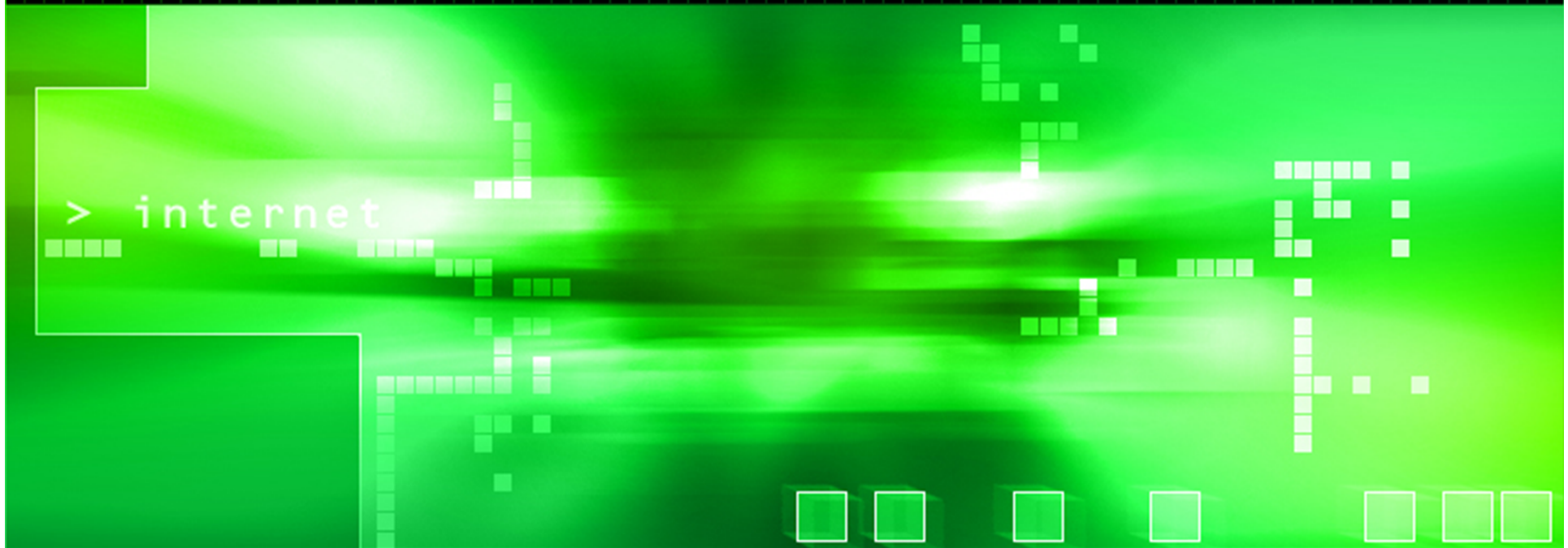


# Pemrograman I

## Bab IV – Input dan Output



Adam Mukharil Bachtiar, S.Kom.  
Universitas Komputer Indonesia

# Input Bahasa C

Format input bahasa C:

---

**scanf("format",&nama\_variabel);**

**dan untuk string**

**scanf("format",nama\_variabel);**

---



# Format Tipe Data Input Bahasa C

| Format            | Kegunaan  |
|-------------------|---|
| <b>%c</b>         | Digunakan untuk inputan bertipe data char               |
| <b>%i atau %d</b> | Digunakan untuk inputan bertipe data integer            |
| <b>%u</b>         | Digunakan untuk inputan bertipe data unsigned integer   |
| <b>%f</b>         | Digunakan untuk inputan bertipe data float              |
| <b>%o</b>         | Digunakan untuk inputan data angka berbasis oktal       |
| <b>%x</b>         | Digunakan untuk inputan data angka berbasis hexadesimal |
| <b>%s</b>         | Digunakan untuk inputan bertipe data string             |



# Contoh Input Bahasa C

```
1  /*
2    Program 4.1
3    Nama File   : Lat-4.1.c
4    Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(int argc, char *argv[])
11 {
12     int x,jam,menit,detik;
13     unsigned int y;
14     float z;
15     char nama[31];
16     scanf("%i",&x);
17     scanf("%u",&y);
18     scanf("%f",&z);
19     scanf("%i %i %i",&jam,&menit,&detik);
20     fflush(stdin);scanf("%s",nama);
21     system("PAUSE");
22     return 0;
23 }
```

# Input String Bahasa C

Format:

---

```
gets(nama_variabel);
```

---

N.B.:

gets mendukung pembacaan string yang mengandung spasi



# Input Karakter Bahasa C

Format:

---

**getch();**

**atau**

**getche();**

---

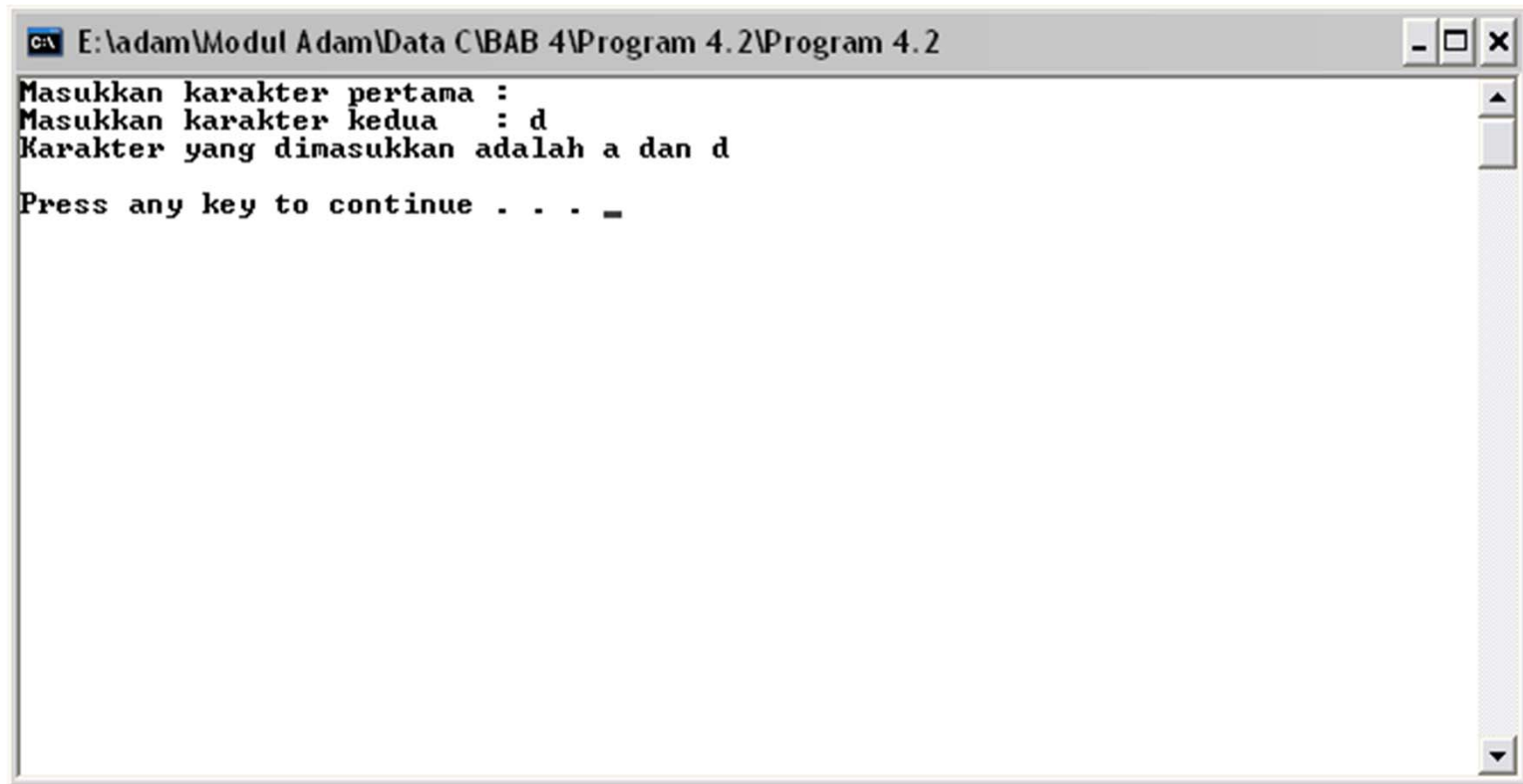
**Apa  
Bedanya???**



# Contoh Input Karakter Bahasa C

```
1  /*
2    Program 4.2
3    Nama File   : Lat-4.2.c
4    Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(int argc, char *argv[])
11 {
12     char k1,k2;
13     printf("Masukkan karakter pertama : ");
14     k1=getch();printf("\n");
15     printf("Masukkan karakter kedua   : ");
16     k2=getche();printf("\n");
17     printf("Karakter yang dimasukkan adalah %c dan
18           %c\n\n",k1,k2);
19     system("PAUSE");
20     return 0;
21 }
```

# Contoh Input Karakter Bahasa C



The screenshot shows a Windows-style application window titled "E:\adam\Modul Adam\Data C\BAB 4\Program 4.2\Program 4.2". The window contains a text-based interface for a C program. The text displayed is as follows:

```
Masukkan karakter pertama :  
Masukkan karakter kedua : d  
Karakter yang dimasukkan adalah a dan d  
Press any key to continue . . . _
```

On the left side of the slide, there is a green vertical bar with a pixelated pattern. It contains the text "> int" and several small, light green 3D cube icons arranged vertically.



# Input Bahasa C++

Format input bahasa C++:

---

```
cin>>nama_variabel;
```

---



# Input String Bahasa C++

Format input bahasa C:

---

```
cin.get(nama_variabel,jumlah_karakter_input);
```

---



# Contoh Input Bahasa C++

```
1  /*
2    Program 4.1
3    Nama File   : Lat-4.1.cpp
4    Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 int main(int argc, char *argv[])
13 {
14     int x,jam,menit,detik;
15     unsigned int y;
16     float z;
17     char nama[31];
18     cin>>x;
19     cin>>y;
20     cin>>z;
21     cin>>jam>>menit>>detik;
22     cin>>nama;
23     system("PAUSE");
24     return EXIT_SUCCESS;
25 }
```

# Output Bahasa C

Format output bahasa C:

---

```
printf("%format",nama_variabel);
```

---

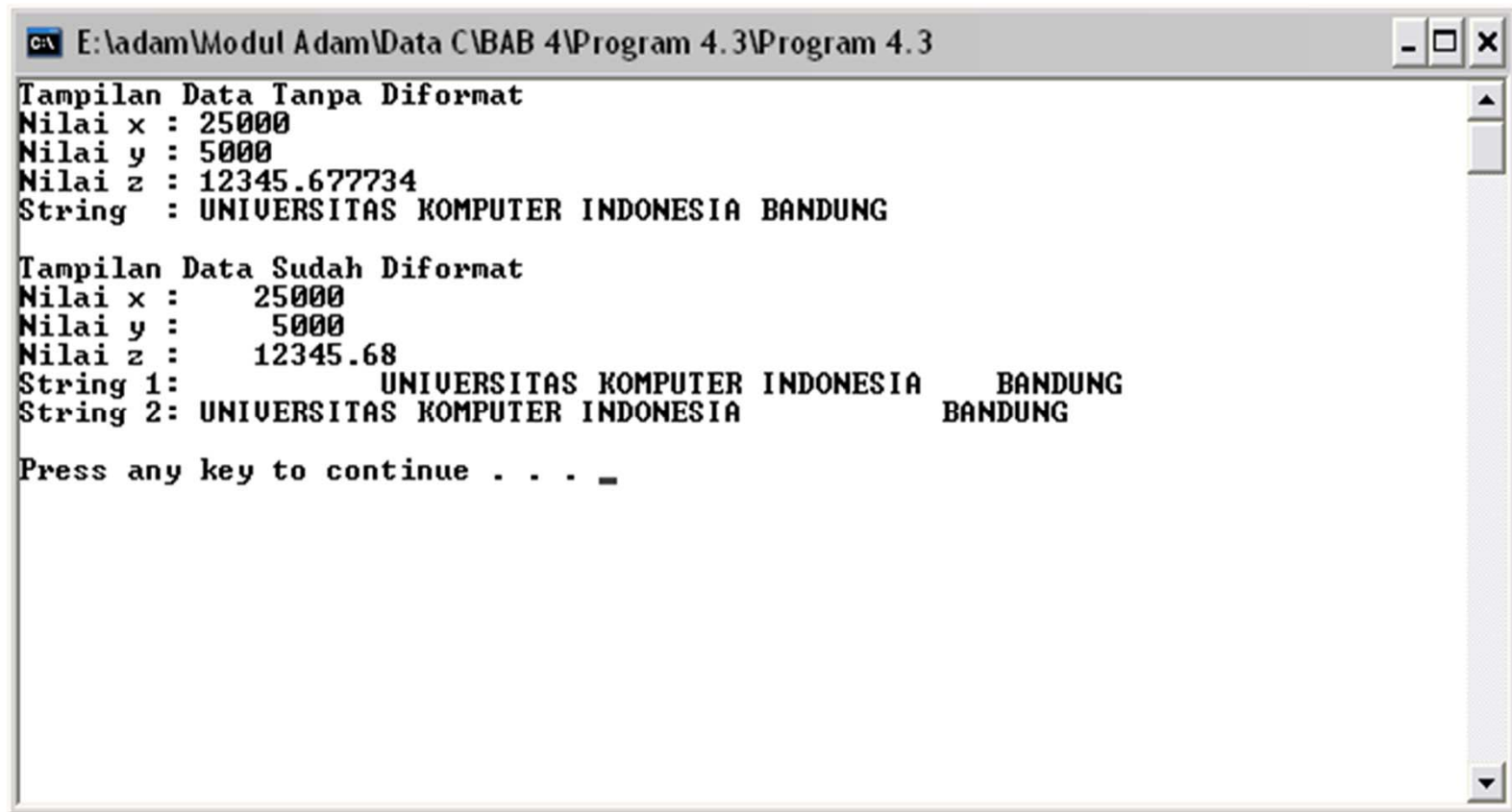




# Contoh Output Bahasa C

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(int argc, char *argv[])
5  {
6      int x=25000;
7      unsigned int y=5000;
8      float z=12345.678;
9      char univ[35]="UNIVERSITAS KOMPUTER INDONESIA";
10     char kota[10]="BANDUNG";
11     system("cls"); //untuk mengosongkan layar
12     printf("Tampilan Data Tanpa Diformat\n");
13     printf("Nilai x : %d\n",x);
14     printf("Nilai y : %u\n",y);
15     printf("Nilai z : %f\n",z);
16     printf("String   : %s %s\n\n",univ,kota);
17     printf("Tampilan Data Sudah Diformat\n");
18     printf("Nilai x : %8d\n",x);
19     printf("Nilai y : %8u\n",y);
20     printf("Nilai z : %11.2f\n",z);
21     printf("String 1: %40s %10s\n",univ,kota);//rata kanan
22     printf("String 2: %-40s %-10s\n\n",univ,kota);//rata kiri
23     system("PAUSE");
24     return 0;
25 }
```

# Contoh Output Bahasa C



The screenshot shows a Windows-style window titled "E:\adam\Modul Adam\Data C\BAB 4\Program 4.3\Program 4.3". The window contains the output of a C program. The output is divided into two sections: "Tampilan Data Tanpa Diformat" and "Tampilan Data Sudah Diformat". The first section shows raw values for variables x, y, z, and a string. The second section shows the same data formatted with tabs and spaces for better readability. At the bottom, it prompts the user to "Press any key to continue . . .".

```
C:\ E:\adam\Modul Adam\Data C\BAB 4\Program 4.3\Program 4.3
Tampilan Data Tanpa Diformat
Nilai x : 25000
Nilai y : 5000
Nilai z : 12345.677734
String  : UNIERSITAS KOMPUTER INDONESIA BANDUNG

Tampilan Data Sudah Diformat
Nilai x :      25000
Nilai y :       5000
Nilai z :    12345.68
String 1:          UNIVERSITAS KOMPUTER INDONESIA      BANDUNG
String 2: UNIVERSITAS KOMPUTER INDONESIA                BANDUNG

Press any key to continue . . . _
```

# Output Bahasa C++

Format output bahasa C++:

---

```
cout<<nama_variabel;
```

---

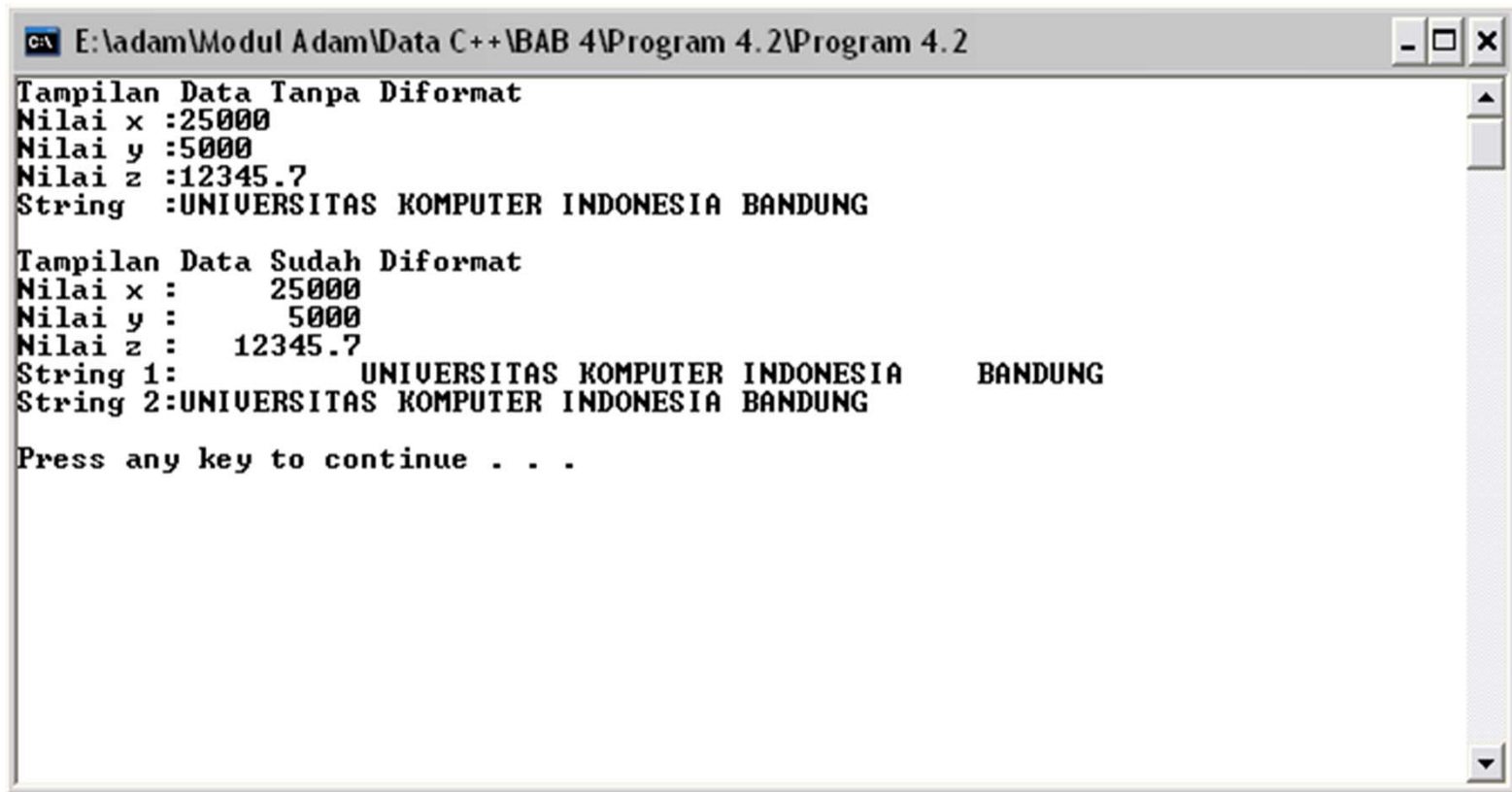


# Contoh Output Bahasa C++

```
1  #include <cstdlib>
2  #include <iostream>
3  #include <iomanip>
4
5  using namespace std;
6
7  int main(int argc, char *argv[])
8  {
9      int x=25000;
10     unsigned int y=5000;
11     float z=12345.678;
12     char univ[35]="UNIVERSITAS KOMPUTER INDONESIA";
13     char kota[10]="BANDUNG";
14     system("cls"); //untuk mengosongkan layar
15     cout<<"Tampilan Data Tanpa Diformat"<<endl;
16     cout<<"Nilai x : "<<x<<endl;
17     cout<<"Nilai y : "<<y<<endl;
18     cout<<"Nilai z : "<<z<<endl;
19     cout<<"String  : "<<univ<<" "<<kota<<"\n\n";
20     cout<<"Tampilan Data Sudah Diformat"<<endl;
21     cout<<"Nilai x : "<<setw(10)<<x<<endl;
22     cout<<"Nilai y : "<<setw(10)<<y<<endl;
23     cout<<"Nilai z : "<<setw(10)<<z<<endl;
24     cout<<"String 1: "<<setw(40)<<univ<<" "<<setw(10)<<kota<<endl;
25     cout<<"String 2: "<<setw(-40)<<univ<<" "<<setw(-10)<<kota<<"\n\n";
26     system("PAUSE");
27     return EXIT_SUCCESS;
28 }
```



# Contoh Output Bahasa C++



The screenshot shows a Windows-style window titled "E:\adam\Modul Adam\Data C++\BAB 4\Program 4.2\Program 4.2". The window contains the following text output from a C++ program:

```
Tampilan Data Tanpa Diformat
Nilai x :25000
Nilai y :5000
Nilai z :12345.7
String :UNIVERSITAS KOMPUTER INDONESIA BANDUNG

Tampilan Data Sudah Diformat
Nilai x :      25000
Nilai y :      5000
Nilai z :    12345.7
String 1:      UNIVERSITAS KOMPUTER INDONESIA      BANDUNG
String 2:UNIVERSITAS KOMPUTER INDONESIA BANDUNG

Press any key to continue . . .
```

The output demonstrates the difference between default and formatted printing of variables. The first section, "Tampilan Data Tanpa Diformat", shows raw values. The second section, "Tampilan Data Sudah Diformat", shows the same values with leading spaces for alignment. The string is also shown with a tab stop for better readability.

# Conio2.h

1. Install conio2.h
2. Tambahkan `#include <conio2.h>`
3. Tambahkan parameter `-lconio` pada bagian linker.
4. Gunakan fungsinya.



# Fungsi Pada Conio2.h

1. **clrscr();** → membersihkan layar.
2. **textcolor(angka);** atau **textcolor(COLOR);** → memberikan warna pada text.
3. **textbackground(angka);** atau **textbackground(COLOR);** → memberikan warna latar pada teks.
4. **gotoxy(horizontal,vertical);** → mengarahkan cursor ke arah tertentu di dos.

# COLOR Pada Conio2.h

- 
- 1. BLACK
  - 2. BLUE
  - 3. GREEN
  - 4. CYAN
  - 5. RED
  - 6. MAGENTA
  - 7. BROWN
  - 8. LIGHTGRAY
  - 9. DARKGRAY,
  - 10. LIGHTBLUE
  - 11. LIGHTGREEN
  - 12. LIGHTCYAN
  - 13. LIGHTRED
  - 14. LIGHTMAGENTA
  - 15. YELLOW
  - 16. WHITE



