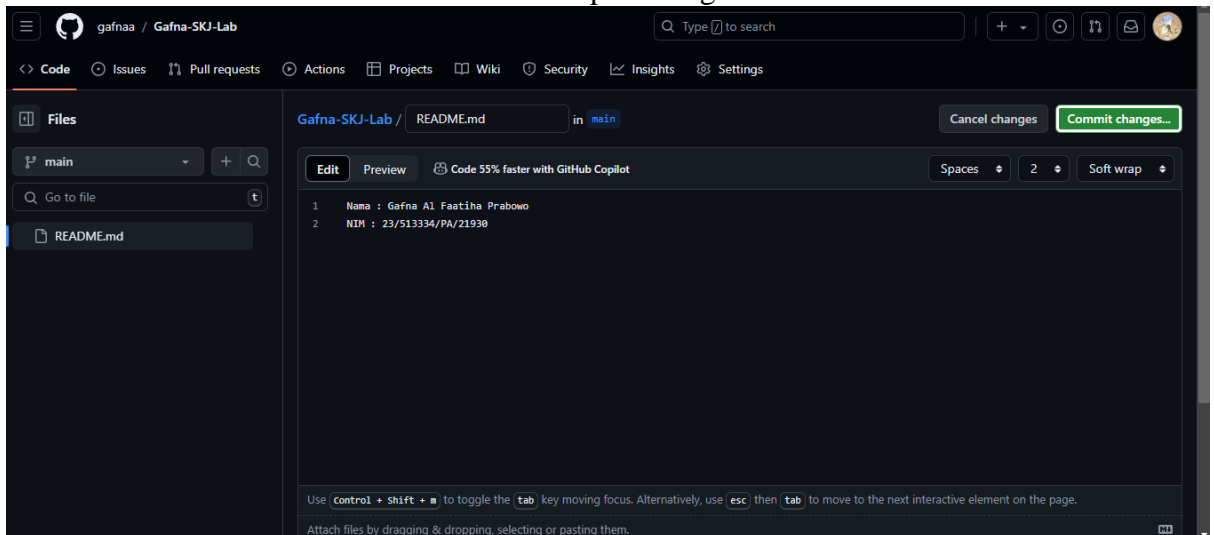
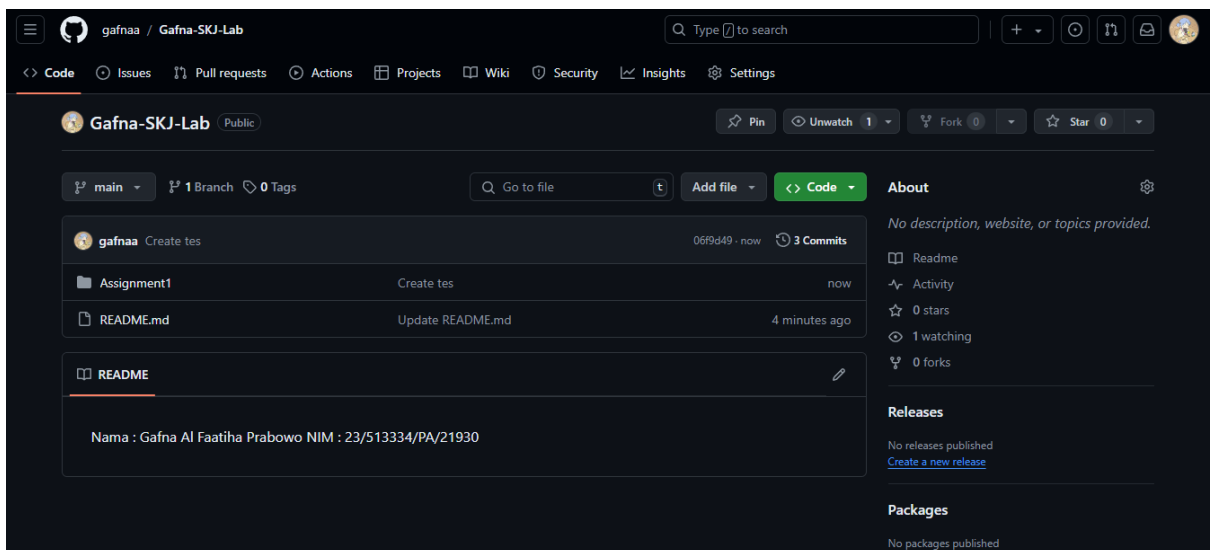


## Tugas 1 Prak-SKJ

1. Membuat akun Github dan melakukan beberapa settingan.



Membuat repositori, lalu membuat file readme.md yang berisi nama dan nim saya.



Membuat folder Assignment1 pada repositori.

Link github : <https://github.com/gafnaa/Gafna-SKJ-Lab/>

2. Tugas C++ ke Assembly
  - Membuat Program C++ sederhana

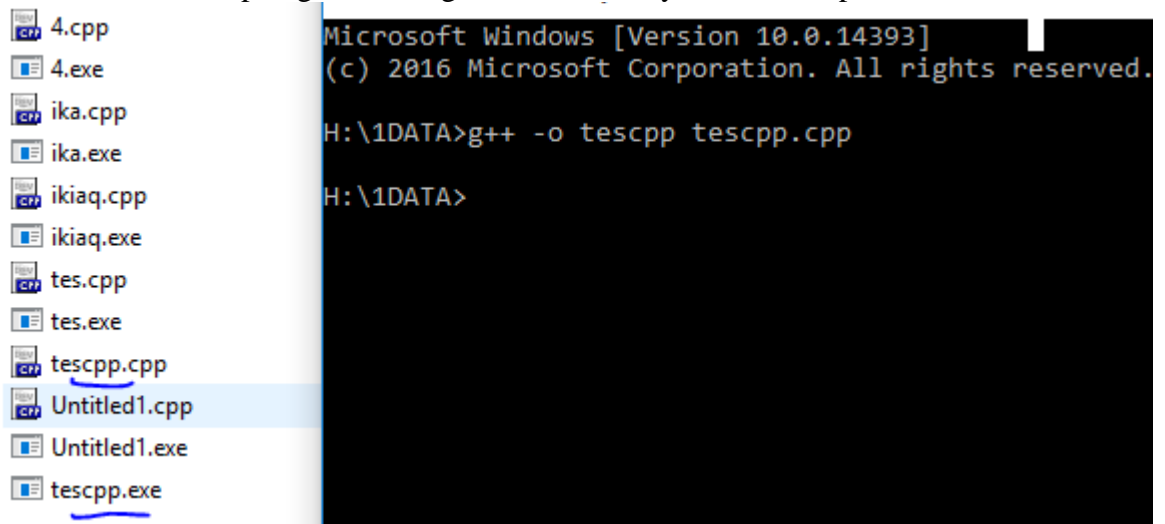
```
#include <iostream>
using namespace std;
int main()
{
    int a,b;
    cin >> a >> b;
    int c = a+b;
    cout << c;
    return 0;
}
```

Contoh output :

```
1 2
3

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

- Melakukan compiling code dengan menuliskan syntax berikut pada terminal



- Melakukan diassemble code, dengan menuliskan syntax berikut

H:\DATA>objdump -d tescpp.exe

tescpp.exe: file format pei-x86-64

Disassembly of section .text:

0000000140001000 <\_mingw\_invalidParameterHandler>:

```
140001000: c3          ret
140001001: 0f 1f 44 00 00    nopl 0x0(%rax,%rax,1)
140001006: 66 2e 0f 1f 84 00 00 cs nopw 0x0(%rax,%rax,1)
14000100d: 00 00 00
```

0000000140001010 <pre\_c\_init>:

```
140001010: 48 83 ec 28      sub $0x28,%rsp
140001014: 48 8b 05 d5 33 00 00 mov 0x33d5(%rip),%rax # 1400043f0 <.refptr._mingw_initiltsdrot_force>
14000101b: 31 c9           xor %ecx,%ecx
14000101d: c7 00 01 00 00 00 movl $0x1,%rax
140001023: 48 8b 05 d6 33 00 00 mov 0x33d6(%rip),%rax # 140004400 <.refptr._mingw_initiltsdyn_force>
14000102a: c7 00 01 00 00 00 movl $0x1,%rax
140001030: 48 8b 05 d9 33 00 00 mov 0x33d9(%rip),%rax # 140004410 <.refptr._mingw_initiltsuoo_force>
140001037: c7 00 01 00 00 00 movl $0x1,%rax
14000103d: 48 8b 05 4c 33 00 00 mov 0x334c(%rip),%rax # 140004390 <.refptr._image_base_>
140001044: 66 81 38 4d 5a    cmpl $0x5a4d,%rax
140001049: 75 0f          jne 14000105a <pre_c_init+0x4a>
14000104b: 48 63 50 3c      movslq 0x3c(%rax),%rdx
14000104f: 48 01 d0        add %rdx,%rax
140001052: 81 38 50 45 00 00 cmpl $0x4550,%rax
14000105d: 74 66          jle 1400010c9 <pre_c_init+0xb0>
14000105a: 48 8b 05 7f 33 00 00 mov 0x337f(%rip),%rax # 1400043e0 <.refptr._mingw_app_type>
140001061: 89 0d a5 5f 00 00 mov %ecx,0x5fa5(%rip) # 14000700c <managedapp>
140001067: 8b 00          mov (%rax),%eax
140001069: 85 00          test %eax,%eax
14000106b: 75 43          jne 1400010b0 <pre_c_init+0xa0>
14000106d: b9 01 00 00 00 00 mov $0x1,%ecx
140001072: e8 31 17 00 00   call 140002838 <__set_app_type>
140001077: e8 34 17 00 00   call 1400027b0 <__p_fmode>
14000107c: 48 8b 15 2d 34 00 00 mov 0x342d(%rip),%rdx # 1400044b0 <.refptr._fmode>
140001083: 8b 12          mov (%rdx),%edx
140001085: 89 10          mov %edx,%rax
140001087: e8 34 17 00 00   call 1400027c0 <__p_commode>
14000108c: 48 8b 15 fd 33 00 00 mov 0x33fd(%rip),%rdx # 140004490 <.refptr._commode>
140001093: 8b 12          mov (%rdx),%edx
140001095: 89 10          mov %edx,%rax
140001097: e8 64 06 00 00   call 140001700 <setargv>
14000109c: 48 8b 05 7d 32 00 00 mov 0x327d(%rip),%rax # 140004320 <.refptr._MINGW_INSTALL_DEBUG_MATHERR>
1400010a3: 83 38 01        cmpl $0x1,%rax
1400010a6: 74 49          je 1400010f1 <pre_c_init+0xe1>
1400010a8: 31 c0          xor %eax,%eax
1400010aa: 48 83 c4 28      add $0x28,%rsp
1400010ae: c3          ret
1400010af: 90          nop
```

```
1400010a8: 31 c0          xor %eax,%eax
1400010aa: 48 83 c4 28      add $0x28,%rsp
1400010ae: c3          ret
1400010af: 90          nop
1400010b0: b9 02 00 00 00 00 mov $0x2,%ecx
1400010b5: e8 7e 17 00 00   call 140002838 <__set_app_type>
1400010ba: eb bb          jmp 140001077 <pre_c_init+0x67>
1400010bc: 0f 1f 40 00      nopl 0x0(%rax)
1400010c0: 0f b7 50 18      movzwl 0x18(%rax),%edx
1400010c4: 66 81 fa 0b 01   cmp $0x10b,%dx
1400010c9: 74 39          je 140001104 <pre_c_init+0xf4>
1400010cb: 66 81 fa 0b 02   cmp $0x20b,%dx
1400010d0: 75 88          jne 14000105a <pre_c_init+0x4a>
1400010d2: 83 b8 84 00 00 00 0e cmpl $0xe,0x84(%rax)
1400010d9: 0f 86 7b ff ff ff jbe 14000105a <pre_c_init+0x4a>
1400010df: 8b 90 f8 00 00 00 mov 0xf8(%rax),%edx
1400010e5: 31 c9          xor %ecx,%ecx
1400010e7: 85 d2          test %edx,%edx
1400010e9: 0f 95 c1        setne %cl
1400010ec: e9 69 ff ff ff  jmp 14000105a <pre_c_init+0x4a>
1400010f1: 48 8d 0d d8 06 00 00 lea 0x6d8(%rip),%rcx # 1400071d0 <_matherr>
1400010f8: e8 13 0e 00 00   call 140001f10 <__mingw_setusermatherr>
1400010fd: 31 c0          xor %eax,%eax
1400010ff: 48 83 c4 28      add $0x28,%rsp
140001103: c3          ret
140001104: 83 78 74 0e      cmpl $0xe,0x74(%rax)
140001108: 0f 86 4c ff ff ff jbe 14000105a <pre_c_init+0x4a>
14000110e: 44 8b 80 e8 00 00 00 mov 0xe8(%rax),%r8d
140001115: 31 c9          xor %ecx,%ecx
140001117: 45 85 c0        test %r8d,%r8d
14000111a: 0f 95 c1        setne %cl
14000111d: e9 38 ff ff ff  jmp 14000105a <pre_c_init+0x4a>
140001122: 0f 1f 40 00      nopl 0x0(%rax)
140001126: 66 2e 0f 1f 84 00 00 cs nopw 0x0(%rax,%rax,1)
14000112d: 00 00 00
```

0000000140001130 <pre\_cpp\_init>:

```
140001130: 48 83 ec 38      sub $0x38,%rsp
140001134: 48 8b 05 a5 33 00 00 mov 0x33a5(%rip),%rax # 1400044e0 <.refptr._newmode>
14000113b: 4c 8d 05 d6 5e 00 00 lea 0x5ed6(%rip),%r8 # 140007018 <envp>
140001142: 48 8d 15 d7 5e 00 00 lea 0x5ed7(%rip),%rdx # 140007020 <argv>
140001149: 48 8d 0d d8 5e 00 00 lea 0x5ed8(%rip),%rcx # 140007028 <argc>
140001150: 8b 00          mov (%rax),%eax
140001152: 89 05 ac 5e 00 00 mov %eax,0x5eac(%rip) # 140007004 <startinfo>
140001158: 48 8d 05 a5 5e 00 00 lea 0x5ea5(%rip),%rax # 140007004 <startinfo>
14000115f: 48 89 44 24 20   mov %rax,0x20(%rsp)
140001164: 48 8b 05 35 33 00 00 mov 0x3335(%rip),%rax # 1400044a0 <.refptr._dowildcard>
14000116b: 44 8b 08        mov (%rax),%r8d
14000116e: e8 b5 16 00 00   call 140002828 <__getmainargs>
140001173: 90          nop
140001174: 48 83 c4 38      add $0x38,%rsp
140001178: c3          ret
140001179: 0f 1f 80 00 00 00 00 nopl 0x0(%rax)
```

0000000140001180 <\_tmainCRTStartup>:

```
140001180: 41 55          push %r13
140001182: 41 54          push %r12
140001184: 55          push %rbp
140001185: 57          push %rdi
140001186: 56          push %rsi
140001187: 53          push %rbx
140001188: 48 81 ec 98 00 00 00 sub $0x98,%rsp
14000118f: 31 c0          xor %eax,%eax
```

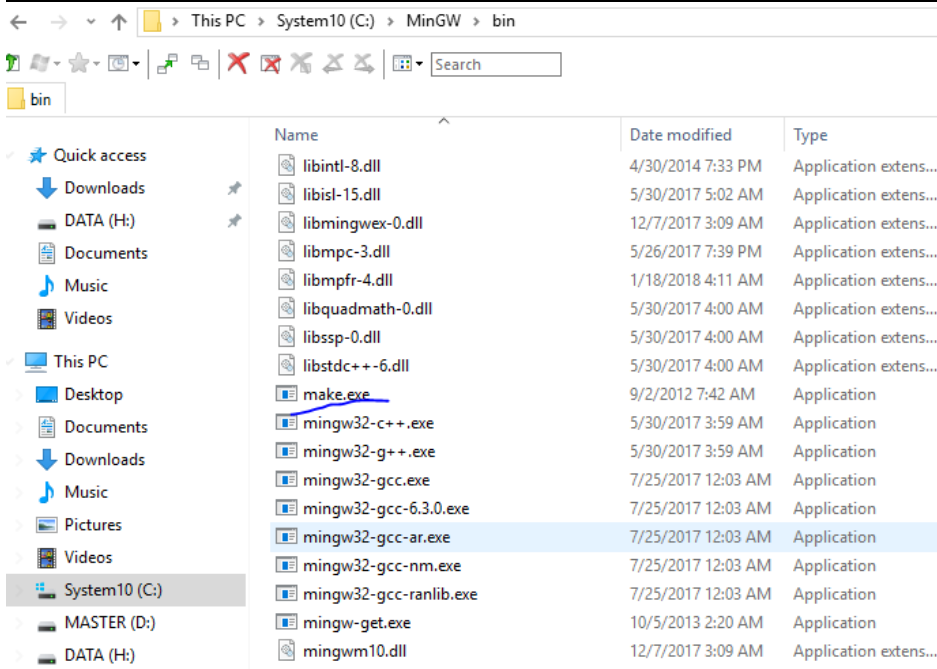
- Membuat Makefile
  - 1) Menginstall 'make' command

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

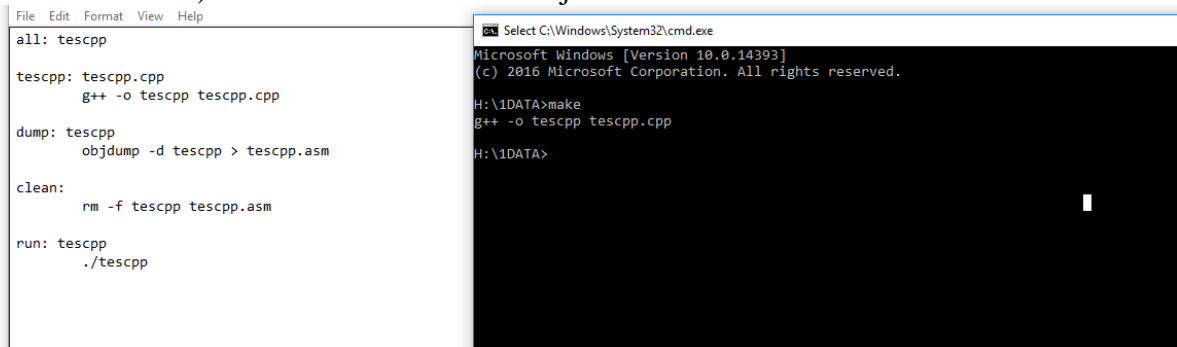
C:\Users\praktikan>mingw-get

C:\Users\praktikan>mingw-get install mingw32-make
http://prdownloads.sourceforge.net/mingw/make-3.82.90-2-mingw32-cvs-20120902-doc.tar.lzma?download
83.46 kB / 183.46 kB |=====| 100%
http://prdownloads.sourceforge.net/mingw/make-3.82.90-2-mingw32-cvs-20120902-lic.tar.lzma?download
11.23 kB / 11.23 kB |=====| 100%
http://prdownloads.sourceforge.net/mingw/make-3.82.90-2-mingw32-cvs-20120902-lang.tar.lzma?download
90.81 kB / 90.81 kB |=====| 100%
install: make-3.82.90-2-mingw32-cvs-20120902-bin.tar.lzma
mingw-get: *** ERROR *** package make-3.82.90-2-mingw32-cvs-20120902-bin.tar.lzma is already installed
install: make-3.82.90-2-mingw32-cvs-20120902-doc.tar.lzma
installing make-3.82.90-2-mingw32-cvs-20120902-doc.tar.lzma
install: make-3.82.90-2-mingw32-cvs-20120902-lic.tar.lzma
installing make-3.82.90-2-mingw32-cvs-20120902-lic.tar.lzma
install: make-3.82.90-2-mingw32-cvs-20120902-lang.tar.lzma
installing make-3.82.90-2-mingw32-cvs-20120902-lang.tar.lzma

C:\Users\praktikan>
```



## 2) Membuat makefile lalu dijalankan



### 3. Assembly ke C++

- Analisis kode assembly

```
section .data
    num1 dw 5
    num2 dw 10
    result dw 0

section .text
    global _start

_start:
    mov ax, [num1]
    imul ax, [num2]
    mov [result], ax

    ; Exit the program
    mov eax, 1
    xor ebx, ebx
    int 0x80
```

Deskripsi dan penjelasan tiap line

- **section .data** = sebagai tanda bahwa bagian tersebut akan digunakan untuk mendeklarasikan variabel
- **num1 dw 5** = menginisialisasi nilai 5 ke variabel *num1* dan mengalokasikannya ke memori. (dw = define word, menisialisasi data 2-byte)
- **num2 dw 10** = menginisialisasi nilai 10 ke variabel *num2* dan mengalokasikannya ke memori. (dw = define word, menisialisasi data 2-byte)
- **result dw 0** = menginisialisasi nilai 0 ke variabel *result* dan mengalokasikannya ke memori. (dw = define word, menisialisasi data 2-byte)
- **section .text** = sebagai tanda bahwa bagian tersebut akan digunakan untuk menyimpan instruksi program yang akan dieksekusi CPU
- **global \_start** = mendeklarasikan label *\_start*
- **\_start** = titik awal eksekusi program
- **mov ax, [num1]** = memindahkan nilai di variabel *num1*(5) ke register *ax*
- **imul ax, [num2]** = mengalikan nilai di register *ax*(5) dengan nilai di variabel *num2*(10)
- **mov [result], ax** = memindahkan nilai di register *ax* ke variabel *result*
- **;Exit the program** = tanda bahwa bagian tersebut berisi instruksi-instruksi yang digunakan untuk menghentikan eksekusi program
- **mov eax, 1** = memindahkan nilai 1 ke register *eax*
- **xor ebx, ebx** = melakukan operasi xor antara *ebx* dengan *ebx* juga
- **int 0x80** = memanggil interrupt 0x80 untuk mengesekusi perintah exit

- Menulis kode C++ yang ekuivalen

```
#include <iostream>

int main() {
    int16_t num1 = 5;
    int16_t num2 = 10;
    int16_t result = 0;

    result = num1 * num2;

    // Exit the program
    return 0;
}
```

Di sini saya menggunakan tipe data `int16_t` yang artinya hanya mendefinisikan data integer yang bernilai 16-bit(2-byte)

- Buat makefile

