

**LAPORAN PRAKTIKUM
SISTEM OPERASI
MODUL 9**



DISUSUN OLEH:

NIM	L200220277
NAMA	MHD. FARHAN LUBIS
KELAS	F

**PROGRAM STUDI INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA**

2023

DAFTAR ISI

DAFTAR ISI	2
LANGKAH KERJA	4
Membuat sebuah file dan menuliskan data di dalamnya.....	4
1. Deklarasi variabel untuk buffer ‘buf’ untuk menyimpan 100 byte data.....	5
2. Gunakan nama file yang diberikan sebagai argumen pada command-line	5
3. Buat sebuah file baru dengan nama file seperti pada (2) menggunakan fungsi system-call open() dengan opsi O_CREAT dan O_TRUNC.....	5
4. Periksa ‘file-descriptor’, jika pembuatan file tidak berhasil, selanjutnya berhenti (stop) dan keluar dari program.....	5
5. Baca input dari console sampai use menekan tombol Ctrl+D. Membaca 100 byte dari data yang dimasukan mellaui console dan menyimpannya ke dalam variabel ‘buf’ menggunakan perintah system-call read(). Memindahkan isi variabel ‘buf’ ke dalam file menggunakan perintah ‘write’.	5
6. Menutup file dengan menggunakan fungsi ‘close’	5
7. Stop	5
Kode program:.....	5
Terminal ketika program dicompile & dijalankan:	6
Membaca sebuah file dan menampilkan isinya di layar.....	6
1. Deklarasi sebuah variabel buffer tipe character untuk menyimpan 100 byte data..	7
2. Gunakan nama-file sesuai dengan argumen yang diberikan dalam perintah command-line.....	7
3. Buka file untuk dibaca menggunakan perintah ‘open’ dengan opsi O_RDONLY. ..	7
4. Periksa isi file-descriptor, Jika file tidak ada maka program berhenti, stop	7
5. Baca isi file per 100 byte data menggunakan perintah ‘read’ sampai ketemu dengan tanda akhir file, ‘END-OFFILE’.	7
6. Tutup file menggunakan perintah ‘close’	7
7. Stop	7
Kode program:.....	7
Terminal ketika program dicompile & dijalankan:	8
Menambah isi file.	8
a. Deklarasi sebuah variabel buffer tipe character untuk menyimpan 100 byte data..	9

b. Gunakan nama-file sesuai dengan argumen yang diberikan dalam perintah command-line.....	9
c. Buka file di atas dengan menggunakan perintah open dengan opsi O_APPEND.	9
d. Periksa file-descriptor, jika nilainya berupa angka negatif, stop program	9
e. Baca input user dari console sampai user menekan tombol Ctrl + D. Jika user sudah menekan tombol Ctrl + D, baca 100 byte dari console dan simpan ke dalam variabel 'buf' menggunakan perintah read. Selanjutnya tuliskan isi variabel 'buf' ke dalam file menggunakan perintah 'write'	9
f. Tutup file dengan menggunakan fungsi 'close'.	9
Kode program:	9
Terminal ketika program dicompile & dijalankan:	10

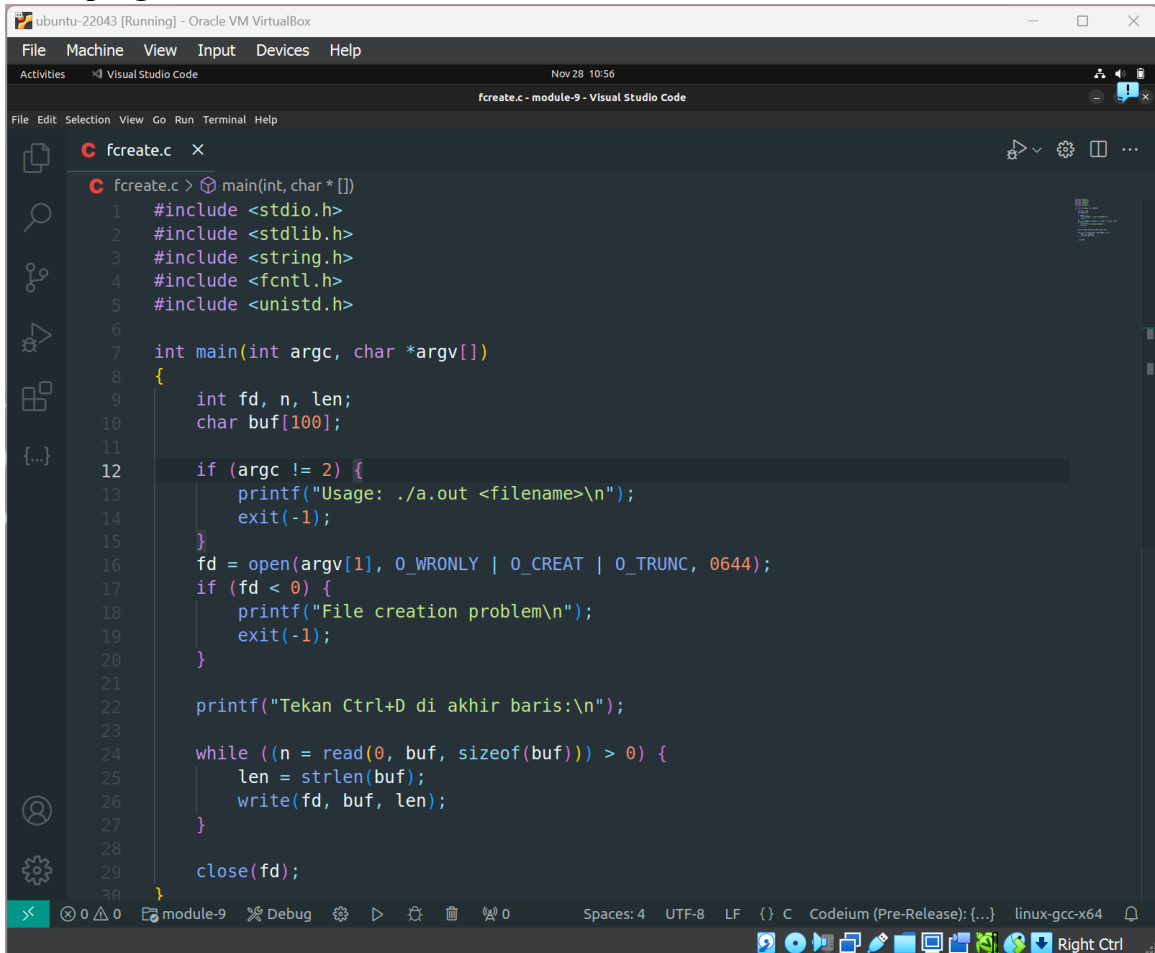
LANGKAH KERJA

Membuat sebuah file dan menuliskan data di dalamnya.

Membuat program dengan algoritma sebagai berikut (Contoh program diberikan pada bagian berikutnya):

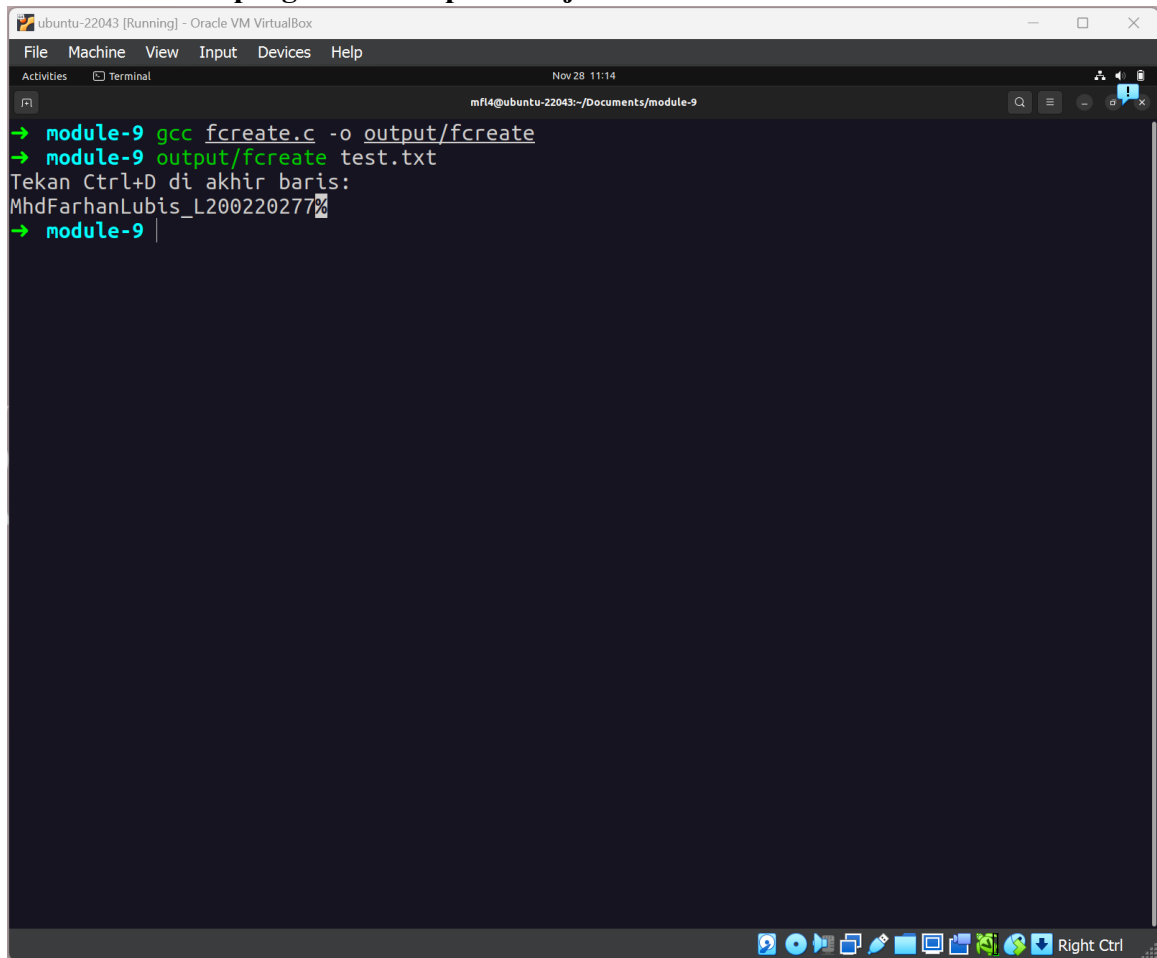
1. Deklarasi variabel untuk buffer 'buf' untuk menyimpan 100 byte data.
2. Gunakan nama file yang diberikan sebagai argumen pada command-line
3. Buat sebuah file baru dengan nama file seperti pada (2) menggunakan fungsi system-call open() dengan opsi O_CREAT dan O_TRUNC.
4. Periksa 'file-descriptor', jika pembuatan file tidak berhasil, selanjutnya berhenti (stop) dan keluar dari program.
5. Baca input dari console sampai user menekan tombol Ctrl+D. Membaca 100 byte dari data yang dimasukkan melalui console dan menyimpannya ke dalam variabel 'buf' menggunakan perintah system-call read(). Memindahkan isi variabel 'buf' ke dalam file menggunakan perintah 'write'.
6. Menutup file dengan menggunakan fungsi 'close'
7. Stop

Kode program:



```
fcreate.c > main(int, char *[])
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6
7  int main(int argc, char *argv[])
8  {
9      int fd, n, len;
10     char buf[100];
11
12     if (argc != 2) {
13         printf("Usage: ./a.out <filename>\n");
14         exit(-1);
15     }
16     fd = open(argv[1], O_WRONLY | O_CREAT | O_TRUNC, 0644);
17     if (fd < 0) {
18         printf("File creation problem\n");
19         exit(-1);
20     }
21
22     printf("Tekan Ctrl+D di akhir baris:\n");
23
24     while ((n = read(0, buf, sizeof(buf))) > 0) {
25         len = strlen(buf);
26         write(fd, buf, len);
27     }
28
29     close(fd);
30 }
```

Terminal ketika program dicompile & dijalankan:



```
ubuntu-22043 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Nov 28 11:14
mfl4@ubuntu-22043:~/Documents/module-9
→ module-9 gcc fcreate.c -o output/fcreate
→ module-9 output/fcreate test.txt
Tekan Ctrl+D di akhir baris:
MhdFarhanLubis_L200220277%
→ module-9 |
```

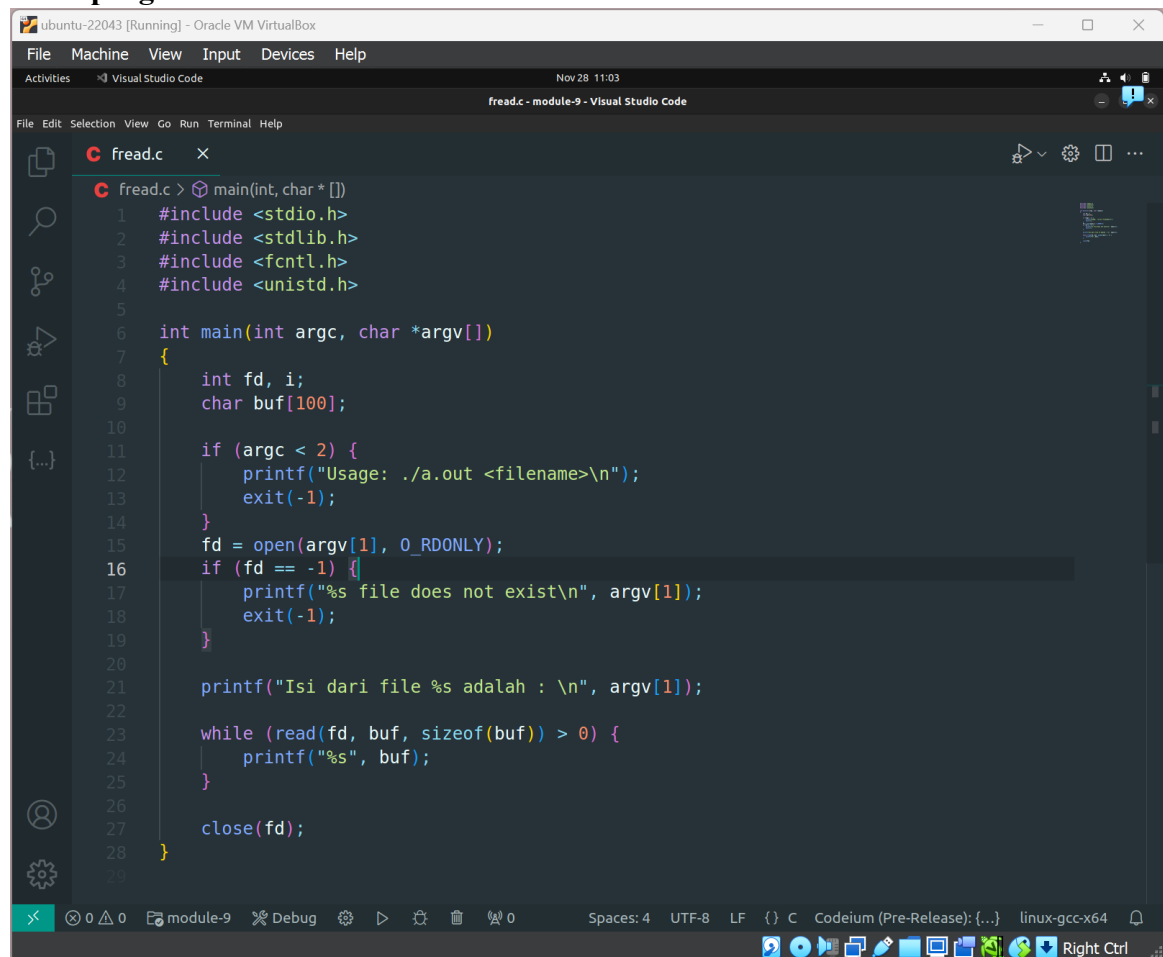
Compilasi program dengan perintah `$gcc -o fcreate fcreate.c` , selanjutnya jika proses kompilasi berhasil jalankan program dengan memanggil nama file `$./fcreate test.txt`, ketikkan text yang akan disimpan dalam file, tekan `Ctrl + D` untuk menandai baris baru, dan tekan `Ctrl + C` jika sudah selesai.

Membaca sebuah file dan menampilkan isinya di layar.

Membuat kode program dengan algorithm sebagai berikut:

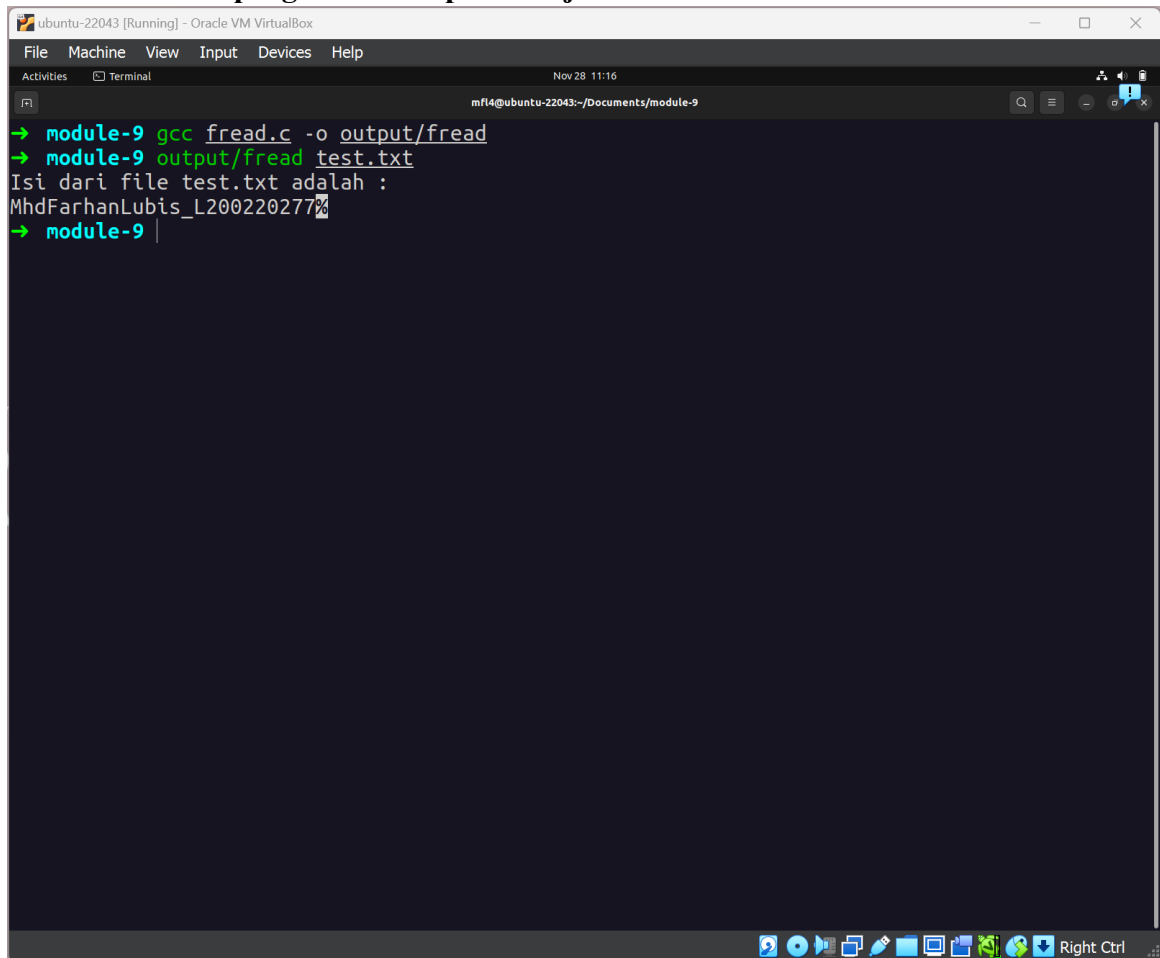
1. Deklarasi sebuah variabel buffer tipe character untuk menyimpan 100 byte data.
2. Gunakan nama-file sesuai dengan argumen yang diberikan dalam perintah command-line.
3. Buka file untuk dibaca menggunakan perintah 'open' dengan opsi O_RDONLY.
4. Periksa isi file-descriptor, Jika file tidak ada maka program berhenti, stop
5. Baca isi file per 100 byte data menggunakan perintah 'read' sampai ketemu dengan tanda akhir file, 'END-OFFILE'.
6. Tutup file menggunakan perintah 'close'
7. Stop

Kode program:



```
fread.c > main(int, char * [])
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <fcntl.h>
4  #include <unistd.h>
5
6  int main(int argc, char *argv[])
7  {
8      int fd, i;
9      char buf[100];
10
11     if (argc < 2) {
12         printf("Usage: ./a.out <filename>\n");
13         exit(-1);
14     }
15     fd = open(argv[1], O_RDONLY);
16     if (fd == -1) {
17         printf("%s file does not exist\n", argv[1]);
18         exit(-1);
19     }
20
21     printf("Isi dari file %s adalah : \n", argv[1]);
22
23     while (read(fd, buf, sizeof(buf)) > 0) {
24         printf("%s", buf);
25     }
26
27     close(fd);
28 }
```

Terminal ketika program dicompile & dijalankan:



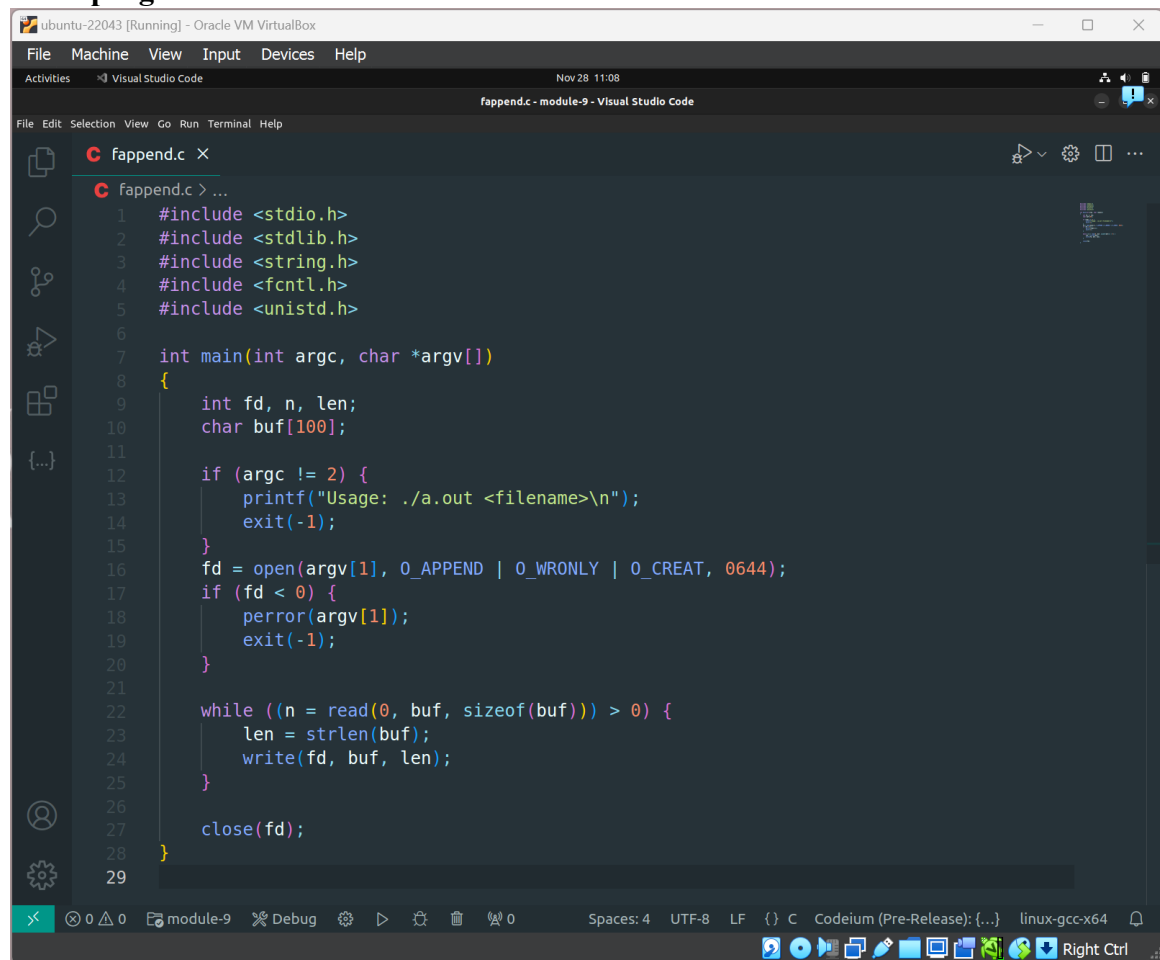
```
ubuntu-22043 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Nov 28 11:16
mfl4@ubuntu-22043:~/Documents/module-9
→ module-9 gcc fread.c -o output/fread
→ module-9 output/fread test.txt
Isi dari file test.txt adalah :
MhdFarhanLubis_L200220277%
→ module-9 |
```

Menambah isi file.

Membuat kode program dengan algorithm sebagai barikut:

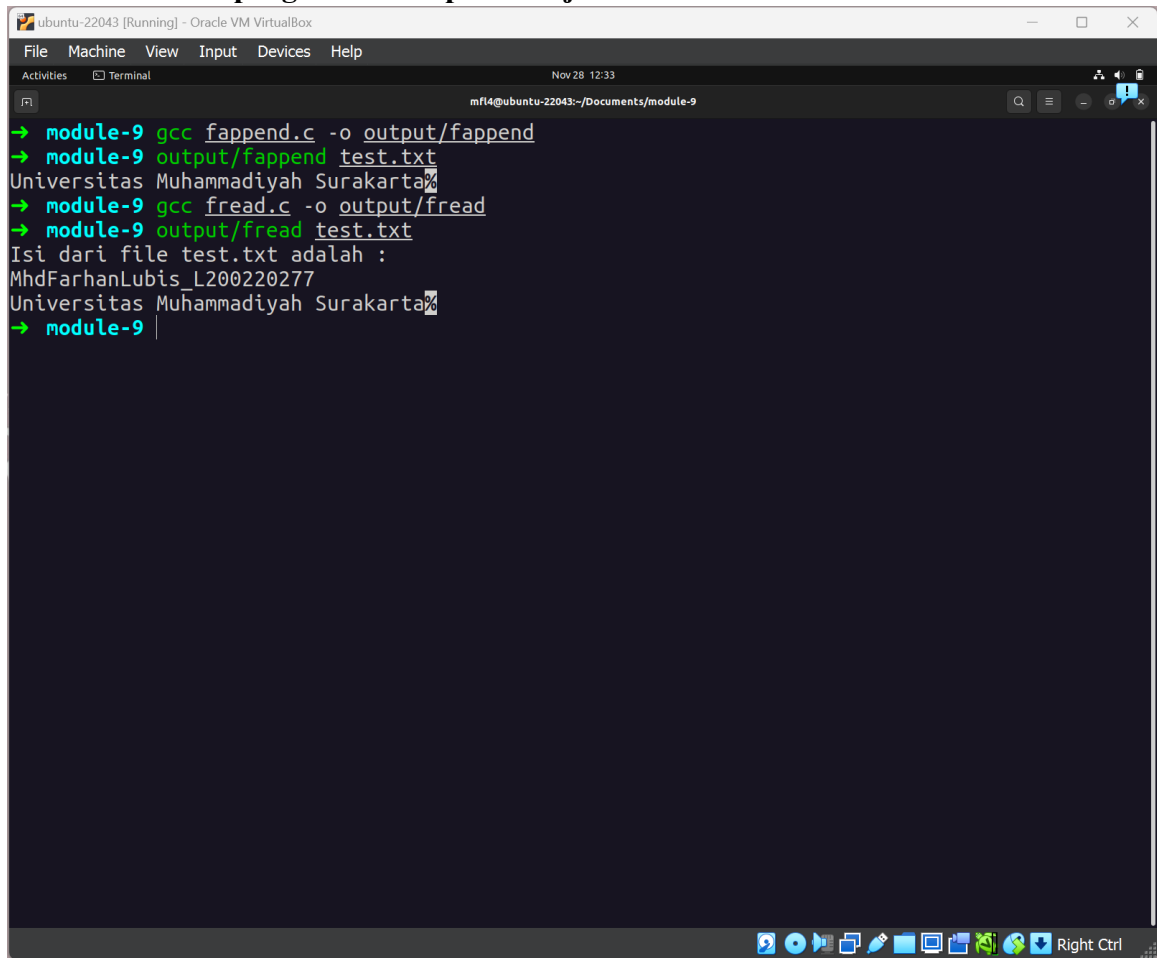
- a. Deklarasi sebuah variabel buffer tipe character untuk menyimpan 100 byte data.
- b. Gunakan nama-file sesuai dengan argumen yang diberikan dalam perintah command-line.
- c. Buka file di atas dengan menggunakan perintah open dengan opsi O_APPEND.
- d. Periksa file-descriptor, jika nilainya berupa angka negatif, stop program
- e. Baca input user dari console sampai user menekan tombol Ctrl + D. Jika user sudah menekan tombol Ctrl + D, baca 100 byte dari console dan simpan ke dalam variabel 'buf' menggunakan perintah read. Selanjutnya tuliskan isi variabel 'buf' ke dalam file menggunakan perintah 'write'.
- f. Tutup file dengan menggunakan fungsi 'close'.

Kode program:



```
fappend.c > ...
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6
7  int main(int argc, char *argv[])
8  {
9      int fd, n, len;
10     char buf[100];
11
12     if (argc != 2) {
13         printf("Usage: ./a.out <filename>\n");
14         exit(-1);
15     }
16     fd = open(argv[1], O_APPEND | O_WRONLY | O_CREAT, 0644);
17     if (fd < 0) {
18         perror(argv[1]);
19         exit(-1);
20     }
21
22     while ((n = read(0, buf, sizeof(buf))) > 0) {
23         len = strlen(buf);
24         write(fd, buf, len);
25     }
26
27     close(fd);
28 }
29
```

Terminal ketika program dicompile & dijalankan:



The screenshot shows a terminal window titled 'ubuntu-22043 [Running] - Oracle VM VirtualBox'. The terminal output is as follows:

```
→ module-9 gcc fappend.c -o output/fappend
→ module-9 output/fappend test.txt
Universitas Muhammadiyah Surakarta%
→ module-9 gcc fread.c -o output/fread
→ module-9 output/fread test.txt
Isi dari file test.txt adalah :
MhdFarhanLubis_L200220277
Universitas Muhammadiyah Surakarta%
→ module-9 |
```

The terminal window includes a menu bar (File, Machine, View, Input, Devices, Help), a title bar, and a status bar at the bottom with system icons and a 'Right Ctrl' label.