COMPUTER ENGINEERING

Basic Programming Practicum

Introduction to CLanguage

2024

1 Goals

- Students can create projects within the IDE.
- Students can demonstrate their knowledge about program structure in the C language
- Students can demonstrate their knowledge about data types in the C language
- Students can demonstrate their knowledge of data types in the C language
- Students are able to use functions to read input from the keyboard
- Students are able to use functions to print text on the screen

2 Introduction to C Programming Language

The C language was developed by Dennis M. Ritchie and Brian W. Kernighan in the early 1970s. There are several standards for the C programming language. There are several guidelines for writing C programming language. Below are some of the standards:

1. Kernighan & Ritchie Definition(K&R)
2. ANSI-C (X-3.159 -1989-)
3. AT&T (for superset C, C++) definition, and
4. GNU Coding Standards

Implementation and uses of the C programming language

1. Creating operating systems and it's system programs
2. Programmin language that is "very close" to hardware (e.g., for device control).
3. Developing toolkits
4. Writing programs

3 IDE (Integrated Development Environment)

IDE stands for "Integrated Development Environment" in English. In Bahasa Indonesia, IDE can be translated as "Lingkungan Pengembangan Terintegrasi" or "Ruang Kerja Pengembangan Terpadu." IDE is a software designed to assist software developers in the process of development, coding, and testing computer applications.
Here are several list of C programming language IDE applications that can be used.

- Code::Blocks

4 Creating new project in IDE Code::Blocks

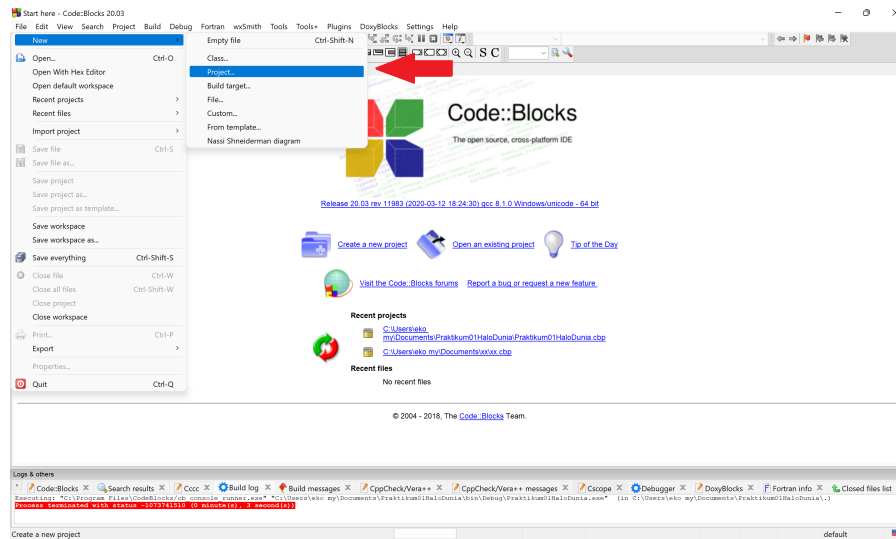## 4.1 Steps to create a new project

1. Go to File > New > Project
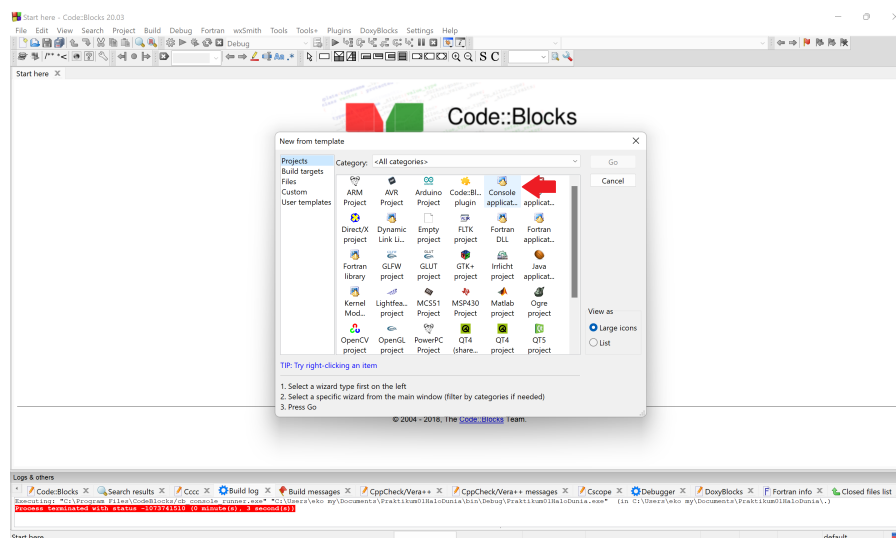

Figure 1

2. Click on Console Application

Figure 2

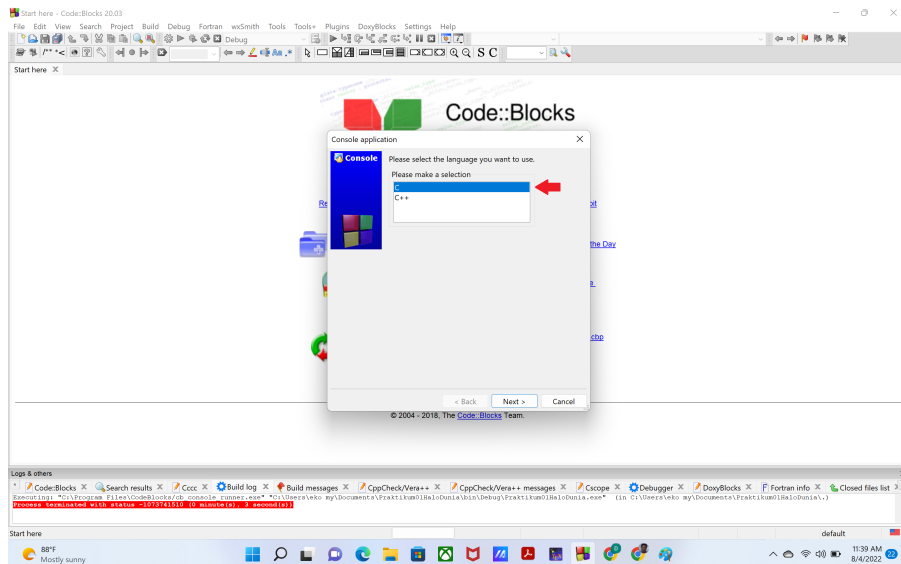3. Choose C as the programming language

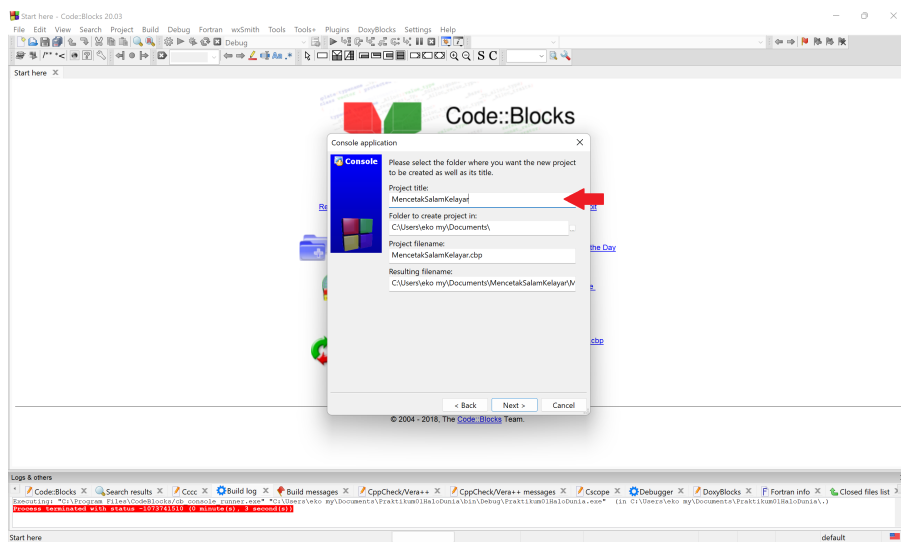**Figure 3**

4. Insert your project name

Figure 4

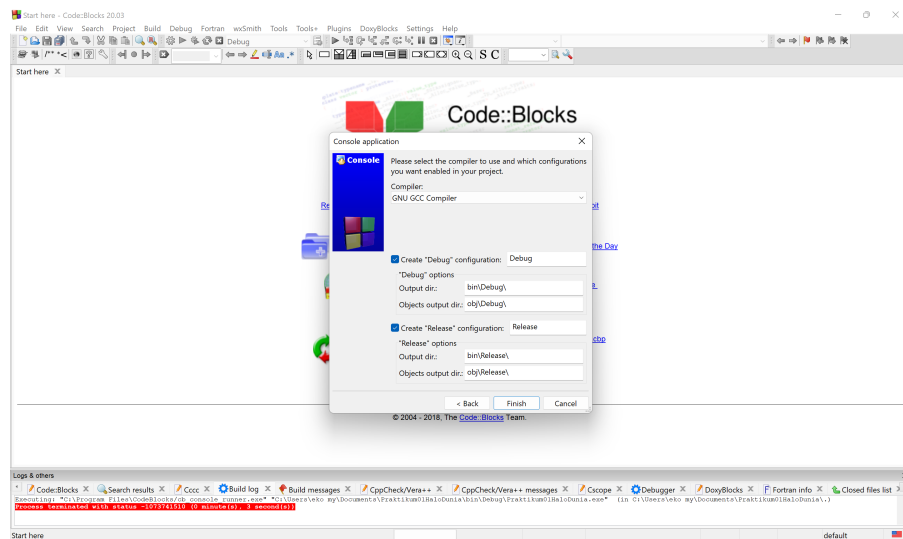5. Choose the compiler (gcc), select a directory to save your project, and click save

Figure 5
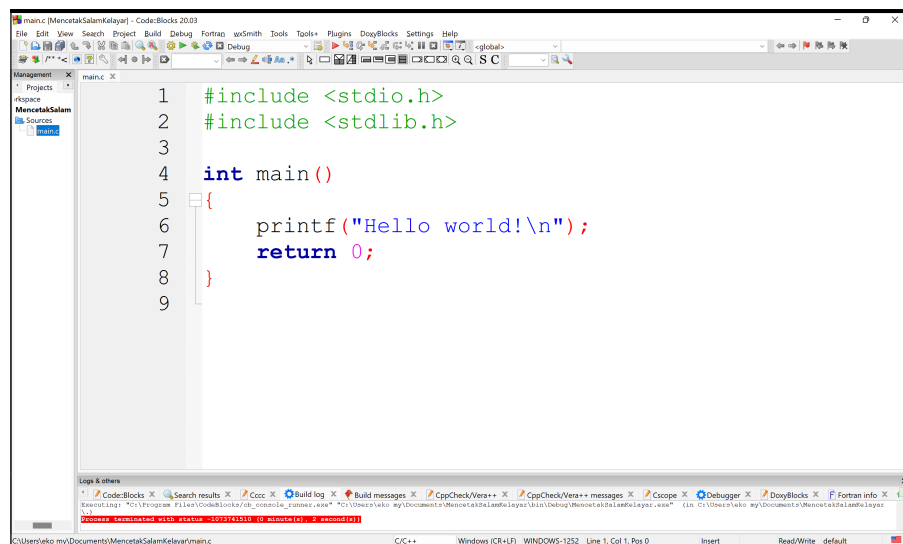
6. Write code as in Figure 6 in Code::Blocks

Figure 6

7. Click Build−>Build and Run or press F9 on your keyboard
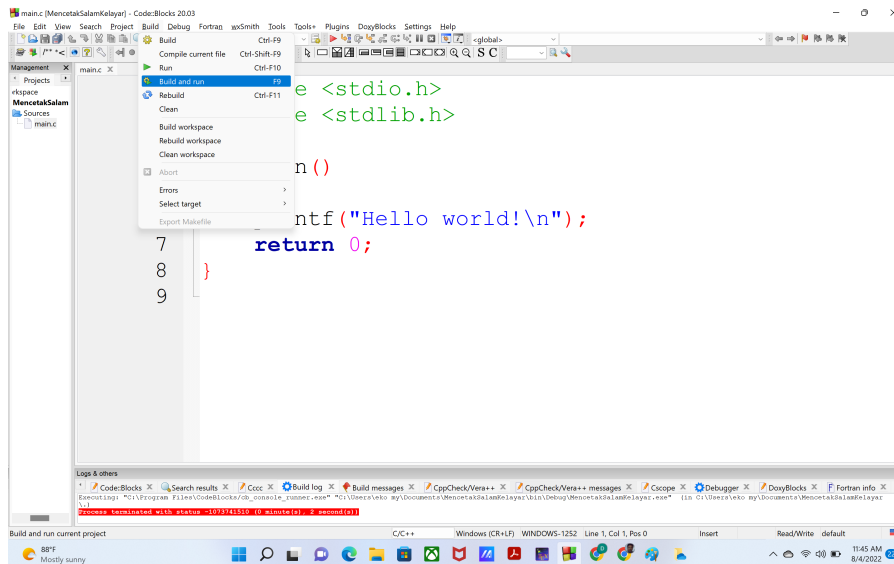
Figure 7

8. The program output can be seen on the console tab
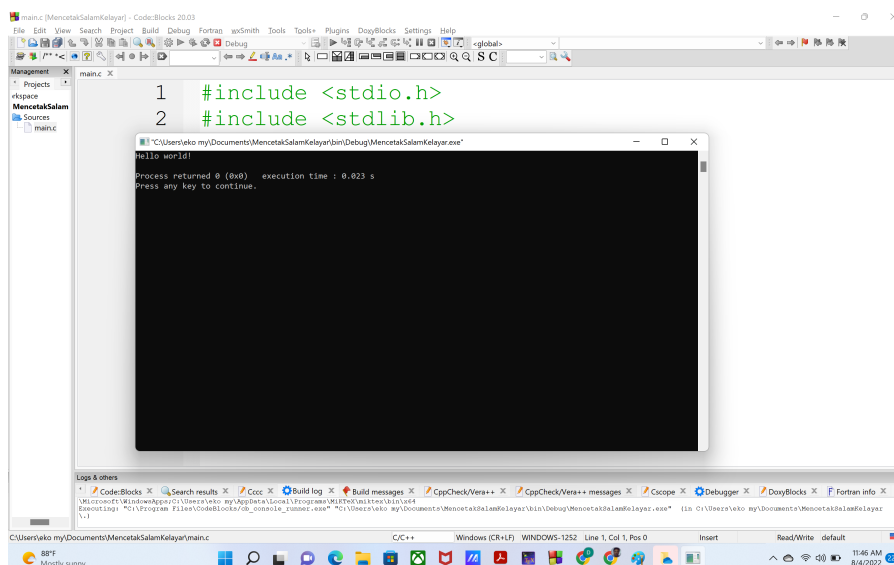
Figure 8

4.2 Pre-lab Assignment

1. Try looking for another IDE besides codeblocks, and explain the advantages and disadvantages of each IDE!

2. What makes C different from other programming languages? explain the advantages and disadvantages!

3. Create a project with name HaloDunia and write the program like can be seen on figure 6 but change `Hello World!` with other words!

5   Structure of C Programming Language

```c
#include <stdio.h>

int main()
{
  //printing to screen
  printf("Halo World");
  return 0;
}
```

Code on Listing 1 is a simple program to print "Halo Dunia" to screen. The following is the explanation what each line of code do in the program.

Row 1 : `#include <stdio.h>`
   header file library for input and output functions like `printf()` (the one used on line 6)

Row 2 : Empty line.

Row 3 : `int main()`
   The main function. The main function is the first function to be ran when the program starts.

Row 4 : {
   Beginning of the `main()` function code block.

Row 5 : `//printing to screen`
   Comments. Comments are used to explain what the program is doing. Comments are ignored by the program, but helps the reader.

Row 6 : `printf("Halo Dunia");`
   Printing "Halo Dunia" to the screen.

Row 7 : `return 0;`
   Returning the `main()` function (A function ends when it returns)

Row 8 : }
   Closing the `main()` function code block.

5.1 Pre-lab Assignment

1. Try to swap line 6 and line 7 in Listing 1.Explain what happened?

2. What if `return 0;` replaced with `return 1;`?

3. What happens if `return 0;` is deleted?

6 Data Types and Variable

6.1 Data Types

In C programming language, there are several data types to represent integer, real number, characters, string, and etc.

Table 1: Some data types in C programming language

Data Types	Size	Range Value
Int (or signed int)	2 bytes	-32,768 to 32,767
unsigned int	2 bytes	0 to 65,535
Short int(or signed short int)	2 bytes	-32,768 to 32,767
Long(or singed short int)	4 bytes	-2,147,483,648 to 2,147,483,647
unsigned long	4 bytes	0 to 4,294,967,295
float	4 bytes	1.2E-38 to 3.4E+38
double	8 bytes	2.3E-308 to 1.7E+308
Long double	10 bytes	3.4E-4932 to 1.1E+4932
char(or signed char)	1 byte	-128 to 127
unsigned char	1 byte	0 to 255

To show the data on screen, every data type has a format specifier that can be used on formatted string. The following is the format specifier for several data types.

Table 2: Format Specifier

Format Specifier	Data Types
%d or %i	int
%f	float
%lf	double
%c	char
%s	string

There are still more data types that what was written on Table 1. These data types and its specification can be found easily on the internet.

6.1.1 Modifier

In general, a modifier is a word, phrase, or clause that modifies or describes a noun or verb in a sentence. Meanwhile, in programming languages, a modifier is a keyword used to alter the behavior or characteristics of an element in a program, such as variables, functions, or classes. We use modifiers to change the range of basic data types to fit programming needs. There are four modifiers, namely:

1. signed
   `int value = -10;` (Can store integer type variables for negative and positive values)

2. unsigned
   `unsigned int count = 100;` (Can stroe integer type variables for positive value only)

3. long
   `long population = 7500000000;` (Using the long data type to store values larger than the int data type.)

4. short
   `short temperature = 20;` (Using the short data type to save memory when we know that the values to be stored will be relatively small.)

6.2   Variable

Variables are places to store data. Declaring a variable can be done in the following ways

Listing 2: C variable declaration

```
1  DataType VariableName;
```

6.2.1   Aithmetic and Assignment Operator

Operators are symbols that represent operations to be performed on one or more operands. Arithmetic Operators can be used to perform arithmetic operations such as addition, subtraction, etc. The Assignment Operator can be used to perform an operation on a variable and change the value of the variable according to the results of the operation.

Table 3: Arithmetic operator in C programming language

Operator	Name	Example
+	Addition	`x + y`
-	Subtraction	`x = y`
*	Multiplication	`x * y`
/	Distribution	`x/y`
%	Modulo	`x % y`

Table 4: Assignment operator

Operator	Example	Similiar meaning
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
&=	x &= 3	x = x & 3
|=	x |= 3	x = x | 3
^=	x ^= 3	x = x ^3
>>=	x >>= 3	x = x >>3
<<=	x <<= 3	x = x <<3

There are also 'abbreviations' for some assignment operators such as `x+=1` and `x-=1`, which `++` and `--`.

```
    x++;
    x--;
    ++x;
    --x;
```

6.2.2 Operator Bitwise

Bitwise operators are special operators used to handle logical operations on binary numbers in bit form. Binary numbers themselves are a type of number consisting of only two digits, which are 0 and 1. If the original value used is not in binary, it will be automatically converted by the C compiler into a binary number. For example, 7 in decimal equals 0111 in binary

Table 5: Bitwise operator

Operator	Name	Example	Binary	Result (binary)	Result (decimal)
&	AND	10 & 12	1010 & 1100	1000	8
|	OR	10 | 12	1010 | 1100	1110	14
^	XOR	10 ^12	1010 ^1100	0110	6
~	NOT	~10	~1010	0101	-11 (two complement)
<<	Left shift	10 <<1	1010 <<1	10100	20
>>	Right shift	10 >>1	1010 >>1	101	5

> **Notes:** There are several operators in the C language. Please study them by seeking references independently

6.3 Pre-lab Assignment

Listing 3: Using assignment operator in a const variable

```c
#include <stdio.h>
int main()
{

  //variable declaration
    const int x=0;
    x=1;
    printf(x);
    return 0;
}
```

1. Try to compile the program in Listing 3, what happened?

2. What must be done so that the output of Listing 3 is 1 ?

Listing 4: Using format specifier

```c
#include <stdio.h>
int main(){
    float a = 3.14;
    printf("%d", a);
}
```

3. Try to compile the program in Listing 4, what happened?

4. What must be done so that the program in Listing 4 has a score of 3.14?

7 Input and Output

7.1 printf()

`printf` is a function in C that is used to print formatted string. You can use format specifier `%` within the formatted string to outputs your variables.

```
printf(const char *format,v1,v2,..,vn)
```

The format specifier for each data types can be seen on Table 2

Contoh 7.1.1 Printing text to the screen.

Listing 5: Print text "C Programming" Ke layar

```c
#include <stdio.h>
int main()
{
  // Printing text inside the " symbol
  printf("C Programming");
  return 0;
}
```

- All C program must have main() function where the program needs to run the code.

- `printf()` function is a function from stdio.h library. This function outputs the string inside the symbol "" to the screen.

- `return 0;` statement in the `main()` function tells the program to exit.

Contoh 7.1.2 Printing integer.

```c
#include <stdio.h>
int main()
{
  int testInteger = 5;
  printf("Number = %d", testInteger); // <- %d format string
  return 0;
}
```

The code above uses the format specifier `%d` to prints `int` data type. The `%d` part of the string will be replaced with the value of `testInteger`.

Contoh 7.1.3 Real number output (float atau double)

- `Base` : using `float` data type.

- `Height`: using `float` data type.

- `Area` : using `float` data type.

$$Area = \frac{1}{2} \times Base \times Height \qquad (1)$$

10

```
1    #include <stdio.h>
2
3    int main()
4    {
5      // variable declaration
6      float Base;
7      float Height;
8      float Area;
9      // value initialization
10     Base = 10;
11     Height = 5;
12     // calculating area
13     Area = 0.5*Base*Height;
14     // printing the text to screen
15     printf("Area = %f",Area);
16     return 0;
17   }
18
19
```

explanation

Row 6-8 `Base`, `Height` and `Area` are `float`data type which use to store triangle area data.

Row 10 dan 11 Value initialization to `Base=10` and `Height=5`

Row 13 Calculating triangle area according to the equation 7.1

Row 15 Printing `Area` to the screen using `printf` command.

.

7.2 scanf

Function `scanf(const char *format, ...)` reads input according to the format string.

1. Syntax

   ```
   scanf(const char *format, ...)
   ```

2. Parameter
   Format string in C consist of one or more whitespace, non-whitespace, and format specifiers.

3. Return Value
   The function will return the value of arguments it has sucessfully read.

Example 7.2.4 Calculating triange `Base` dan tinggi `Height` yang diinputkan dari keyboard.
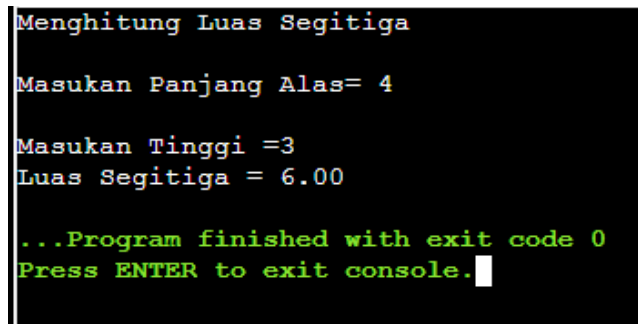
```
1 #include <stdio.h>
2
3 int main()
4 {
5   float Base, Height, Area;
```

```
 6
 7  printf("Calculate triangle area\n");
 8  printf("\Insert Base= ");
 9  scanf("%f",&Base);
10  printf("\nMasukkan Height=");
11  scanf("%f",&Height);
12  Area = 0.5*Base *Height;
13  printf("Triangle Area = %.2f", Area);
14  return 0;
15 }
16
```



```
Menghitung Luas Segitiga

Masukan Panjang Alas= 4

Masukan Tinggi =3
Luas Segitiga = 6.00

...Program finished with exit code 0
Press ENTER to exit console.
```

Figure 9

Row 9 `scanf("%f",&Area);` requesting input for triangle base

Row 11 `scanf("%f",&Height);` requesting input for triangle height

Row 13 `printf("Triangle Area = %.2f", Area);`, `.2` in `%.2f` indicating that only 2 digits after the decimal point need to be printed

Example 7.2.5 Program to input name and email.

This example shows how to input string or text from keyboard and outputs it on the screen. Input from this program consist of `sName` and `sEmail`. Because the text contains many characters, each variable is declared as an array of characters with the number of characters for sName=20 and sEmailAddress=30.

```
 1  #include <stdio.h>
 2
 3  int main ()
 4  {
 5    char sName[20], sEmail[30];
 6
 7    printf("Enter Name: ");
 8    scanf("%19s", sName);
 9
10    printf("Enter Email: ");
11    scanf("%29s", sEmail);
12
13    printf("Name : %s\n", sName);
14    printf("Email:%s", sEmail);
15    return(0);
16  }
17
```

12

7.3 Escape Sequence

Some characters can't be written on the format string because they are used to format the outputs.
So, to outputs those special characters we use escape sequences.

Table 6: Escape Sequence

Escape sequence	Output
\a	Bell, alarm
\b	Backspace
\f	Change Page
\n	Change Row
\r	Carriage return
\t	Tab Horizontal
\v	Tab Vertikal
\'	Single Quotes
\"	Double Quotes
\?	Question Mark
\\	Backslash

Contoh 7.3.6 Change row with escape sequence \n.

```
#include <stdio.h>

int main()
{
  printf("Halo \nI'm learning C programming language.\nand it's so fun!");
  return 0;
}
```
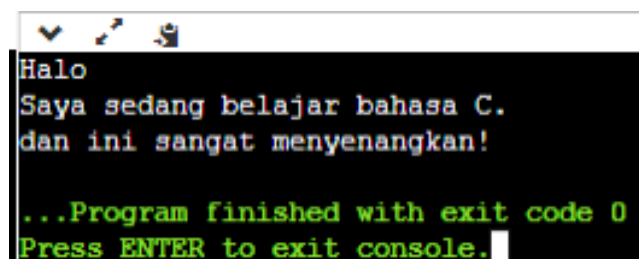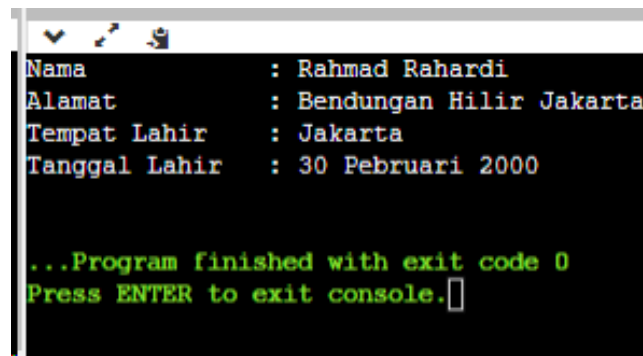

Figure 10

Contoh 7.3.7 Using escape sequence\t to change tab.

```
#include <stdio.h>
int main(void)
{
  printf("Name \t\t: Rahmad Rahardi\n");
  printf("Address \t\t: Bendungan Hilir Jakarta\n");
  printf("Place of Birth \t: Jakarta\n");
  printf("Date of Birth \t: 30 February 2000\n");

  return (0);
}
```

Figure 11

7.4 Pre-lab Assignment

1. `scanf("%d", &height)`, the code will ask for input to be entered in the variable `height`. Explain why there is "&" on `&height`! what happens if it is simply written as `height` whitout &!

2. Try to build a program to read input for the name and NRP and display it to the screen!

3. Try to build a program that asks user to enter a number, then displays the squared dan cubed value of it!

> **Optional:** Learn Git and Github. You can start learning from the following resources:
> GitHub - https://github.com
> Git -https://git-scm.com/doc