Assignment 1 Prak SKJ
Rayhan Firdaus Ardian
23/519095/PA/22279

Link github : https://github.com/HappyRehund/Rayhan-SKJ-Lab/tree/main

1.6.2 C++ Code to Assembly

1. Simple C++ program that add 2 num

test.cpp mul.cpp

test.cpp > main()

```cpp
#include <iostream>
using namespace std;

int main()
{
    int num1, num2, sum;
    num1 = 10;
    num2 = 20;
    sum = num1 + num2;

    cout << "Hasil penjumlahan dari " << num1 << " dan " << num2 << " adalah " << sum << endl ;
    return 0;
}
```

2. Compile the code

```
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ ls
test  test.cpp
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ g++ -o test test.cpp
```

3. Disassemble the code

```
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ ls
test  test.cpp
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ g++ -o test test.cpp
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ objdump -d test

test:     file format elf64-x86-64


Disassembly of section .init:

0000000000001000 <_init>:
    1000:       f3 0f 1e fa             endbr64
    1004:       48 83 ec 08             sub    $0x8,%rsp
    1008:       48 8b 05 e1 2f 00 00    mov    0x2fe1(%rip),%rax        # 3ff0 <__gmon_start__@Base>
    100f:       48 85 c0                test   %rax,%rax
    1012:       74 02                   je     1016 <_init+0x16>
    1014:       ff d0                   call   *%rax
    1016:       48 83 c4 08             add    $0x8,%rsp
    101a:       c3                      ret

Disassembly of section .plt:

0000000000001020 <.plt>:
    1020:       ff 35 9a 2f 00 00       push   0x2f9a(%rip)        # 3fc0 <_GLOBAL_OFFSET_TABLE_+0x8>
    1026:       ff 25 9c 2f 00 00       jmp    *0x2f9c(%rip)        # 3fc8 <_GLOBAL_OFFSET_TABLE_+0x10>
    102c:       0f 1f 40 00             nopl   0x0(%rax)
    1030:       f3 0f 1e fa             endbr64
    1034:       68 00 00 00 00          push   $0x0
    1039:       e9 e2 ff ff ff          jmp    1020 <_init+0x20>
    103e:       66 90                   xchg   %ax,%ax
```

```
Disassembly of section .plt.got:

0000000000001040 <__cxa_finalize@plt>:
    1040:       f3 0f 1e fa             endbr64
    1044:       ff 25 8e 2f 00 00       jmp    *0x2f8e(%rip)        # 3fd8 <__cxa_finalize@GLIBC_2.2.5>
    104a:       66 0f 1f 44 00 00       nopw   0x0(%rax,%rax,1)

Disassembly of section .plt.sec:

0000000000001050 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@plt>:
    1050:       f3 0f 1e fa             endbr64
    1054:       ff 25 76 2f 00 00       jmp    *0x2f76(%rip)        # 3fd0 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_
PKc@GLIBCXX_3.4>
    105a:       66 0f 1f 44 00 00       nopw   0x0(%rax,%rax,1)

Disassembly of section .text:

0000000000001060 <_start>:
    1060:       f3 0f 1e fa             endbr64
    1064:       31 ed                   xor    %ebp,%ebp
    1066:       49 89 d1                mov    %rdx,%r9
    1069:       5e                      pop    %rsi
    106a:       48 89 e2                mov    %rsp,%rdx
    106d:       48 83 e4 f0             and    $0xfffffffffffffff0,%rsp
    1071:       50                      push   %rax
    1072:       54                      push   %rsp
    1073:       45 31 c0                xor    %r8d,%r8d
    1076:       31 c9                   xor    %ecx,%ecx
    1078:       48 8d 3d ca 00 00 00    lea    0xca(%rip),%rdi        # 1149 <main>
    107f:       ff 15 5b 2f 00 00       call   *0x2f5b(%rip)        # 3fe0 <__libc_start_main@GLIBC_2.34>
    1085:       f4                      hlt
    1086:       66 2e 0f 1f 84 00 00    cs nopw 0x0(%rax,%rax,1)
    108d:       00 00 00

0000000000001090 <deregister_tm_clones>:
    1090:       48 8d 3d 79 2f 00 00    lea    0x2f79(%rip),%rdi        # 4010 <__TMC_END__>
    1097:       48 8d 05 72 2f 00 00    lea    0x2f72(%rip),%rax        # 4010 <__TMC_END__>
    109e:       48 39 f8                cmp    %rdi,%rax
    10a1:       74 15                   je     10b8 <deregister_tm_clones+0x28>
    10a3:       48 8b 05 3e 2f 00 00    mov    0x2f3e(%rip),%rax        # 3fe8 <_ITM_deregisterTMCloneTable@Base>
    10aa:       48 85 c0                test   %rax,%rax
    10ad:       74 09                   je     10b8 <deregister_tm_clones+0x28>
    10af:       ff e0                   jmp    *%rax
    10b1:       0f 1f 80 00 00 00 00    nopl   0x0(%rax)
    10b8:       c3                      ret
    10b9:       0f 1f 80 00 00 00 00    nopl   0x0(%rax)

00000000000010c0 <register_tm_clones>:
    10c0:       48 8d 3d 49 2f 00 00    lea    0x2f49(%rip),%rdi        # 4010 <__TMC_END__>
    10c7:       48 8d 35 42 2f 00 00    lea    0x2f42(%rip),%rsi        # 4010 <__TMC_END__>
    10ce:       48 29 fe                sub    %rdi,%rsi
    10d1:       48 89 f0                mov    %rsi,%rax
    10d4:       48 c1 ee 3f             shr    $0x3f,%rsi
    10d8:       48 c1 f8 03             sar    $0x3,%rax
    10dc:       48 01 c6                add    %rax,%rsi
    10df:       48 d1 fe                sar    $1,%rsi
    10e2:       74 14                   je     10f8 <register_tm_clones+0x38>
    10e4:       48 8b 05 0d 2f 00 00    mov    0x2f0d(%rip),%rax        # 3ff8 <_ITM_registerTMCloneTable@Base>
    10eb:       48 85 c0                test   %rax,%rax
    10ee:       74 08                   je     10f8 <register_tm_clones+0x38>
    10f0:       ff e0                   jmp    *%rax
    10f2:       66 0f 1f 44 00 00       nopw   0x0(%rax,%rax,1)
    10f8:       c3                      ret
    10f9:       0f 1f 80 00 00 00 00    nopl   0x0(%rax)

0000000000001100 <__do_global_dtors_aux>:
    1100:       f3 0f 1e fa             endbr64
    1104:       80 3d 45 30 00 00 00    cmpb   $0x0,0x3045(%rip)        # 4150 <completed.0>
    110b:       75 2b                   jne    1138 <__do_global_dtors_aux+0x38>
    110d:       55                      push   %rbp
    110e:       48 83 3d c2 2e 00 00    cmpq   $0x0,0x2ec2(%rip)        # 3fd8 <__cxa_finalize@GLIBC_2.2.5>
    1115:       00
    1116:       48 89 e5                mov    %rsp,%rbp
    1119:       74 0c                   je     1127 <__do_global_dtors_aux+0x27>
    111b:       48 8b 3d e6 2e 00 00    mov    0x2ee6(%rip),%rdi        # 4008 <__dso_handle>
    1122:       e8 19 ff ff ff          call   1040 <__cxa_finalize@plt>
    1127:       e8 64 ff ff ff          call   1090 <deregister_tm_clones>
    112c:       c6 05 1d 30 00 00 01    movb   $0x1,0x301d(%rip)        # 4150 <completed.0>
    1133:       5d                      pop    %rbp
    1134:       c3                      ret
    1135:       0f 1f 00                nopl   (%rax)
    1138:       c3                      ret
    1139:       0f 1f 80 00 00 00 00    nopl   0x0(%rax)

0000000000001140 <frame_dummy>:
    1140:       f3 0f 1e fa             endbr64
    1144:       e9 77 ff ff ff          jmp    10c0 <register_tm_clones>

0000000000001149 <main>:
    1149:       f3 0f 1e fa             endbr64
    114d:       55                      push   %rbp
    114e:       48 89 e5                mov    %rsp,%rbp
    1151:       48 83 ec 10             sub    $0x10,%rsp
```

```
    1155:       c7 45 f4 0a 00 00 00    movl   $0xa,-0xc(%rbp)
    115c:       c7 45 f8 14 00 00 00    movl   $0x14,-0x8(%rbp)
    1163:       8b 55 f4                mov    -0xc(%rbp),%edx
    1166:       8b 45 f8                mov    -0x8(%rbp),%eax
    1169:       01 d0                   add    %edx,%eax
    116b:       89 45 fc                mov    %eax,-0x4(%rbp)
    116e:       48 8d 05 8f 0e 00 00    lea    0xe8f(%rip),%rax        # 2004 <_IO_stdin_used+0x4>
    1175:       48 89 c6                mov    %rax,%rsi
    1178:       48 8d 05 c1 2e 00 00    lea    0x2ec1(%rip),%rax        # 4040 <_ZSt4cout@GLIBCXX_3.4>
    117f:       48 89 c7                mov    %rax,%rdi
    1182:       e8 c9 fe ff ff          call   1050 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@plt>
    1187:       b8 00 00 00 00          mov    $0x0,%eax
    118c:       c9                      leave
    118d:       c3                      ret

Disassembly of section .fini:

0000000000001190 <_fini>:
    1190:       f3 0f 1e fa             endbr64
    1194:       48 83 ec 08             sub    $0x8,%rsp
    1198:       48 83 c4 08             add    $0x8,%rsp
    119c:       c3                      ret
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$
```

4. Write a makefile

```
  GNU nano 7.2                                                batch.sh *
#!/bin/bash

# Compiles the code
compile() {
    g++ -o test test.cpp
}

# Disassembles the compiled code
disassemble() {
    objdump -d test > test.asm
}

# Removes all output files created by compile and disassemble functions
clean() {
    rm -f test test.asm
}

# Runs the compiled executable
run() {
    ./test
}
```

```
# Main function to process command-line arguments
main() {
    case "$1" in
        all)
            compile
            ;;
        dump)
            disassemble
            ;;
        clean)
            clean
            ;;
        run)
            run
            ;;
        *)
            echo "Usage: $0 {all|dump|clean|run}"
            exit 1
            ;;
    esac
}

# Call the main function with all arguments passed to the script
main "$@"
```

Hasil :

```
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ ./batch.sh all
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ ./batch.sh dump
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ ls
batch.sh  test  test.asm  test.cpp
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ cat test.asm

test:     file format elf64-x86-64


Disassembly of section .init:

0000000000001000 <_init>:
    1000:       f3 0f 1e fa             endbr64
    1004:       48 83 ec 08             sub    $0x8,%rsp
    1008:       48 8b 05 e1 2f 00 00    mov    0x2fe1(%rip),%rax        # 3ff0 <__gmon_start__@Base>
    100f:       48 85 c0                test   %rax,%rax
    1012:       74 02                   je     1016 <_init+0x16>
    1014:       ff d0                   call   *%rax
    1016:       48 83 c4 08             add    $0x8,%rsp
    101a:       c3                      ret

Disassembly of section .plt:

0000000000001020 <.plt>:
    1020:       ff 35 9a 2f 00 00       push   0x2f9a(%rip)        # 3fc0 <_GLOBAL_OFFSET_TABLE_+0x8>
    1026:       ff 25 9c 2f 00 00       jmp    *0x2f9c(%rip)        # 3fc8 <_GLOBAL_OFFSET_TABLE_+0x10>
    102c:       0f 1f 40 00             nopl   0x0(%rax)
    1030:       f3 0f 1e fa             endbr64
    1034:       68 00 00 00 00          push   $0x0
```

```
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ ./batch.sh all
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ ./batch.sh run
Hasil penjumlahan dari 10 dan 20 adalah 30
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ ./batch.sh dump
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ ls
batch.sh  mul  mul.asm  mul.cpp  mul.sh  test  test.asm  test.cpp
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ ./batch.sh clean
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ ls
batch.sh  mul  mul.asm  mul.cpp  mul.sh  test.cpp
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$
```

1.6.3 Assembly to C++

1. Analyze the Provided Assembly Code

section .data -> “deklarasikan variabel atau data (tdk akan berubah)”
num1 dw 5 -> “deklarasi variabel `num1` (beri nilai 5 dg size 2 byte)”
num2 dw 10 -> “deklarasi variabel `num2`(beri nilai 10 dg size 2 byte)”
result dw 0 -> “deklarasikan variabel `result` dengan nilai awal 0”

section .text -> “baris ini menunjukkan bahwa bagian kode program didefinisikan di bagian .text”
global _start -> “Menandakan bahwa label _start dapat diakses dari luar (global), serta titik awal eksekusi”

_start: -> “awal eksekusi program”

mov ax, [num1] -> “ambil nilai dari `num 1` lalu simpan dalam registe `ax`”
imul ax, [num2] -> “operasi perkalian antara nilai di register `ax` degan `num 2`”
mov [result], ax -> “pindah hasil perkalian dari register `ax` ke variabel `result`”

; Exit the program
mov eax, 1 -> “simpan nilai 1 ke register `eax` (system call untuk exit program)
xor ebx, ebx -> “kosongkan register `ebx` dengan xor dengan dirinya sendiri”
int 0x80 -> “aktifkan interrupt `0x80` (mengakhiri program)”

Explain and describe each line of the code.

2. C++ code :

test.cpp | mul.cpp | test

mul.cpp > main()

```cpp
#include <iostream>
using namespace std;

int main()
{
    int num1, num2, mul;
    num1 = 10;
    num2 = 20;
    mul = num1 * num2;

    cout << "Hasil perkalian dari " << num1 << " dan " << num2 << " adalah " << mul << endl;
    return 0;
}
```

3. Makefile :

GNU nano 7.2 mul.sh

```
#!/bin/bash

# Compiles the code
compile() {
    g++ -o mul mul.cpp
}

# Disassembles the compiled code
disassemble() {
    objdump -d mul > mul.asm
}

# Removes all output files created by compile and disassemble functions
clean() {
    rm -f mul mul.asm
}

# Runs the compiled executable
run() {
    ./mul
}
```

```
# Main function to process command-line arguments
main() {
    case "$1" in
        all)
            compile
            ;;
        dump)
            disassemble
            ;;
        clean)
            clean
            ;;
        run)
            run
            ;;
        *)
            echo "Usage: $0 {all|dump|clean|run}"
            exit 1
            ;;
    esac
}

# Call the main function with all arguments passed to the script
main "$@"
```

```
^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify
```

Hasil exec di terminal :

```
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ chmod +x mul.sh
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ ./mul.sh all
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ ./mul.sh dump
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ ls
batch.sh  mul  mul.asm  mul.cpp  mul.sh  test  test.cpp
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ cat mul.asm

mul:     file format elf64-x86-64


Disassembly of section .init:

0000000000001000 <_init>:
    1000:       f3 0f 1e fa             endbr64
    1004:       48 83 ec 08             sub    $0x8,%rsp
    1008:       48 8b 05 e1 2f 00 00    mov    0x2fe1(%rip),%rax        # 3ff0 <__gmon_start__@Base>
    100f:       48 85 c0                test   %rax,%rax
    1012:       74 02                   je     1016 <_init+0x16>
    1014:       ff d0                   call   *%rax
    1016:       48 83 c4 08             add    $0x8,%rsp
    101a:       c3                      ret

Disassembly of section .plt:

0000000000001020 <.plt>:
    1020:       ff 35 82 2f 00 00       push   0x2f82(%rip)        # 3fa8 <_GLOBAL_OFFSET_TABLE_+0x8>
    1026:       ff 25 84 2f 00 00       jmp    *0x2f84(%rip)        # 3fb0 <_GLOBAL_OFFSET_TABLE_+0x10>
    102c:       0f 1f 40 00             nopl   0x0(%rax)
    1030:       f3 0f 1e fa             endbr64
    1034:       68 00 00 00 00          push   $0x0
    1039:       e9 e2 ff ff ff          jmp    1020 <_init+0x20>
```

```
    103e:       66 90                   xchg   %ax,%ax
    1040:       f3 0f 1e fa             endbr64
    1044:       68 01 00 00 00          push   $0x1
    1049:       e9 d2 ff ff ff          jmp    1020 <_init+0x20>
    104e:       66 90                   xchg   %ax,%ax
    1050:       f3 0f 1e fa             endbr64
    1054:       68 02 00 00 00          push   $0x2
    1059:       e9 c2 ff ff ff          jmp    1020 <_init+0x20>
    105e:       66 90                   xchg   %ax,%ax

Disassembly of section .plt.got:

0000000000001060 <__cxa_finalize@plt>:
    1060:       f3 0f 1e fa             endbr64
    1064:       ff 25 66 2f 00 00       jmp    *0x2f66(%rip)        # 3fd0 <__cxa_finalize@GLIBC_2.2.5>
    106a:       66 0f 1f 44 00 00       nopw   0x0(%rax,%rax,1)

Disassembly of section .plt.sec:

0000000000001070 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@plt>:
    1070:       f3 0f 1e fa             endbr64
    1074:       ff 25 3e 2f 00 00       jmp    *0x2f3e(%rip)        # 3fb8 <_ZStlsISt11char_traitsIcEERSt13ba
PKc@GLIBCXX_3.4>
    107a:       66 0f 1f 44 00 00       nopw   0x0(%rax,%rax,1)

0000000000001080 <_ZNSolsEPFRSoS_E@plt>:
    1080:       f3 0f 1e fa             endbr64
    1084:       ff 25 36 2f 00 00       jmp    *0x2f36(%rip)        # 3fc0 <_ZNSolsEPFRSoS_E@GLIBCXX_3.4>
    108a:       66 0f 1f 44 00 00       nopw   0x0(%rax,%rax,1)
```

```
00000000000010d0 <deregister_tm_clones>:
    10d0:       48 8d 3d 39 2f 00 00    lea    0x2f39(%rip),%rdi        # 4010 <__TMC_END__>
    10d7:       48 8d 05 32 2f 00 00    lea    0x2f32(%rip),%rax        # 4010 <__TMC_END__>
    10de:       48 39 f8                cmp    %rdi,%rax
    10e1:       74 15                   je     10f8 <deregister_tm_clones+0x28>
    10e3:       48 8b 05 fe 2e 00 00    mov    0x2efe(%rip),%rax        # 3fe8 <_ITM_deregisterTMCloneTable@B
    10ea:       48 85 c0                test   %rax,%rax
    10ed:       74 09                   je     10f8 <deregister_tm_clones+0x28>
    10ef:       ff e0                   jmp    *%rax
    10f1:       0f 1f 80 00 00 00 00    nopl   0x0(%rax)
    10f8:       c3                      ret
    10f9:       0f 1f 80 00 00 00 00    nopl   0x0(%rax)

0000000000001100 <register_tm_clones>:
    1100:       48 8d 3d 09 2f 00 00    lea    0x2f09(%rip),%rdi        # 4010 <__TMC_END__>
    1107:       48 8d 35 02 2f 00 00    lea    0x2f02(%rip),%rsi        # 4010 <__TMC_END__>
    110e:       48 29 fe                sub    %rdi,%rsi
    1111:       48 89 f0                mov    %rsi,%rax
    1114:       48 c1 ee 3f             shr    $0x3f,%rsi
    1118:       48 c1 f8 03             sar    $0x3,%rax
    111c:       48 01 c6                add    %rax,%rsi
    111f:       48 d1 fe                sar    $1,%rsi
    1122:       74 14                   je     1138 <register_tm_clones+0x38>
    1124:       48 8b 05 cd 2e 00 00    mov    0x2ecd(%rip),%rax        # 3ff8 <_ITM_registerTMCloneTable@Bas
    112b:       48 85 c0                test   %rax,%rax
    112e:       74 08                   je     1138 <register_tm_clones+0x38>
    1130:       ff e0                   jmp    *%rax
    1132:       66 0f 1f 44 00 00       nopw   0x0(%rax,%rax,1)
    1138:       c3                      ret
    1139:       0f 1f 80 00 00 00 00    nopl   0x0(%rax)
```

```
0000000000001189 <main>:
    1189:       f3 0f 1e fa             endbr64
    118d:       55                      push   %rbp
    118e:       48 89 e5                mov    %rsp,%rbp
    1191:       48 83 ec 10             sub    $0x10,%rsp
    1195:       c7 45 f4 0a 00 00 00    movl   $0xa,-0xc(%rbp)
    119c:       c7 45 f8 14 00 00 00    movl   $0x14,-0x8(%rbp)
    11a3:       8b 45 f4                mov    -0xc(%rbp),%eax
    11a6:       0f af 45 f8             imul   -0x8(%rbp),%eax
    11aa:       89 45 fc                mov    %eax,-0x4(%rbp)
    11ad:       48 8d 05 50 0e 00 00    lea    0xe50(%rip),%rax        # 2004 <_IO_stdin_used+0x4>
    11b4:       48 89 c6                mov    %rax,%rsi
    11b7:       48 8d 05 82 2e 00 00    lea    0x2e82(%rip),%rax        # 4040 <_ZSt4cout@GLIBCXX_3.4>
    11be:       48 89 c7                mov    %rax,%rdi
    11c1:       e8 aa fe ff ff          call   1070 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@
    11c6:       48 89 c2                mov    %rax,%rdx
    11c9:       8b 45 f4                mov    -0xc(%rbp),%eax
    11cc:       89 c6                   mov    %eax,%esi
    11ce:       48 89 d7                mov    %rdx,%rdi
    11d1:       e8 ba fe ff ff          call   1090 <_ZNSolsEi@plt>
    11d6:       48 89 c2                mov    %rax,%rdx
    11d9:       48 8d 05 3a 0e 00 00    lea    0xe3a(%rip),%rax        # 201a <_IO_stdin_used+0x1a>
    11e0:       48 89 c6                mov    %rax,%rsi
    11e3:       48 89 d7                mov    %rdx,%rdi
    11e6:       e8 85 fe ff ff          call   1070 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@
    11eb:       48 89 c2                mov    %rax,%rdx
    11ee:       8b 45 f8                mov    -0x8(%rbp),%eax
    11f1:       89 c6                   mov    %eax,%esi
    11f3:       48 89 d7                mov    %rdx,%rdi
    11f6:       e8 95 fe ff ff          call   1090 <_ZNSolsEi@plt>
```

```
    11fb:       48 89 c2                mov    %rax,%rdx
    11fe:       48 8d 05 1b 0e 00 00    lea    0xe1b(%rip),%rax        # 2020 <_IO_stdin_used+0x20>
    1205:       48 89 c6                mov    %rax,%rsi
    1208:       48 89 d7                mov    %rdx,%rdi
    120b:       e8 60 fe ff ff          call   1070 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@
    1210:       48 89 c2                mov    %rax,%rdx
    1213:       8b 45 fc                mov    -0x4(%rbp),%eax
    1216:       89 c6                   mov    %eax,%esi
    1218:       48 89 d7                mov    %rdx,%rdi
    121b:       e8 70 fe ff ff          call   1090 <_ZNSolsEi@plt>
    1220:       48 8b 15 b1 2d 00 00    mov    0x2db1(%rip),%rdx        # 3fd8 <_ZSt4endlIcSt11char_traitsIc
IT_T0_ES6_@GLIBCXX_3.4>
    1227:       48 89 d6                mov    %rdx,%rsi
    122a:       48 89 c7                mov    %rax,%rdi
    122d:       e8 4e fe ff ff          call   1080 <_ZNSolsEPFRSoS_E@plt>
    1232:       b8 00 00 00 00          mov    $0x0,%eax
    1237:       c9                      leave
    1238:       c3                      ret

Disassembly of section .fini:

000000000000123c <_fini>:
    123c:       f3 0f 1e fa             endbr64
    1240:       48 83 ec 08             sub    $0x8,%rsp
    1244:       48 83 c4 08             add    $0x8,%rsp
    1248:       c3                      ret
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$ ./mul.sh run
Hasil perkalian dari 10 dan 20 adalah 200
reyy@reyy-Aspire-A514-51KG:~/prak-skj/assignment1$
```