Compile.
- Enter.
- Akan ditampilkan hasil kompilasi Program, tekan sembarang tombol.

**c. Untuk menjalankan Program**

- Pilih menu Run dengan menekan Alt + R
- Pilih submenu Run dan tekan Enter.

**d. Menu-menu dalam Turbo C**

File, terdiri dari:

- New, untuk memulai program baru
- Open, untuk mengambil atau membuka program
- Save, untuk menyimpan file/program
- Save as, untuk menyimpan file/program
- Save all, untuk menyimpan seluruh file/program
- Change dir, untuk mengubah directory
- Print, untuk mencetak program
- DOS Shell, untuk menuju ke DOS Shell
- Quit, untuk keluar dari Turbo C

Edit, terdiri dari:

- Undo, untuk membatalkan pengeditan terakhir
- Redo, untuk kembali ke pengeditan terakhir yang telah di undo.
- Cut, untuk memotong bagian tertentu dari program.
- Copy, untuk menduplikasi bagian program
- Paste
- Clear, untuk menghapus bagian tertentu dari program
- Copy example
- Show Clipboard

Search, terdiri dari:

- Find
- Replace
- Search again
- Previous error
- Next error
- Locate function.

Run, terdiri dari:

- Run., untuk menjalankan program
- Program reset
- Go to cursor
- Dst.

Compile, terdiri dari:

- Compile, untuk mengkompilasi program
- Make
- Link
- Build all, dst.

Debug, terdiri dari:

- Inspect
- Evaluate/modify
- Dst.

Project, terdiri dari:

- Open project
- Close project
- Dst.

Options, terdiri dari:

- Application
- Compiler
- Transfer
- dst.

Window, terdiri dari:

- Size/Move
- Zoom
- Tile
- Cascade
- Next
- Dst.

Help, terdiri dari:

- Contens
- Index
- Topic search
- Previous topic
- Dst.

#### **4. Penulisan Program Bahasa C**

- e. Program Bahasa C tidak mengenal aturan penulisan di kolom tertentu, jadi bisa dimulai dari kolom manapun. Namun demikian, untuk mempermudah pembacaan program dan untuk keperluan dokumentasi, sebaiknya penulisan bahasa C diatur sedemikian rupa sehingga mudah dan enak dibaca. (R. Munir, 2005 : 15<sup>2</sup>)

---

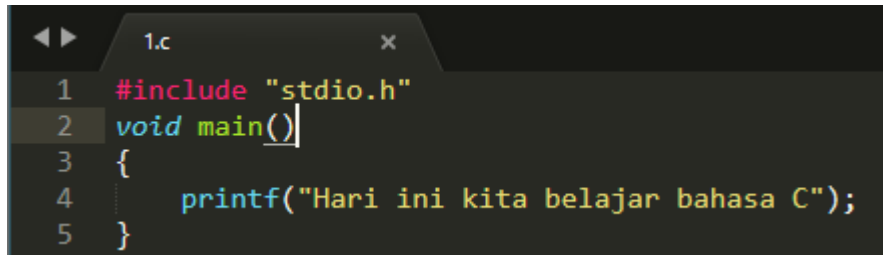
<sup>2</sup> R. Munir, *Algoritma dan Pemrogram dalam Bahasa Pascal dan C*, Bandung: Informatika Bandung, 2005, Hal 15.

f. Berikut ini contoh penulisan Program Bahasa C yang baik dan yang kurang baik:

### Percobaan 1

Penulisan Program Bahasa C yang baik

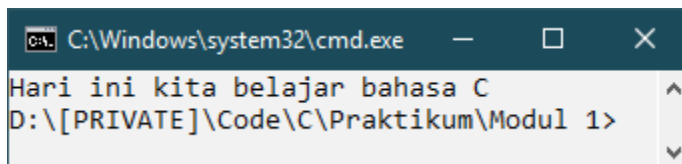
Program C:



```
1 #include "stdio.h"
2 void main()
3 {
4     printf("Hari ini kita belajar bahasa C");
5 }
```

Gambar 1. Program Bahasa C yang baik

Output:



```
C:\Windows\system32\cmd.exe
Hari ini kita belajar bahasa C
D:\[PRIVATE]\Code\C\Praktikum\Modul 1>
```

Gambar 2. Output Program Bahasa C yang baik

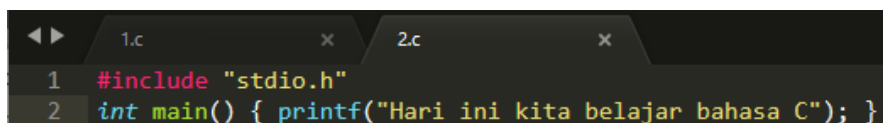
Pembahasan :

- Penggunaan praprosesor yang benar adalah #include "stdio.h".
- printf menghasilkan keluaran "Hari ini kita belajar Bahasa C".

### Percobaan 2

Penulisan Program Bahasa C yang kurang baik

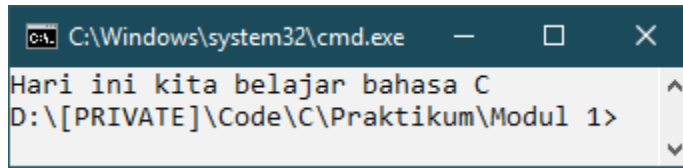
Program C:



```
1 #include "stdio.h"
2 int main() { printf("Hari ini kita belajar bahasa C"); }
```

Gambar 3. Program Bahasa C yang kurang baik

## Output:

A screenshot of a Windows command prompt window. The title bar shows 'C:\Windows\system32\cmd.exe'. The window contains two lines of text: 'Hari ini kita belajar bahasa C' followed by a new line and 'D:\[PRIVATE]\Code\C\Praktikum\Modul 1>'. There are scroll arrows on the right side of the window.

```
C:\Windows\system32\cmd.exe
Hari ini kita belajar bahasa C
D:\[PRIVATE]\Code\C\Praktikum\Modul 1>
```

Gambar 4. Output Program Bahasa C yang kurang baik

## Pembahasan :

Kedua Program di atas bila dijalankan akan menghasilkan hasil yang sama berupa tulisan “Hari ini kita belajar Bahasa C” di layar, namun dari segi penulisannya program yang pertama tampaknya lebih mudah dibaca dan lebih rapih dibanding dengan program yang kedua.

## MODUL 2

### STRUKTUR BAHASA C

#### 1. Konstanta

Konstanta merupakan suatu nilai yang tidak dapat diubah selama proses program berlangsung. Konstanta nilainya selalu tetap. Konstanta harus didefinisikan terlebih dahulu di awal program. Konstanta dapat bernilai integer, pecahan, karakter dan string. Contoh konstanta : 50; 13; 3.14; 4.50005; 'A'; 'Bahasa C' (Utami, Ema. dan Sukrisno, 2005 : 10<sup>3</sup>). Selain itu, bahasa C juga menyediakan beberapa karakter khusus yang disebut karakter escape, antara lain :

- \a : untuk bunyi bell (alert)
- \b : mundur satu spasi (backspace)
- \f : ganti halaman (form feed)
- \n : ganti baris baru (new line)
- \r : ke kolom pertama, baris yang sama (carriage return)
- \v : tabulasi vertical
- \0 : nilai kosong (null)
- \' : karakter petik tunggal
- \" : karakter petik ganda
- \\ : karakter garis miring

#### 2. Tipe Data

Tipe data merupakan bagian program yang paling penting karena tipe data mempengaruhi setiap instruksi yang akan dilaksanakan oleh computer. Misalnya saja 5 dibagi 2 bisa saja menghasilkan hasil yang berbeda tergantung tipe datanya. Jika 5 dan 2 bertipe integer maka akan menghasilkan nilai 2, namun jika keduanya bertipe float maka akan menghasilkan nilai 2.5000000 (Utami, Ema. dan Sukrisno, 2005 : 13<sup>4</sup>). Pemilihan tipe data yang tepat akan membuat proses operasi data menjadi lebih efisien dan efektif. Dalam bahasa C terdapat lima tipe data dasar, yaitu :

---

<sup>3</sup> Utami, Ema. dan Sukrisno. 2005. *10 Langkah Belajar Logika dan Algoritma Menggunakan Bahasa C dan C++ di GNU/Linux*. Andi Offset., Yogyakarta. Hal 10.

<sup>4</sup> Utami, Ema. dan Sukrisno. 2005. *10 Langkah Belajar Logika dan Algoritma Menggunakan Bahasa C dan C++ di GNU/Linux*. Andi Offset., Yogyakarta. Hal 13.

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

Tabel 1. Tipe Data

### 3. Variabel

Variabel adalah suatu pengenal (identifier) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program. Berbeda dengan konstanta yang nilainya selalu tetap, nilai dari suatu variable bisa diubah-ubah sesuai kebutuhan (R. Munir, 2005 : 20<sup>5</sup>). Nama dari suatu variable dapat ditentukan sendiri oleh pemrogram dengan aturan sebagai berikut :

- Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf. Bahasa C bersifat case-sensitive artinya huruf besar dan kecil dianggap berbeda. Jadi antara nim, NIM dan Nim dianggap berbeda.
- Tidak boleh mengandung spasi.
- Tidak boleh mengandung symbol-simbol khusus, kecuali garis bawah (underscore). Yang termasuk symbol khusus yang tidak diperbolehkan antara lain : \$, ?, %, #, !, &, \*, (, ), -, +, = dsb
- Panjangnya bebas, tetapi hanya 32 karakter pertama yang terpakai.

Contoh penamaan variabel yang benar:

- NIM, a, x, nama\_mhs, f3098, f4, nilai, budi, dsb.

Contoh penamaan variabel yang salah:

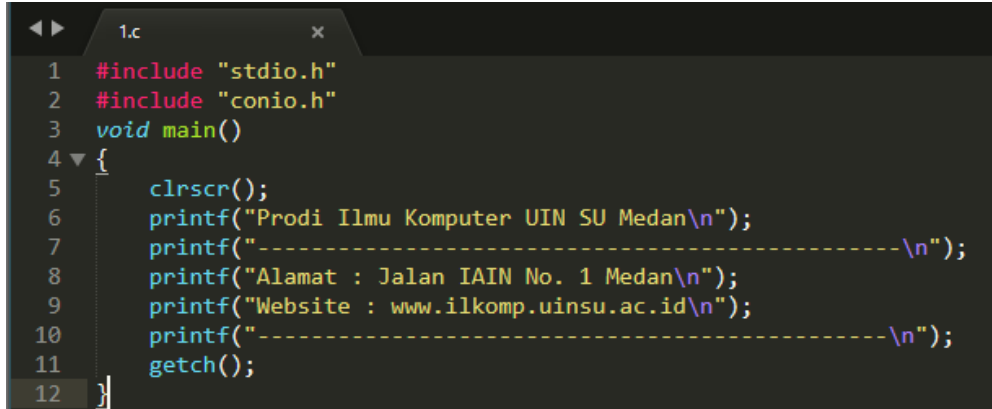
- %nilai\_mahasiswa, 80mahasiswa, rata-rata, ada spasi, penting!, dsb.

<sup>5</sup> R. Munir, Algoritma dan Pemrogram dalam Bahasa Pascal dan C, Bandung: Informatika Bandung, 2005, Hal 20.

## Percobaan 1

### Penggunaan Konstanta

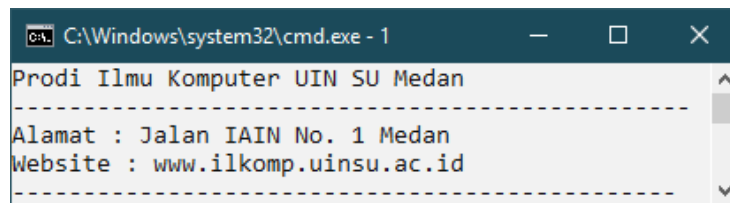
Program C:



```
1 #include "stdio.h"
2 #include "conio.h"
3 void main()
4 {
5     clrscr();
6     printf("Prodi Ilmu Komputer UIN SU Medan\n");
7     printf("-----\n");
8     printf("Alamat : Jalan IAIN No. 1 Medan\n");
9     printf("Website : www.ilkomp.uinsu.ac.id\n");
10    printf("-----\n");
11    getch();
12 }
```

Gambar 5. Program Penggunaan Konstanta

### Output:



```
C:\Windows\system32\cmd.exe - 1
Prodi Ilmu Komputer UIN SU Medan
-----
Alamat : Jalan IAIN No. 1 Medan
Website : www.ilkomp.uinsu.ac.id
-----
```

Gambar 6. Ouput Program Penggunaan Konstanta

### Pembahasan :

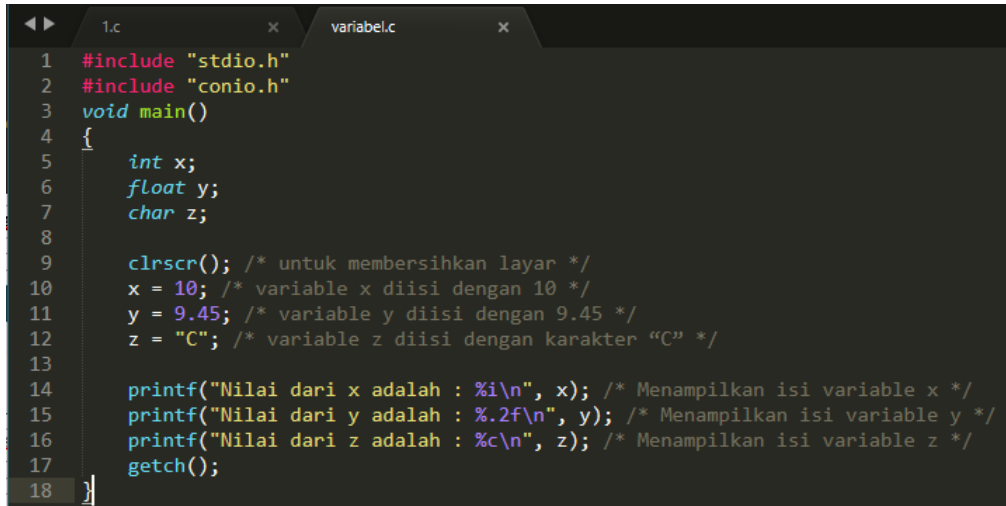
- Penulisan pada printf pertama menghasilkan keluaran “Prodi Ilmu Komputer UIN SU Medan” di tambah \n yang menyatakan ganti baris.
- Penulisan pada printf kedua memberikan keluaran “-----” di tambah \n yang menyatakan ganti baris.
- Penulisan pada printf ketiga memberikan keluaran “Alamat : Jalan IAIN No. 1 Medan” di tambah \n yang menyatakan ganti baris.
- Penulisan pada printf keempat memberikan keluaran “Website : ilkomp.uinsu.ac.id” di tambah \n yang menyatakan ganti baris.
- Penulisan pada printf kedua memberikan keluaran “-----” di tambah \n yang menyatakan ganti baris.



## Percobaan 2

### Penggunaan Tipe Data dan Variabel

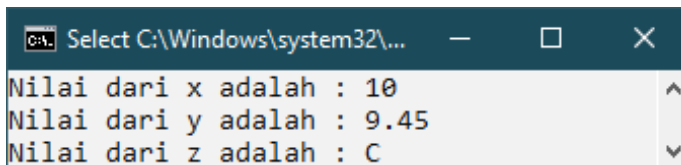
Program C:



```
1 #include "stdio.h"
2 #include "conio.h"
3 void main()
4 {
5     int x;
6     float y;
7     char z;
8
9     clrscr(); /* untuk membersihkan layar */
10    x = 10; /* variable x diisi dengan 10 */
11    y = 9.45; /* variable y diisi dengan 9.45 */
12    z = "C"; /* variable z diisi dengan karakter "C" */
13
14    printf("Nilai dari x adalah : %i\n", x); /* Menampilkan isi variable x */
15    printf("Nilai dari y adalah : %.2f\n", y); /* Menampilkan isi variable y */
16    printf("Nilai dari z adalah : %c\n", z); /* Menampilkan isi variable z */
17    getch();
18 }
```

Gambar 7. Program Tipe Data dan Variabel

Output:



```
Nilai dari x adalah : 10
Nilai dari y adalah : 9.45
Nilai dari z adalah : C
```

Gambar 8. Output Tipe Data dan Variabel

Pembahasan :

- Penulisan pada printf pertama menampilkan isi variable x, penggunaan %i menyatakan bilangan bulat dari angka 10.
- Penulisan pada printf kedua menampilkan isi variable y, penggunaan %.2f menyatakan bahwa dua angka dibelakang koma bilangan pecahan dari 9.45.
- Penulisan pada printf kedua menampilkan isi variable z, penggunaan %c menyatakan karakter/string dari C.

## MODUL 3

### ARITMATIKA DAN OPERATOR

#### 1. Operator Penugasan

Operator Penugasan (Assignment operator) dalam bahasa C berupa tanda sama dengan (“=”). Contoh : nilai = 80; A = x \* y; Artinya : variable “nilai” diisi dengan 80 dan variable “A” diisi dengan hasil perkalian antara x dan y (Sismoro, Heri. 2005 : 20<sup>6</sup>).

#### 2. Operator Aritmatika

Bahasa C menyediakan lima operator aritmatika, yaitu :

- \* : untuk perkalian
- / : untuk pembagian
- % : untuk sisa pembagian (modulus)
- + : untuk penambahan
- - : untuk pengurangan

Catatan : operator % digunakan untuk mencari sisa pembagian antara dua bilangan.

Misalnya :

$$9 \% 2 = 1 \quad 9 \% 3 = 0$$

$$9 \% 5 = 4 \quad 9 \% 6 = 3$$

#### 3. Operator Hubungan (Perbandingan)

Operator Hubungan digunakan untuk membandingkan hubungan antara dua buah operand (sebuah nilai atau variable. Operator hubungan dalam bahasa C :

Operator	Arti	Contoh	
<	Kurang dari	$x < y$	Apakah x kurang dari y
<=	Kurang dari sama dengan	$x <= y$	Apakah x kurang dari sama dengan y
>	Lebih dari	$x > y$	Apakah x lebih dari y
>=	Lebih dari sama dengan	$x >= y$	Apakah x lebih dari sama dengan y
==	Sama dengan	$x == y$	Apakah x sama dengan y

<sup>6</sup> Sismoro, Heri. 2005, Pengantar Logika Informatika, Algoritma dan Pemrograman Komputer, Yogyakarta : ANDI Publishing, Hal. 20.

!=	Tidak sama dengan	x != y	Apakah x tidak sama dengan y
----	-------------------	--------	------------------------------

Tabel 2. Operator Hubungan (Perbandingan)

## Percobaan 1

Operator Aritmatika

Program C:

```

operator.c
1  #include "stdio.h"
2  #include "conio.h"
3  void main()
4  {
5      clrscr();
6      printf("Nilai dari 9 + 4 = %i\n", 9 + 4);
7      printf("Nilai dari 9 - 4 = %i\n", 9 - 4);
8      printf("Nilai dari 9 * 4 = %i\n", 9 * 4);
9      printf("Nilai dari 9 / 4 = %i\n", 9 / 4);
10     printf("Nilai dari 9 % 4 = %i\n", 9 % 4);
11     getch();
12 }

```

Gambar 9. Program Operator Aritmatika

Output:

```

C:\Windows\system32\cmd.exe - operator
Nilai dari 9 + 4 = 13
Nilai dari 9 - 4 = 5
Nilai dari 9 * 4 = 36
Nilai dari 9 / 4 = 2
Nilai dari 9 % 4 = 1

```

Gambar 10. Ouput Program Operator Aritmatika

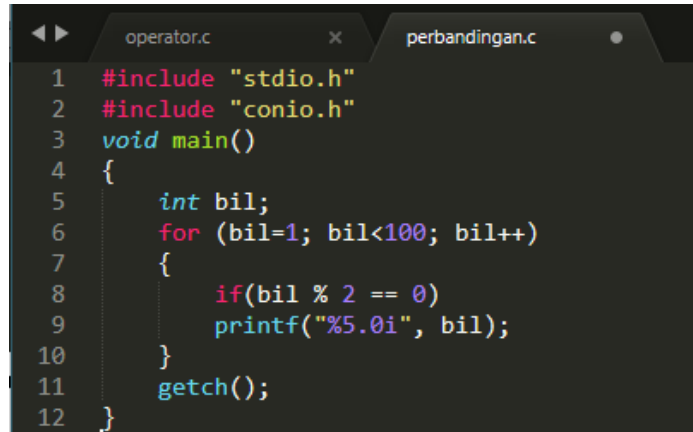
Pembahasan :

- Fungsi dari + ialah menghitung pertambahan.
- Fungsi dari – ialah menghitung pengurangan.
- Fungsi dari \* ialah menghitung perkalian.
- Fungsi dari / ialah menghitung pembagian.
- Fungsi dari % ialah menghitung sisa pembagian.
- Dalam prioritas operator hitung dikerjakan dari kiri ke kanan. Bilangan dalam kurung didahulukan. Perkalian atau pembagian yang muncul dari kiri dikerjakan terlebih dahulu. Pada dasarnya pemprioritasan seperti yang diterapkan dalam ilmu matematika.

## Percobaan 2

### Operator Hubungan (Perbandingan)

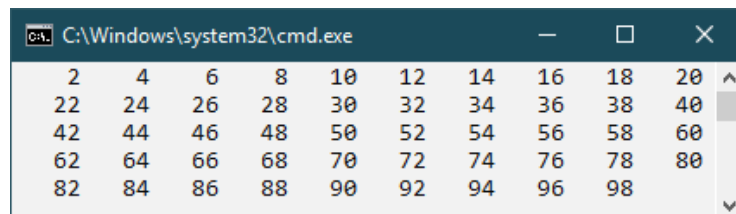
Program C:



```
1 #include "stdio.h"
2 #include "conio.h"
3 void main()
4 {
5     int bil;
6     for (bil=1; bil<100; bil++)
7     {
8         if(bil % 2 == 0)
9             printf("%5.0i", bil);
10    }
11    getch();
12 }
```

Gambar 11. Program Operator Perbandingan

### Output:



```
C:\Windows\system32\cmd.exe
 2  4  6  8 10 12 14 16 18 20
22 24 26 28 30 32 34 36 38 40
42 44 46 48 50 52 54 56 58 60
62 64 66 68 70 72 74 76 78 80
82 84 86 88 90 92 94 96 98
```

Gambar 12. Output Program Perbandingan

### Pembahasan :

- Nilai nol berarti bahwa perbandingan memberikan hasil bernilai salah.
- Nilai 1 berarti bahwa perbandingan memberikan hasil bernilai benar.

## MODUL 4

### SCANF

#### Percobaan 1

Perintah `scanf`, atau lebih tepatnya function `scanf()` adalah perintah bahasa C untuk menerima masukan ke dalam program, yakni sebagai sarana input dari pengguna. Dengan menggunakan perintah `scanf`, kita bisa membuat program yang lebih interaktif, yakni meminta data dari user / pengguna (Sismoro, Heri. 2005 : 25<sup>7</sup>). Data ini nantinya bisa disimpan ke dalam variabel dan diolah lebih lanjut untuk kemudian ditampilkan kembali . Dengan format :

**`scanf(kode_format, &nama_variabel_penampung)`**

Bagian `kode_format` adalah format untuk tipe data inputan. Kode format ini sama seperti yang dipakai untuk fungsi `printf`, misalnya kode “%d” untuk tipe data integer, atau “%c” untuk tipe data char.

%c	Membaca sebuah karakter
%s	Membaca sebuah string
%i, %d	Membaca sebuah bilangan bulat (integer)
%f, %e	Membaca sebuah bilangan pecahan (real)
%o	Membaca sebuah bilangan octal
%x	Membaca sebuah bilangan heksadesimal
%u	Membaca sebuah bilangan tak bertanda

Tabel 3. Kode Penentu Format

---

<sup>7</sup> Sismoro, Heri. 2005, *Pengantar Logika Informatika, Algoritma dan Pemrograman Komputer*, Yogyakarta : ANDI Publishing, Hal. 25.

## Bahasa C

```
operator.c x perbandingan.c x scanf.c 1.C x
1  /* Program memasukan inputan dengan beberapa tipe data */
2  #include "stdio.h"
3  #include "conio.h"
4  void main()
5  {
6      int jumlah;
7      char huruf, nim[10];
8      float nilai;
9
10     clrscr();
11     printf("Masukkan sebuah Bilangan Bulat : "); scanf("%d", &jumlah ); /* membaca sebuah bilangan bulat */
12     printf("Masukkan sebuah Karakter : "); scanf("%c", &huruf ); /* membaca sebuah karakter */
13     printf("Masukkan NIM : "); scanf("%s", &nim ); /* membaca sebuah string */
14     printf("Masukkan Pecahan : "); scanf("%f", &nilai ); /* membaca sebuah bilangan float */
15     printf("\nNilai variabel yang Anda masukkan adalah :\n");
16     printf("Jumlah : %d\n", jumlah );
17     printf("Huruf : %c\n", huruf );
18     printf("NIM : %s\n", nim );
19     printf("Nilai : %.1f\n", nilai );
20     getch();
21 }
```

Gambar 7. Program Penggunaan Scanf

## Output

```
C:\Windows\system32\cmd.exe
Masukkan sebuah Bilangan Bulat : 8
Masukkan sebuah Karakter : D
Masukkan NIM : 0701182140
Masukkan Pecahan : 85.5

Nilai variabel yang anda masukkan adalah :

Jumlah : 8
Huruf : D
NIM : 0701182140
Nilai : 85.5
```

Gambar 14. Output Program Penggunaan Scanf

## MODUL 5

### GETS()

#### Pengertian GETS()

Fungsi ini adalah fungsi yang sempurna dalam penginputan string. Fungsi ini seperti menambal kekurangan dari scanf() ketika menginput string. Tentu saja, fungsi ini hanya bisa menginput string (Suprpto, Kadarisman Tejo Yuwono, Totok Sukardiyono dan Adi Dewanto, 2008, Hal. 30<sup>8</sup>). Dengan fungsi ini kita bisa menambahkan spasi dalam string inputan juga.

```
1 gets(nama_variabel);
```

- Fungsi gets() digunakan untuk memasukkan data bertipe karakter dan tidak dapat digunakan untuk.
- Memasukkan data numerik.
- Harus diakhiri dengan penekanan tombol enter.
- Cursor secara otomatis akan pindah baris.
- Tidak memerlukan penentu format.

#### Percobaan 1

##### Penggunaan gets()

Program C:

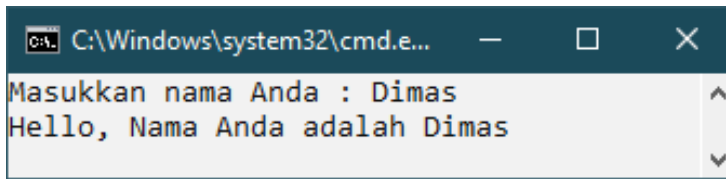


```
1 #include "stdio.h"
2 #include "conio.h"
3 void main()
4 {
5     char nama[20];
6
7     printf("Masukkan nama Anda : ");
8     gets(nama);
9
10    printf("Hello, Nama Anda adalah %s", nama);
11    getch();
12 }
```

Gambar 15. Program Penggunaan gets()

<sup>8</sup> Suprpto, Kadarisman Tejo Yuwono, Totok Sukardiyono dan Adi Dewanto, 2008, Bahasa Pemrograman untuk SMK, Jakarta : Depdiknas, Hal. 30.

## Output:



```
C:\Windows\system32\cmd.e...
Masukkan nama Anda : Dimas
Hello, Nama Anda adalah Dimas
```

Gambar 16. Ouput Program Penggunaan gets()

## Pembahasan :

- a. Tipe data karakter maupun string, pendeklarasiannya menggunakan **char**. Variabel yang digunakan adalah **nama** dengan batas 20 karakter.
- b. Penulisan pada printf pertama menghasilkan keluaran “Masukkan nama Anda : “. Pesan ini ditujukan kepada user (pengguna) yang meminta user untuk menginputkan data **nama**.
- c. Setelah data diinputkan oleh user. Data tersebut akan dibaca oleh **gets(nama)** dan dimasukkan kedalam variabel **nama**.
- d. Penulisan pada printf kedua menghasilkan keluaran “Hello, Nama Anda adalah Dimas”. Pada bagian inilah program akan menampilkan data **nama** yang telah dimasukkan oleh user. Dengan kode penentu format **%s**.



## MODUL 6

### GETCHAR(), GETCH(), GETCHE()

#### 1. GETCHAR()

Getchar merupakan Singkatan dari get character. Fungsinya sama dengan fungsi getch(), namun jika kita menggunakan fungsi ini user harus mengakhiri inputan dengan tombol enter.

- Fungsi getchar() digunakan untuk membaca data karakter
- Harus diakhiri dengan penekanan tombol enter
- Karakter yang dimasukkan terlihat pada layar
- Pergantian baris secara otomatis

#### 2. GETCH()

Fungsi ini memiliki fungsi yang sama dengan fungsi getchar(), yaitu sama - sama digunakan untuk memasukkan sebuah karakter dan tanpa diakhiri dengan enter. Namun fungsi getch() tidak akan menampilkan karakter yang telah diketik. jadi karakter tersebut tidak akan muncul namun masuk dalam program dan tersimpan secara otomatis (Suprpto, Kadarisman Tejo Yuwono, Totok Sukardiyono dan Adi Dewanto, 2008, Hal. 35<sup>9</sup>). Contohnya dalam permintaan sandi saat membuat akun ataupun log in sebuah situs.

- Fungsi getch() digunakan untuk membaca data karakter.
- Karakter yang dimasukkan tidak perlu diakhiri dengan penekanan tombol enter.
- Tidak memberikan efek pergantian baris secara otomatis.
- Jika menggunakan fungsi getch() karakter yang dimasukkan tidak akan ditampilkan pada layer sehingga sering digunakan untuk meminta inputan berupa password.

#### 3. GETCHE()

Fungsi ini meminta satu buah karakter dari keyboard tanpa diakhiri dengan tombol enter dan karakter yang telah diketik oleh user akan ditampilkan di layar.

- Fungsi getch() digunakan untuk membaca data karakter.

---

<sup>9</sup> Suprpto, Kadarisman Tejo Yuwono, Totok Sukardiyono dan Adi Dewanto, 2008, Bahasa Pemrograman untuk SMK, Jakarta : Depdiknas, Hal. 35.

- Pada `getche()` karakter yang dimasukkan akan ditampilkan pada layar.

## Percobaan 1

Penggunaan `gets()`

Program C:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     char c[20];
6
7     printf("masukkan nama anda: \n");
8     gets(c);
9     printf("nama yang anda masukkan adalah %s\n",c);
10
11     getch();
12     return 0;
13 }
```

Gambar 17. Program Penggunaan `gets()`

**Output:**



```
Command Prompt
masukkan nama anda:
yevin
nama yang anda masukkan adalah yevin
D:\Bahasa Program C basic>
```

Gambar 18. Output Program Penggunaan `gets()`

**Pembahasan :**

Sintak: `int getch(void):`

Fungsi

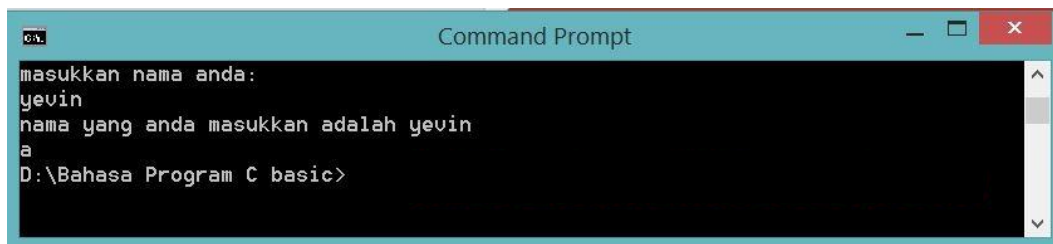
- mengembalikan satu karakter dari buffer keyboard
- karakter tidak ditampilkan di layar monitor (no echo)
- Tidak menunggu sampai ada ENTER
- Cocok untuk membuat password
- Header file ada di `conio.h`

**Percobaan 2**

Getche :

```
1 #include <stdio.h>
2
3 int main()
4 {
5     char c[20];
6
7     printf("masukkan nama anda: \n");
8     gets(c);
9     printf("nama yang anda masukkan adalah %s\n",c);
10
11     getch();
12     return 0;
13 }
```

Gambar 19. Program Penggunaan getch()



```
masukkan nama anda:
yevin
nama yang anda masukkan adalah yevin
a
D:\Bahasa Program C basic>
```

Gambar 20. Output Program Penggunaan getch()

### Pembahasan :

Sintak : int getch(void)

Fungsi :

- mengembalikan satu karakter dari keyboard
- Karakter ditampilkan di layar (echo)
- Tidak menunggu sampai ada ENTER
- Header file ada di conio.h

### Percobaan 3

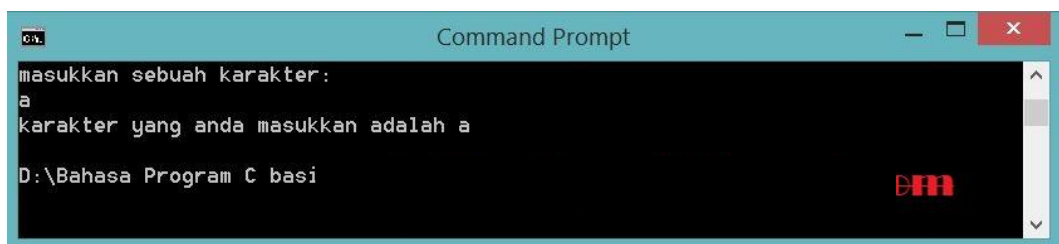
Getchar :

```

1  #include <stdio.h>
2
3  int main()
4  {
5      char c; // jika menggunakan getchar() tidak perlu memasukkan jumlah byte.
6              // karena getchar adalah void.
7      printf("masukkan sebuah karakter: \n");
8      c=getchar();
9      printf("karakter yang anda masukkan adalah %c\n",c);
10
11     getch();
12     return 0;
13 }

```

Gambar 21. Program Penggunaan getchar()



```

Command Prompt
masukkan sebuah karakter:
a
karakter yang anda masukkan adalah a
D:\Bahasa Program C basi

```

Gambar 22. Output Program Penggunaan getchar()

### Pembahasan :

Sintak: int getchar(void):

Fungsi:

- mengembalikan sebuah karakter (nilai ASCII) berikutnya dari buffer keyboard.
- Karakter ditampilkan di layar monitor
- Menunggu sampai ada ENTER
- Header file ada di stdio.h

## MODUL 7

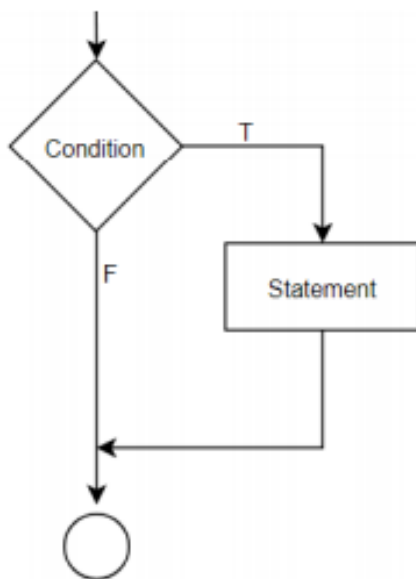
### IF

#### 1. IF

Merupakan struktur pilihan yang terdiri dari ekspresi boolean yang diikuti oleh satu atau lebih statement. Ekspresi boolean selalu bernilai benar atau salah sebagai syarat terpilihnya suatu keputusan yang akan diambil. Statement yang dipilih harus memenuhi persyaratan yang ditentukan berdasarkan kebutuhan. Struktur if dibentuk dari pernyataan if dan sering digunakan untuk menyeleksi suatu kondisi tunggal (Suprpto, Kadarisman Tejo Yuwono, Totok Sukardiyono dan Adi Dewanto, 2008 : 40<sup>10</sup>). Bila proses yang diseleksi terpenuhi atau bernilai benar, maka pernyataan yang ada di dalam blok if akan diproses dan dikerjakan. Bentuk umum struktur kondisi if adalah :

```
1  if(boolean_expression) {  
2      statement;  
3      ...  
4  }
```

Dalam pembuatan program, ada saatnya kita butuh suatu percabangan, yakni jika sebuah kondisi terpenuhi, jalankan kode program ini, jika tidak, jalankan kode program yang lain.



Dimana jika nilai suatu variabel memenuhi syarat pada condition maka statement akan dikerjakan dan jika tidak memenuhi syarat, maka statement akan diabaikan. Dengan kata lain, struktur If Statement merupakan struktur dengan opsi tunggal. Sehingga jika kondisi variabel tidak terpenuhi pada nilai benar, maka statement diabaikan (Drs.Suarga, M.sc.,M.Math.,Ph.D. 2006 : 18<sup>11</sup>)

<sup>10</sup> Suprpto, Kadarisman Tejo Yuwono, Totok Sukardiyono dan Adi Dewanto, 2008, *Bahasa Pemrograman untuk SMK*, Jakarta : Depdiknas, Hal. 40.

<sup>11</sup> Drs.Suarga, M.sc.,M.Math.,Ph.D. 2006. *Algoritma Pemrograman*, Yogyakarta :CV ANDI OFFSET, Hal. 18)

## Percobaan 1

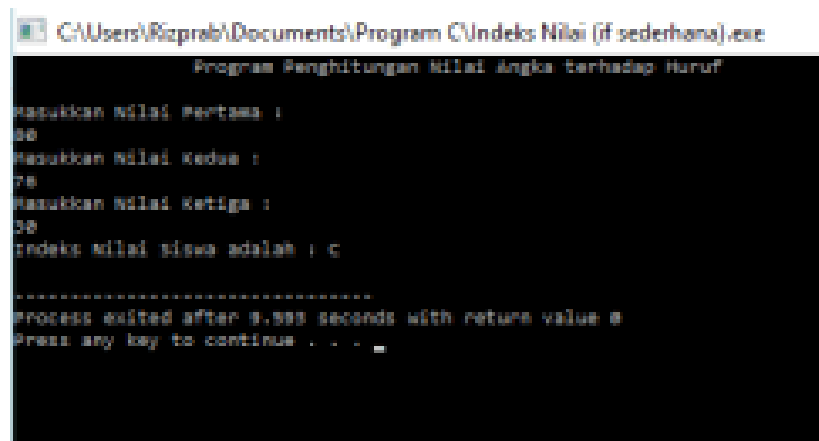
Penggunaan gets()

Program C:

```
1 #include <stdio.h>
2
3 main(){
4     int a, b, c, x;
5     char indeks;
6
7     printf("\t\tProgram Penghitungan Nilai Angka terhadap Huruf\n");
8     printf("\nMasukkan Nilai Pertama : \n");
9     scanf("%d", &a);
10
11    printf("Masukkan Nilai Kedua : \n");
12    scanf("%d", &b);
13
14    printf("Masukkan Nilai Ketiga : \n");
15    scanf("%d", &c);
16
17    x = (a+b+c)/3;
18
19    if(x<=40)indeks = 'E'; //bila nilai kurang dari 40 maka indeks nilai E
20    else if(x>40 && x<=55)indeks = 'D'; //bila nilai lebih dari 40 dan kurang dari 55 maka indeks nilai D
21    else if(x>55 && x<=70)indeks = 'C';
22    else if(x>70 && x<=85)indeks = 'B';
23    else indeks = 'A'; //bila nilai diatas semua angka diatas maka indeks nilai A
24
25    printf("Indeks Nilai Siswa adalah : %c\n", indeks);
26
27    return 0;
28 }
```

Gambar 23. Program Penggunaan if

Output:



```
C:\Users\Rizprab\Documents\Program C\Indeks Nilai (if sederhana).exe
Program Penghitungan nilai angka terhadap Huruf
Masukkan nilai pertama :
10
Masukkan nilai kedua :
70
Masukkan nilai ketiga :
30
Indeks nilai siswa adalah : C
.....
Process exited after 0.000 seconds with return value 0
Press any key to continue . . .
```

Gambar 24. Ouput Program Penggunaan if

## MODUL 8

### IF ELSE

#### 1. IF ELSE

Dalam struktur kondisi if... else minimal terdapat dua pernyataan. Jika kondisi yang bernilai benar atau terpenuhi maka pernyataan yang dilakasaan dan jika kondisi yang diperiksa bernilai salah maka pernyataan yang kedua yang dihasilkan. Bentuk umumnya adalah sebagai berikut : if(kondisi). IF dan ELSE digunakan saat kita akan mengecek lebih dari 1 kali, tentu saja kita tetap memberikan pernyataan di setiap if,else if, dan else (Drs.Suarga, M.sc.,M.Math.,Ph.D. 2006 : 20<sup>12</sup>). Bentuk umumnya adalah sebagai berikut :

```
1  if(kondisi)
    {
2  Pertanyaan-1;
3  {
4  Else
5  {
6  Pernyataan-2;
7  }
```

---

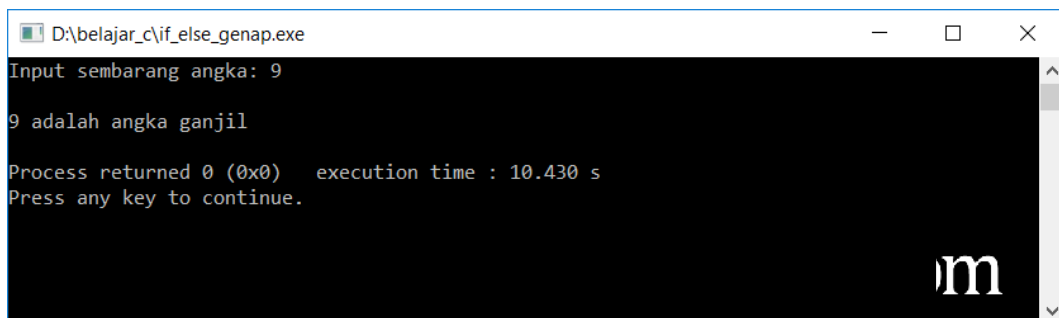
<sup>12</sup> Drs.Suarga, M.sc.,M.Math.,Ph.D. 2006. *Algoritma Pemrograman*, Yogyakarta :CV ANDI OFFSET, Hal. 20

## Percobaan 1

```
wewe.html
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int a;
6
7      printf("Input sembarang angka: ");
8      scanf("%d",&a);
9
10     printf("\n");
11
12     if (a % 2 == 0) {
13         printf("%d adalah angka genap \n", a);
14     }
15     else {
16         printf("%d adalah angka ganjil \n", a);
17     }
18
19     return 0;
20 }
```

Gambar 25. Program Penggunaan If Else

## Output:



```
D:\belajar_c\if_else_genap.exe
Input sembarang angka: 9
9 adalah angka ganjil
Process returned 0 (0x0) execution time : 10.430 s
Press any key to continue.
```

Gambar 26. Output Program Penggunaan If Else

## Pembahasan :

Sekarang jika kondisi `if (a % 2 == 0)` menghasilkan false, bagian ELSE lah yang akan di proses. Kode program akan jadi lebih efisien karena pemeriksaan kondisi hanya perlu dilakukan 1 kali saja.



## MODUL 9

### ELSE IF

#### Percobaan 1

Logika Else If disebut juga IF bertingkat, digunakan untuk menangani beberapa kondisi yang berbeda dalam satu rutin pengecekan. Pada logika else If, statemen pada masing-masing IF akan dijalankan hanya jika kondisi didalam logika IF masing-masing bernilai benar. Pada akhir Else IF biasa ditambahkan Else saja untuk menangani jika semua kondisi IF dan Else IF diatasnya bernilai salah.

#### Bahasa C

```
1 #include "stdio.h"
2 #include "conio.h"
3 int main()
4 {
5     float nilai;
6     printf ("\n Masukkan nilai Anda : "); scanf("%f",&nilai);
7     if (nilai>=85)
8     {
9         printf ("Anda Lulus Memuaskan :D\n");
10    }
11    else if (nilai>=65)
12    {
13        printf ("Anda Lulus :)\n");
14    }
15    else
16    {
17        printf ("Anda Tidak Lulus :('");
18    }
19    getch ();
20 }
```

Gambar 27. Program Penggunaan Else If

#### Output

```
D:\else if.exe
Masukkan nilai Anda : 90
Anda Lulus Memuaskan :D
```

Gambar 28. Output Program Penggunaan Else If

#### Pembahasan :

- Variabel *nilai* memakai tipe data *float* karena nilai bisa saja berupa angka pecahan.
- If* pertama menyatakan keadaan jika nilai yang di-input lebih besar dari 85. Begitupun dengan *if* selanjutnya.

## MODUL 10

### SWITCH...CASE...DEFAULT

#### Percobaan 1

Struktur kondisi switch....case....default digunakan untuk penyeleksian kondisi dengan kemungkinan yang terjadi cukup banyak. Struktur ini akan melaksanakan salah satu dari beberapa pernyataan 'case' tergantung nilai kondisi yang ada di dalam switch. Selanjutnya proses diteruskan hingga ditemukan pernyataan 'break'. Jika tidak ada nilai pada case yang sesuai dengan nilai kondisi, maka proses akan diteruskan kepada pernyataan yang ada di bawah 'default' (Drs.Suarga, M.sc.,M.Math.,Ph.D. 2006 : 30<sup>13</sup>)

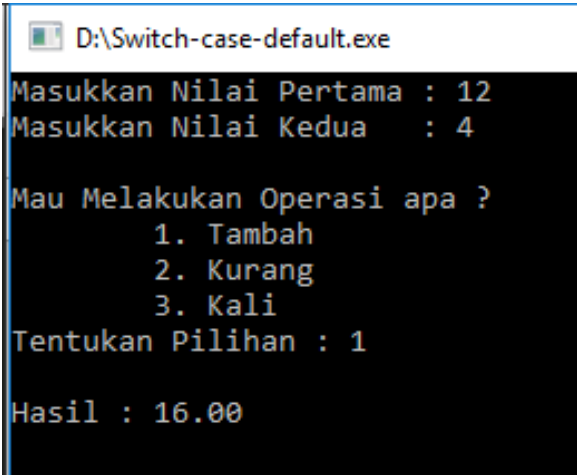
#### Bahasa C

```
1 #include<stdio.h>
2 #include<conio.h>
3 int main()
4 {
5     int pilihan,tambah, kurang, kali;
6     float a, b, hasil;
7     printf("Masukkan Nilai Pertama : ");scanf("%f", &a);
8     printf("Masukkan Nilai Kedua : ");scanf("%f", &b);
9     printf("\nMau Melakukan Operasi apa?");
10    printf("\n\t1. Tambah\n\t2. Kurang\n\t3. Kali");
11    printf("\nTentukan Pilihan : ");scanf("%d", &pilihan);
12    switch(pilihan)
13    {
14    {
15        case 1 : hasil = a + b;
16        break;
17        case 2 : hasil = a - b;
18        break;
19        case 3 : hasil = a * b;
20        break;
21        default : printf("\nTidak ada dalam Pilihan");
22        break;
23    }
24
25    printf("\nHasil : %.2f\n", hasil);
26    getch();
27 }
```

Gambar 29. Program Penggunaan Switch..Case..Default

<sup>13</sup> Drs.Suarga, M.sc.,M.Math.,Ph.D. 2006. *Algoritma Pemrograman*, Yogyakarta :CV ANDI OFFSET, Hal. 30

## Output



```
D:\Switch-case-default.exe
Masukkan Nilai Pertama : 12
Masukkan Nilai Kedua   : 4

Mau Melakukan Operasi apa ?
    1. Tambah
    2. Kurang
    3. Kali
Tentukan Pilihan : 1

Hasil : 16.00
```

Gambar 30. Output Program Penggunaan Switch..Case..Defaul

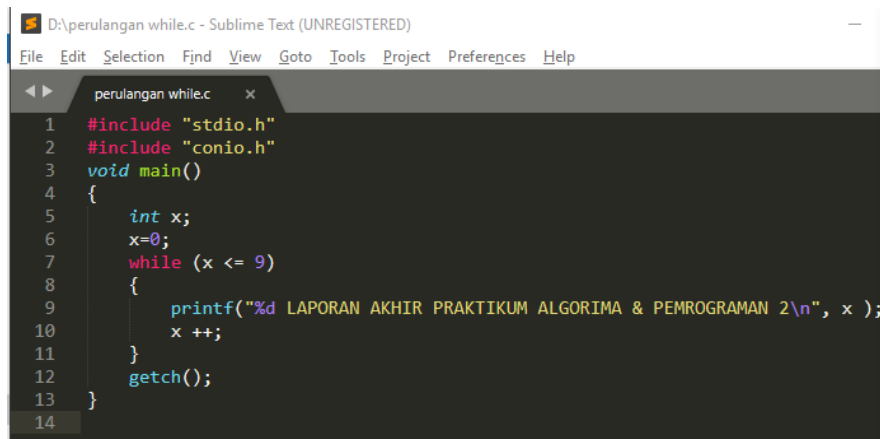
# MODUL 11

## WHILE

### Percobaan 1

While merupakan struktur algoritma dimana perulangan akan terus dijalankan selama kondisi bernilai benar. Perulangan yang dimaksud disini bisa saja berupa perulangan terhadap proses menghitung, mencetak (menampilkan) atau justru melakukan iterasi lain didalam perulangan tersebut. Kondisi nilai benar dan salah adalah kondisi yang menjadi syarat bagi variabel yang menjadi kontrol keputusan.

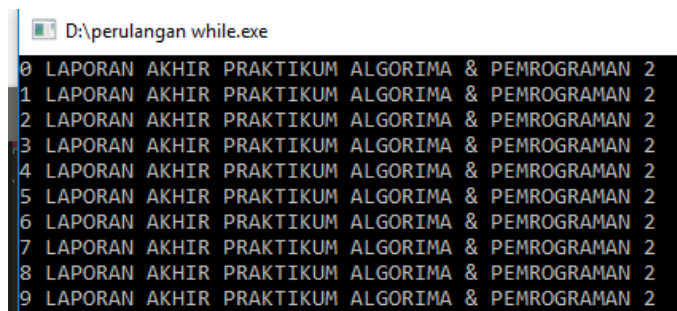
### Bahasa C



```
D:\perulangan while.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
perulangan while.c x
1 #include "stdio.h"
2 #include "conio.h"
3 void main()
4 {
5     int x;
6     x=0;
7     while (x <= 9)
8     {
9         printf("%d LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2\n", x );
10        x ++;
11    }
12    getch();
13 }
14
```

Gambar 31. Program Penggunaan While

### Output



```
D:\perulangan while.exe
0 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
1 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
2 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
3 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
4 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
5 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
6 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
7 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
8 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
9 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
```

Gambar 32. Output Program Penggunaan While

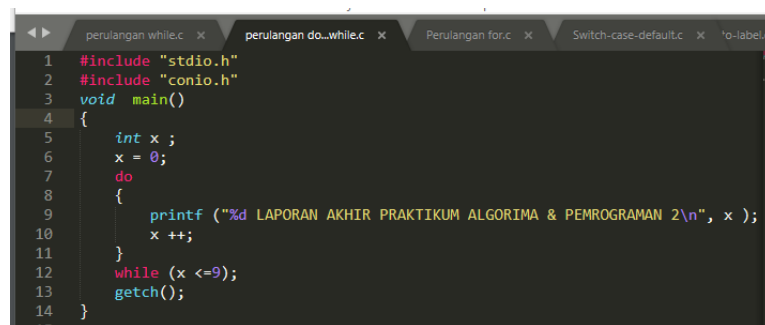
## MODUL 12

### DO...WHILE

#### Percobaan 1

Perulangan ini merupakan perulangan hampir sama dengan perulangan while. Dimana perulangan hanya akan dilakukan selama kondisi bernilai benar. Hanya saja pada perulangan do..while ini ekspresi dilakukan minimal satu kali. Jika kondisi benar maka ekspresi selanjutnya akan dilakukan. Dengan kata lain perulangan do..while mengerjakan ekspresi satu kali terlebih dahulu, kemudian melakukan pengujian untuk membuat keputusan apakah perulangan dilakukan atau tidak (R. Munir, *Algoritma dan Pemrograman dalam Bahasa Pascal dan C*, 2005 : 42<sup>14</sup>).

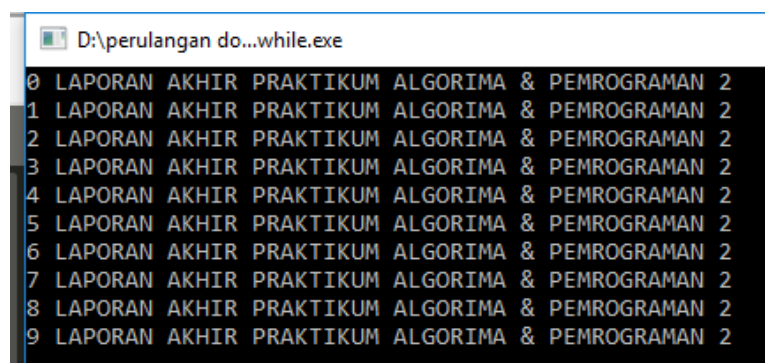
#### Bahasa C



```
1 #include "stdio.h"
2 #include "conio.h"
3 void main()
4 {
5     int x ;
6     x = 0;
7     do
8     {
9         printf ("%d LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2\n", x );
10        x ++;
11    }
12    while (x <=9);
13    getch();
14 }
```

Gambar 33. Program Penggunaan Do..While

#### Output



```
D:\perulangan do...while.exe
0 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
1 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
2 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
3 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
4 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
5 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
6 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
7 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
8 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
9 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
```

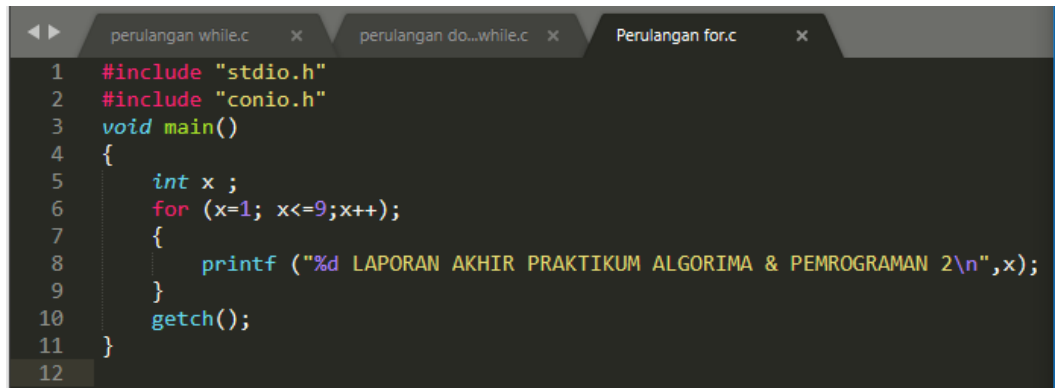
Gambar 34. Output Program Penggunaan While

<sup>14</sup> R. Munir, *Algoritma dan Pemrograman dalam Bahasa Pascal dan C*, Bandung: Informatika Bandung, 2005, Hal 45.

## Percobaan 1

Perulangan model ini merupakan perulangan dengan jumlah iterasi yang sudah tetap berapa kali iterasi akan dilakukan.

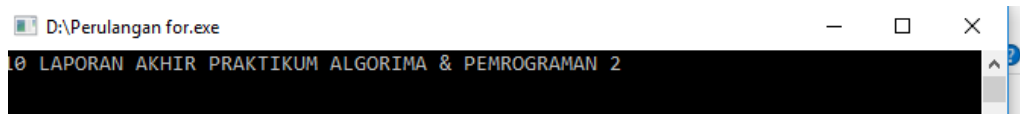
Bahasa C



```
1 #include "stdio.h"
2 #include "conio.h"
3 void main()
4 {
5     int x ;
6     for (x=1; x<=9;x++);
7     {
8         printf ("%d LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2\n",x);
9     }
10    getch();
11 }
12
```

Gambar 35. Output Program Penggunaan For

## Output



```
D:\Perulangan for.exe
0 LAPORAN AKHIR PRAKTIKUM ALGORIMA & PEMROGRAMAN 2
```

Gambar 36. Output Program Penggunaan For

## MODUL 14

### GO TO & LABEL

#### Percobaan 1

Goto" adalah sebuah statement yang mengarahkan pada sebuah statement label. Nama Label diakhiri dengan " :“ .

#### Bahasa C

```
16 {
17     case 1 : hasil = a + b;
18     break;
19     case 2 : hasil = a - b;
20     break;
21     case 3 : hasil = a * b;
22     break;
23     default : printf("\nTidak ada dalam Pilihan");
24     break;
25 }
26
27 printf("\nHasil : %.2f\n", hasil);
28 printf("\nApakah Anda ingin mengulang program? (1. Ya,tombol lain jika tidak !);scanf("%d",&ulang);
29 if (ulang==1) goto mulai;
30 else printf("\nTerimakasih :)");
31 getch();
32 }
```

Gambar 37. Program Penggunaan GoTo & Label

#### Output

```
D:\goto-label.exe
Masukkan Nilai Pertama : 12
Masukkan Nilai Kedua : 2

Mau Melakukan Operasi apa ?
1. Tambah
2. Kurang
3. Kali
Tentukan Pilihan : 1

Hasil : 14.00

Apakah Anda ingin mengulang program? (1. Ya,tombol
lain jika tidak !)1

Masukkan Nilai Pertama : 12
Masukkan Nilai Kedua : 2

Mau Melakukan Operasi apa ?
1. Tambah
2. Kurang
3. Kali
Tentukan Pilihan : 1

Hasil : 14.00

Apakah Anda ingin mengulang program? (1. Ya,tombol
lain jika tidak !)2

Terimakasih :)
```

Gambar 38. Output Program Penggunaan GoTo & Label

## DAFTAR PUSTAKA

- Utami, Ema. dan Sukrisno. 2005. 10 Langkah Belajar Logika dan Algoritma Menggunakan Bahasa C dan C++ di GNU/Linux. Andi Offset., Yogyakarta.
- R. Munir, Algoritma dan Pemrogram dalam Bahasa Pascal dan C, Bandung: Informatika Bandung, 2005.
- Sismoro, Heri. 2005, Pengantar Logika Informatika, *Algoritma dan Pemrograman Komputer*, Yogyakarta : ANDI Publishing.
- Suprpto, Kadarisman Tejo Yuwono, Totok Sukardiyono dan Adi Dewanto, 2008, Bahasa Pemrograman untuk SMK, Jakarta : Depdiknas.
- Drs.Suarga, M.sc.,M.Math.,Ph.D. 2006. Algoritma Pemrograman, Yogyakarta : CV ANDI OFFSET.