

**TUGAS 4**  
**BAHASA PEMROGRAMAN ASSEMBLY**



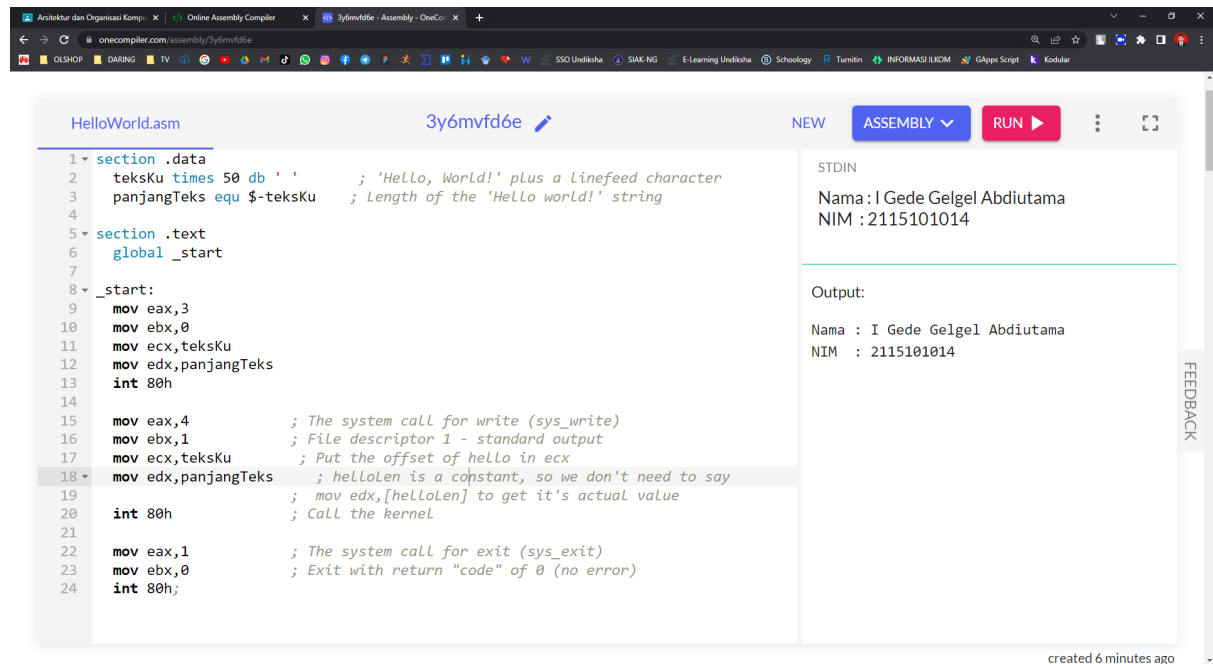
**Dosen Pengampu :**  
**I Ketut Purnamawan, S.Kom., M.Kom.**

**Disusun Oleh :**  
**I Gede Gelgel Abdiutama ; 2115101014**

**MATA KULIAH ARSITEKTUR DAN ORGANISASI KOMPUTER**  
**UNIVERSITAS PENDIDIKAN GANESHA**  
**SINGARAJA**  
**TA. 2022**

## A. KEGIATAN BELAJAR

### Percobaan 1



The screenshot shows a web browser with the 'Online Assembly Compiler' open. The file 'HelloWorld.asm' is loaded. The code is as follows:

```
1 section .data
2 teksKu times 50 db ' ' ; 'Hello, World!' plus a linefeed character
3 panjangTeks equ $-teksKu ; Length of the 'Hello world!' string
4
5 section .text
6 global _start
7
8 _start:
9 mov eax,3
10 mov ebx,0
11 mov ecx,teksKu
12 mov edx,panjangTeks
13 int 80h
14
15 mov eax,4 ; The system call for write (sys_write)
16 mov ebx,1 ; File descriptor 1 - standard output
17 mov ecx,teksKu ; Put the offset of hello in ecx
18 mov edx,panjangTeks ; helloLen is a constant, so we don't need to say
19 ; mov edx,[helloLen] to get it's actual value
20 int 80h ; Call the kernel
21
22 mov eax,1 ; The system call for exit (sys_exit)
23 mov ebx,0 ; Exit with return "code" of 0 (no error)
24 int 80h;
```

The right panel shows the 'STDIN' input as 'Nama : I Gede Gelgel Abdiutama' and 'NIM : 2115101014'. The 'Output' panel shows the same text: 'Nama : I Gede Gelgel Abdiutama' and 'NIM : 2115101014'.

#### Kode Program :

section .data

```
teksKu times 50 db ' ' ; 'Hello, World!' plus a linefeed character
panjangTeks equ $-teksKu ; Length of the 'Hello world!' string
```

section .text

```
global _start
```

\_start:

```
mov eax,3
```

```
mov ebx,0
```

```
mov ecx,teksKu
```

```
mov edx,panjangTeks
```

```
int 80h
```

```
mov eax,4 ; The system call for write (sys_write)
```

```
mov ebx,1 ; File descriptor 1 - standard output
```

```
mov ecx,teksKu ; Put the offset of hello in ecx
```

```
mov edx,panjangTeks ; helloLen is a constant, so we don't need to say
```

```
; mov edx,[helloLen] to get it's actual value
```

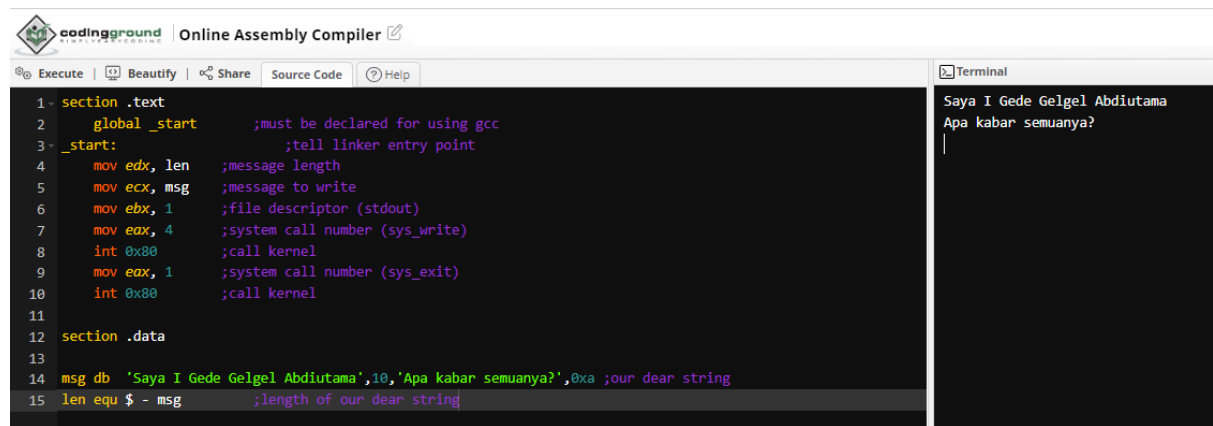
```
int 80h ; Call the kernel
```

```
mov eax,1 ; The system call for exit (sys_exit)
```

```
mov ebx,0 ; Exit with return "code" of 0 (no error)
```

```
int 80h;
```

## Percobaan 2



The screenshot shows the 'Online Assembly Compiler' interface. The main editor contains assembly code for a program that prints a message and exits. The code is as follows:

```
1 section .text
2 global _start ;must be declared for using gcc
3 _start: ;tell linker entry point
4 mov edx, len ;message length
5 mov ecx, msg ;message to write
6 mov ebx, 1 ;file descriptor (stdout)
7 mov eax, 4 ;system call number (sys_write)
8 int 0x80 ;call kernel
9 mov eax, 1 ;system call number (sys_exit)
10 int 0x80 ;call kernel
11
12 section .data
13
14 msg db 'Saya I Gede Gelgel Abdiutama',10,'Apa kabar semuanya?',0xa ;our dear string
15 len equ $ - msg ;length of our dear string
```

The terminal on the right shows the output of the program:

```
Saya I Gede Gelgel Abdiutama
Apa kabar semuanya?
```

### Code Program :

```
section .text
    global _start ;must be declared for using gcc
_start:
    ;tell linker entry point
    mov     edx, len ;message length
    mov     ecx, msg ;message to write
    mov     ebx, 1 ;file descriptor (stdout)
    mov     eax, 4 ;system call number (sys_write)
    int     0x80 ;call kernel
    mov     eax, 1 ;system call number (sys_exit)
    int     0x80 ;call kernel

section .data

msg     db     'Saya I Gede Gelgel Abdiutama',10,'Apa kabar semuanya?',0xa ;our dear string
len     equ    $ - msg ;length of our dear string
```