

**LAPORAN PRAKTIKUM
SISTEM OPERASI
MODUL 10**



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
Program untuk mensimulasi perintah 'ls'	4
1. Menyimpan 'path' dari direktori kerja saat ini menggunakan perintah system call 'getcwd'	4
2. Membaca isi direktori dari path di atas menggunakan perintah system call 'scandir' dan mengurutkan hasil pembacaannya dan menyimpannya dalam sebuah variabel array.....	4
3. Menampilkan nama direktori (dname) dan nama file didalamnya jika file atau direktori tersebut tidak memiliki properti 'HIDE'.	4
4. Stop.....	4
Kode program:	4
Terminal ketika program dicompile & dijalankan:.....	5
Program untuk mensimulasi perintah 'grep'.....	5
1. Gunakan nama file yang diberikan dalam argumen command-line.....	6
2. Buka file dalam mode 'read-only' menggunakan perintah system call 'open'	6
3. Jika file tidak ada, keluar program, stop	6
4. Misal panjang string yang dicari adalah n.....	6
5. Baca file perbaris sampai akhir file (END-OF-FILE), untuk setiap baris lakukan hal-hal berikut: (a) Periksa untuk mencari string dalam baris tersebut dengan dalam range 1-n, 2- n+1, dan seterusnya, (b) Jika string ditemukan tampilan baris tersebut di layar.	6
6. Tutup file menggunakan perintah 'close'.....	6
7. Stop.....	6
Kode program:	6
Terminal ketika program dicompile & dijalankan:.....	8
Program untuk mensimulasi perintah 'cp'.	8
1. Gunakan nama file untuk sumber dan tujuan dari argumen yang diberikan dalam command line.	9
2. Deklarasi sebuah buffer berukuran 1 KB.....	9
3. Buka file sumber dalam mode 'read-only' menggunakan fungsi 'open'	9
4. Jika file sumber tidak ditemukan, stop keluar dari program	9
5. Membuat file baru sebagai file target dengan menggunakan perintah 'creat'.	9

6. Jika proses pembuatan file gagal, stop keluar dari program.	9
7. Proses penyalinan (copy) file dilakukan dengan cara berikut: (a) Membaca 1KB data dari file sumber dan menyimpan hasilnya dalam buffer menggunakan perintah 'read'. (b) Menuliskan isi buffer dalam file target menggunakan perintah 'write'. (c) Jika bertemudengan kode 'END-OF-FILE' lanjut ke nomor 8, yang lain kembali ke perintah (a). 9	9
8. Tutup file sumber dan target menggunakna perintah 'close'.	9
9. Stop.....	9
Kode program:	9
Terminal ketika program dicompile & dijalankan:.....	11
Program untuk mensimulasi perintah 'rm'	11
1. Gunakan nama file yang diberikan dalam argumen command line	12
2. Buka file dalam mode 'read-only' menggunakan perintah 'read'	12
3. Jika file tidak ditemukan, stop keluar program	12
4. Tutup file menggunakan perintah 'close'	12
5. Menghapus file menggunakan perintah 'unlink'	12
6. Stop.....	12
Kode program:	12
Terminal ketika program dicompile & dijalankan:.....	13

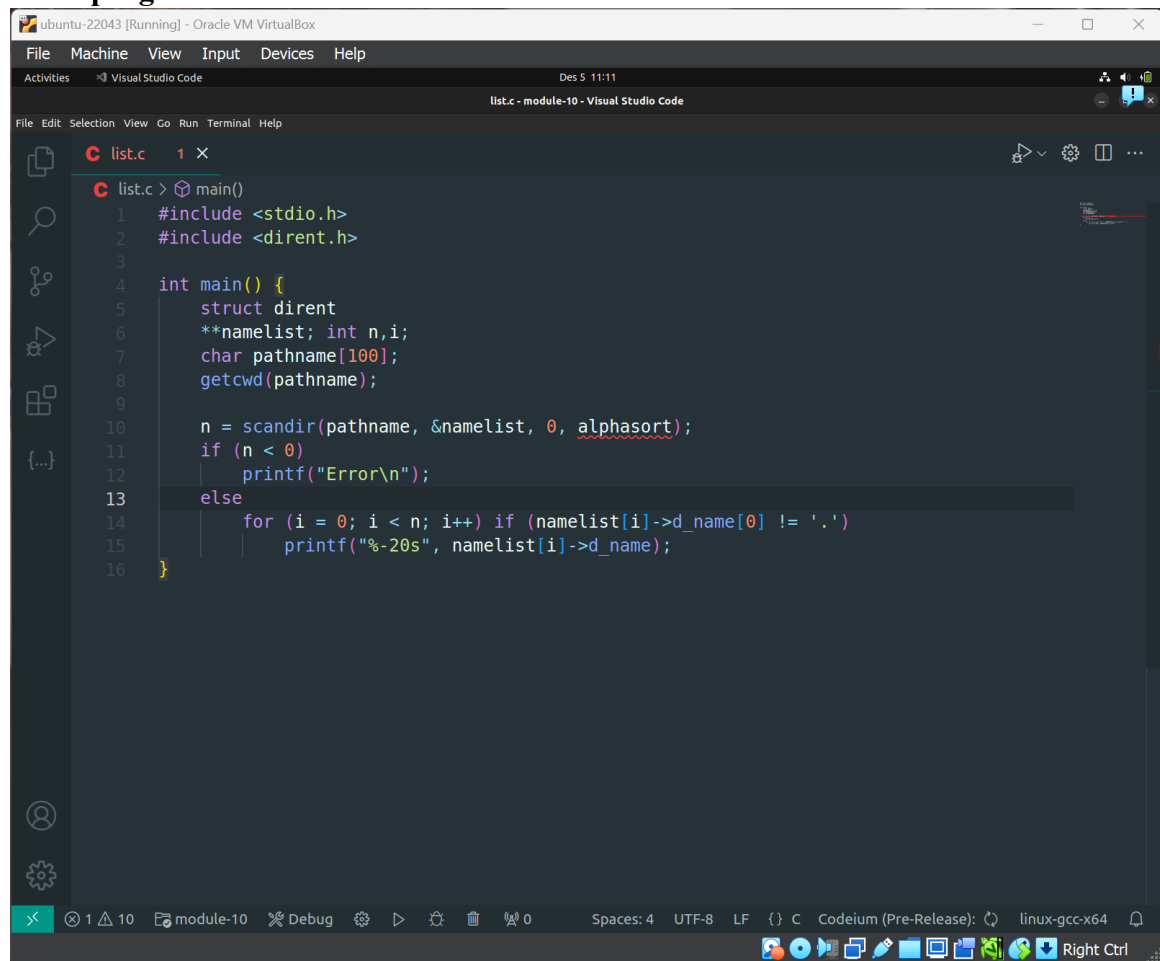
LANGKAH KERJA

Program untuk mensimulasi perintah 'ls'.

Membuat kode program dengan algorithm sebagai berikut:

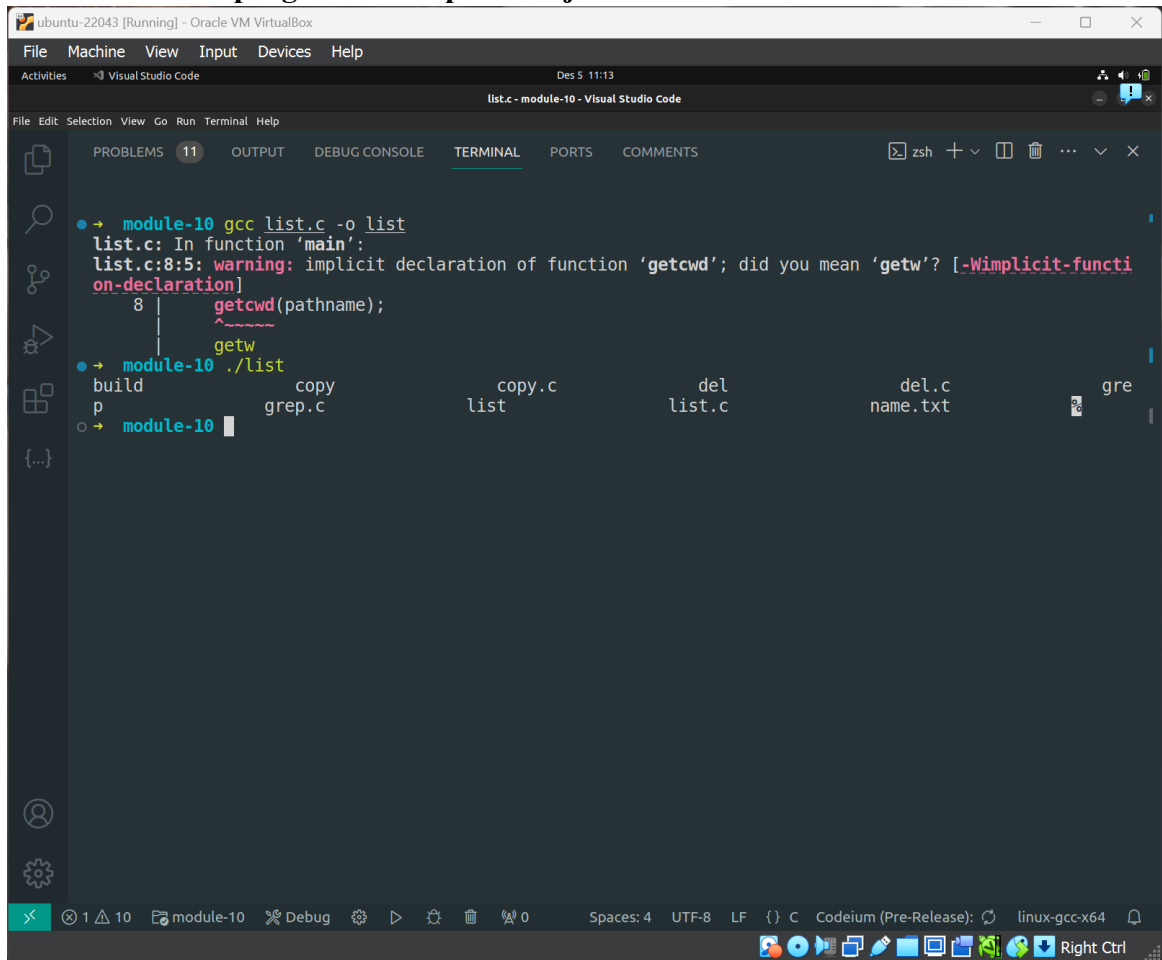
1. Menyimpan 'path' dari direktori kerja saat ini menggunakan perintah system call 'getcwd'
2. Membaca isi direktori dari path di atas menggunakan perintah system call 'scandir' dan mengurutkan hasil pembacaannya dan menyimpannya dalam sebuah variabel array.
3. Menampilkan nama direktori (dname) dan nama file didalamnya jika file atau direktori tersebut tidak memiliki properti 'HIDE'.
4. Stop

Kode program:



```
list.c 1 X
list.c > main()
1  #include <stdio.h>
2  #include <dirent.h>
3
4  int main() {
5      struct dirent
6      **namelist; int n,i;
7      char pathname[100];
8      getcwd(pathname);
9
10     n = scandir(pathname, &namelist, 0, alphasort);
11     if (n < 0)
12         printf("Error\n");
13     else
14         for (i = 0; i < n; i++) if (namelist[i]->d_name[0] != '.')
15             printf("%-20s", namelist[i]->d_name);
16 }
```

Terminal ketika program dicompile & dijalankan:



The screenshot shows a terminal window within the Visual Studio Code editor. The terminal is running a Zsh shell. The first command executed is `gcc list.c -o list`, which compiles the file `list.c` into an executable named `list`. A warning is displayed: `list.c: In function 'main': list.c:8:5: warning: implicit declaration of function 'getcwd'; did you mean 'getw'? [-Wimplicit-function-declaration]`. The second command is `./list`, which runs the compiled program. The output of the program is a directory listing: `build copy copy.c del del.c gre p grep.c list list.c name.txt`. The Visual Studio Code interface is visible in the background, showing the file explorer on the left and the terminal output in the center.

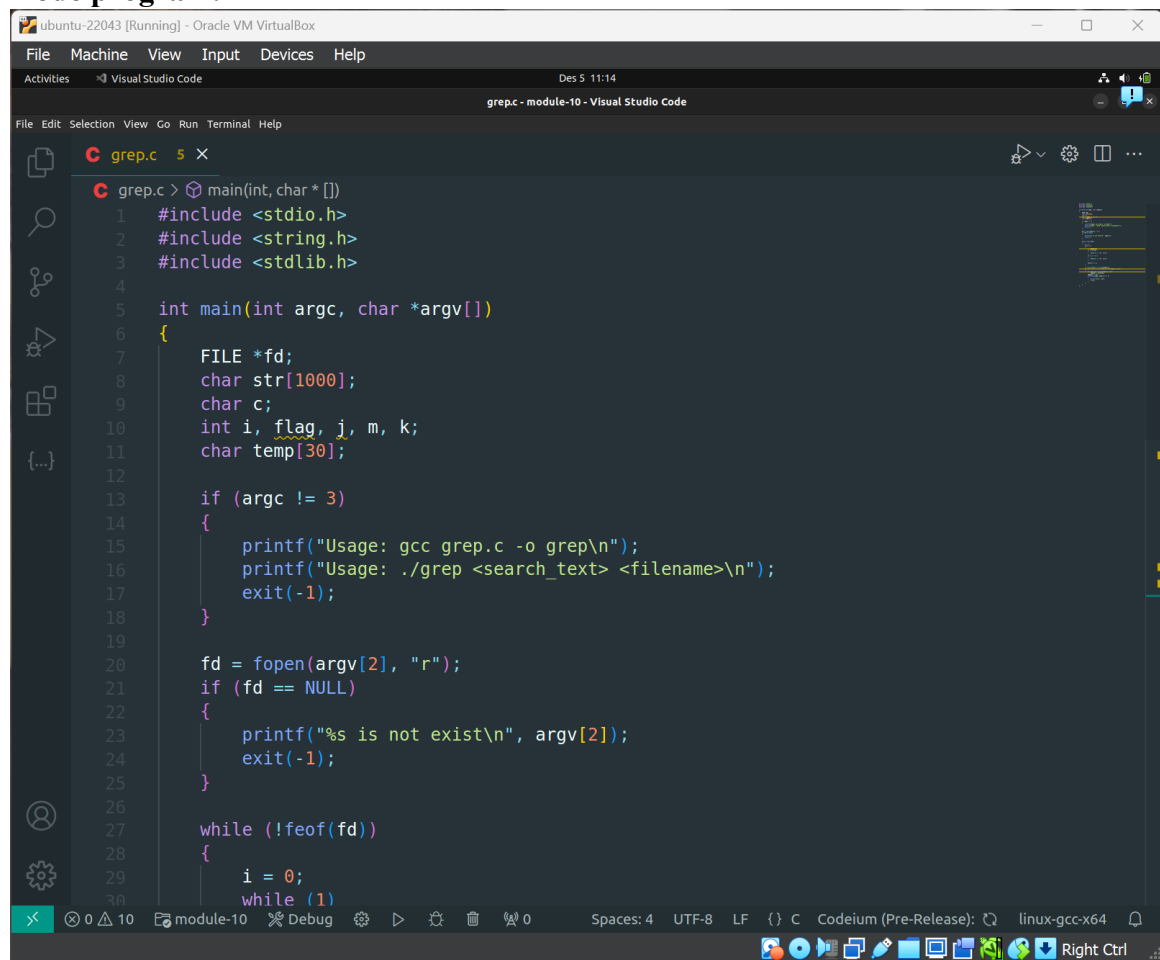
```
module-10 gcc list.c -o list
list.c: In function 'main':
list.c:8:5: warning: implicit declaration of function 'getcwd'; did you mean 'getw'? [-Wimplicit-function-declaration]
      8 |         getcwd(pathname);
         |         ^~~~~~
         |         getw
module-10 ./list
build      copy      copy.c      del      del.c      gre
p          grep.c    list      list.c    name.txt
```

Program untuk mensimulasi perintah 'grep'.

Membuat kode program dengan algorithm sebagai berikut:

1. Gunakan nama file yang diberikan dalam argumen command-line
2. Buka file dalam mode 'read-only' menggunakan perintah system call 'open'
3. Jika file tidak ada, keluar program, stop
4. Misal panjang string yang dicari adalah n
5. Baca file perbaris sampai akhir file (END-OF-FILE), untuk setiap baris lakukan hal-hal berikut: (a) Periksa untuk mencari string dalam baris tersebut dengan dalam range $1-n$, $2-n+1$, dan seterusnya, (b) Jika string ditemukan tampilan baris tersebut di layar.
6. Tutup file menggunakan perintah 'close'.
7. Stop

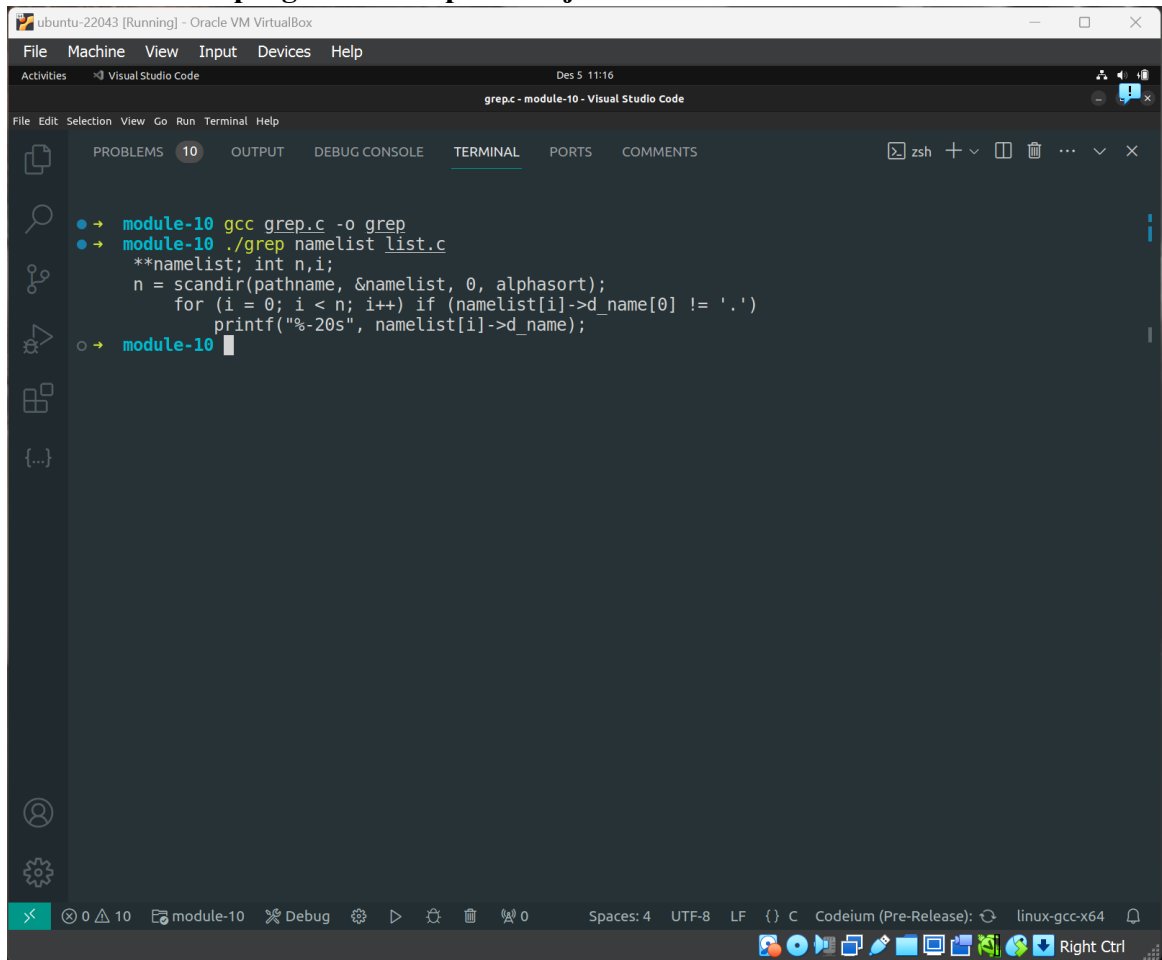
Kode program:



```
grep.c 5 X
C grep.c > main(int, char *[])
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4
5  int main(int argc, char *argv[])
6  {
7      FILE *fd;
8      char str[1000];
9      char c;
10     int i, flag, j, m, k;
11     char temp[30];
12
13     if (argc != 3)
14     {
15         printf("Usage: gcc grep.c -o grep\n");
16         printf("Usage: ./grep <search_text> <filename>\n");
17         exit(-1);
18     }
19
20     fd = fopen(argv[2], "r");
21     if (fd == NULL)
22     {
23         printf("%s is not exist\n", argv[2]);
24         exit(-1);
25     }
26
27     while (!feof(fd))
28     {
29         i = 0;
30         while (1)
```

```
30     while (1)
31     {
32         c = fgetc(fd);
33         if (feof(fd))
34         {
35             str[i++] = '\0'; break;
36         }
37         if (c == '\n')
38         {
39             str[i++] = '\0'; break;
40         }
41
42         str[i++] = c;
43     }
44
45     if (strlen(str) >= strlen(argv[1]))
46     for (k = 0; k <= strlen(str)-strlen(argv[1]); k++)
47     {
48         for (m=0; m < strlen(argv[1]); m++)
49             temp[m] = str[k+m];
50         temp[m] = '\0';
51         if(strcmp(temp, argv[1]) == 0)
52         {
53             printf("%s\n", str);
54             break;
55         }
56     }
57 }
58 }
```

Terminal ketika program dicompile & dijalankan:



The screenshot shows a terminal window within the Visual Studio Code editor. The terminal is titled 'grep.c - module-10 - Visual Studio Code'. The command prompt is 'module-10'. The terminal shows the following commands and output:

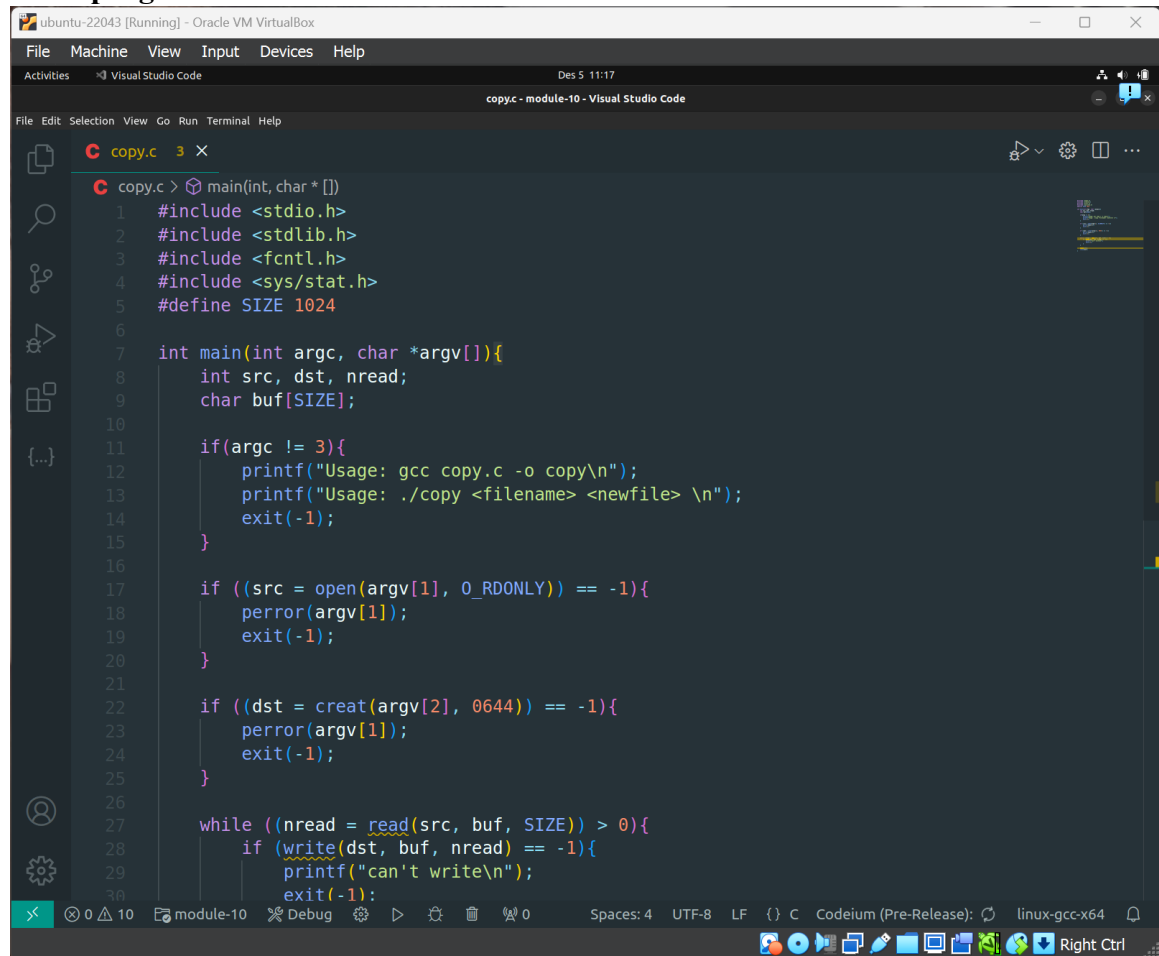
```
module-10 gcc grep.c -o grep
module-10 ./grep namelist list.c
**namelist; int n,i;
n = scandir(pathname, &namelist, 0, alphasort);
for (i = 0; i < n; i++) if (namelist[i]->d_name[0] != '.')
printf("%-20s", namelist[i]->d_name);
module-10
```

Program untuk mensimulasi perintah ‘cp’.

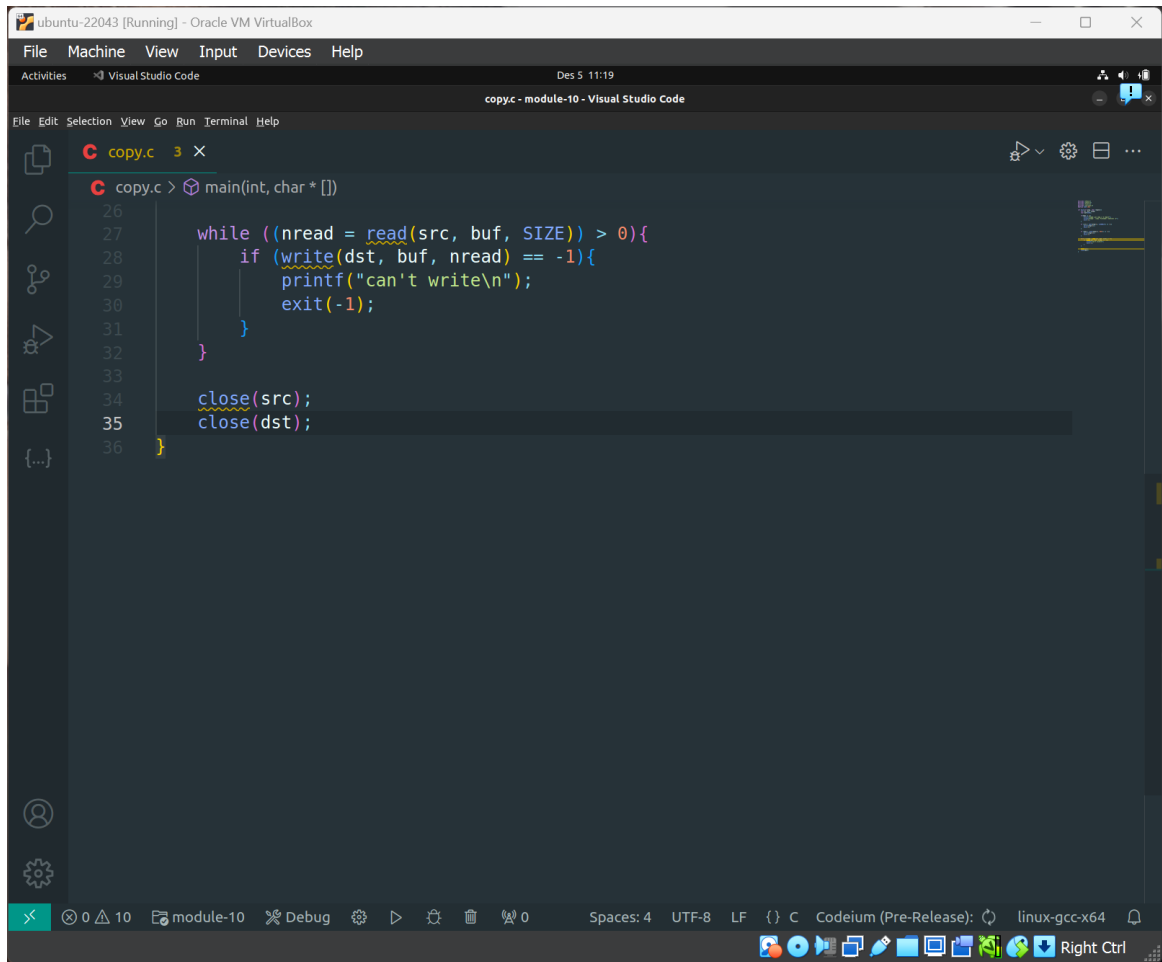
Membuat kode program dengan algorithm sebagai berikut:

1. Gunakan nama file untuk sumber dan tujuan dari argumen yang diberikan dalam command line.
2. Deklarasi sebuah buffer berukuran 1 KB
3. Buka file sumber dalam mode 'read-only' menggunakan fungsi 'open'
4. Jika file sumber tidak ditemukan, stop keluar dari program
5. Membuat file baru sebagai file target dengan menggunakan perintah 'creat'.
6. Jika proses pembuatan file gagal, stop keluar dari program.
7. Proses penyalinan (copy) file dilakukan dengan cara berikut: (a) Membaca 1KB data dari file sumber dan menyimpan hasilnya dalam buffer menggunakan perintah 'read'. (b) Menuliskan isi buffer dalam file target menggunakan perintah 'write'. (c) Jika bertemu dengan kode 'END-OF-FILE' lanjut ke nomor 8, yang lain kembali ke perintah (a)
8. Tutup file sumber dan target menggunakan perintah 'close'.
9. Stop

Kode program:

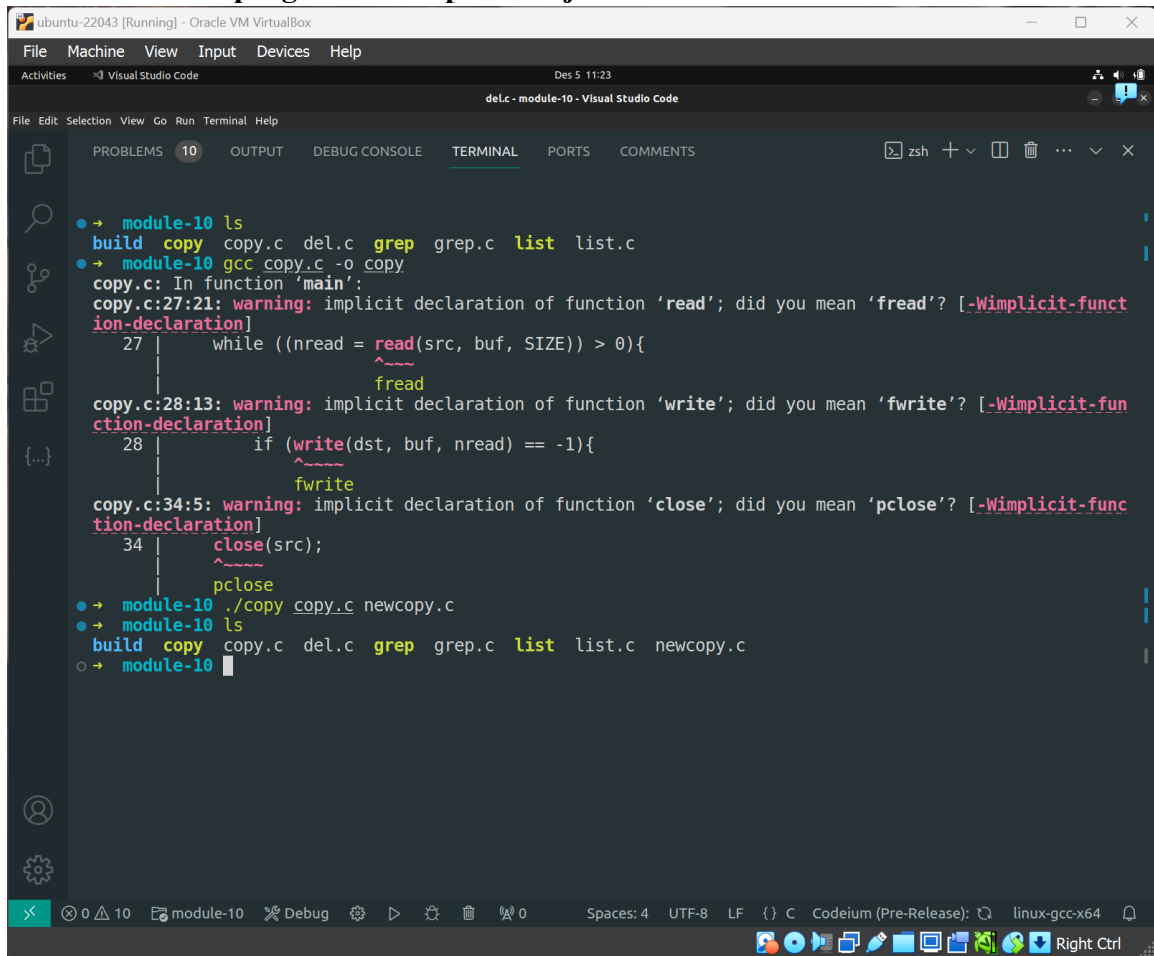


```
copy.c > main(int, char *[])
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <fcntl.h>
4  #include <sys/stat.h>
5  #define SIZE 1024
6
7  int main(int argc, char *argv[]){
8      int src, dst, nread;
9      char buf[SIZE];
10
11     if(argc != 3){
12         printf("Usage: gcc copy.c -o copy\n");
13         printf("Usage: ./copy <filename> <newfile> \n");
14         exit(-1);
15     }
16
17     if ((src = open(argv[1], O_RDONLY)) == -1){
18         perror(argv[1]);
19         exit(-1);
20     }
21
22     if ((dst = creat(argv[2], 0644)) == -1){
23         perror(argv[1]);
24         exit(-1);
25     }
26
27     while ((nread = read(src, buf, SIZE)) > 0){
28         if (write(dst, buf, nread) == -1){
29             printf("can't write\n");
30             exit(-1);
31         }
32     }
```



```
copy.c 3 x
copy.c > main(int, char * [])
26
27     while ((nread = read(src, buf, SIZE)) > 0){
28         if (write(dst, buf, nread) == -1){
29             printf("can't write\n");
30             exit(-1);
31         }
32     }
33     close(src);
34     close(dst);
35 }
36 }
```

Terminal ketika program dicompile & dijalankan:



The screenshot shows a Visual Studio Code window with a terminal open. The terminal displays the following commands and output:

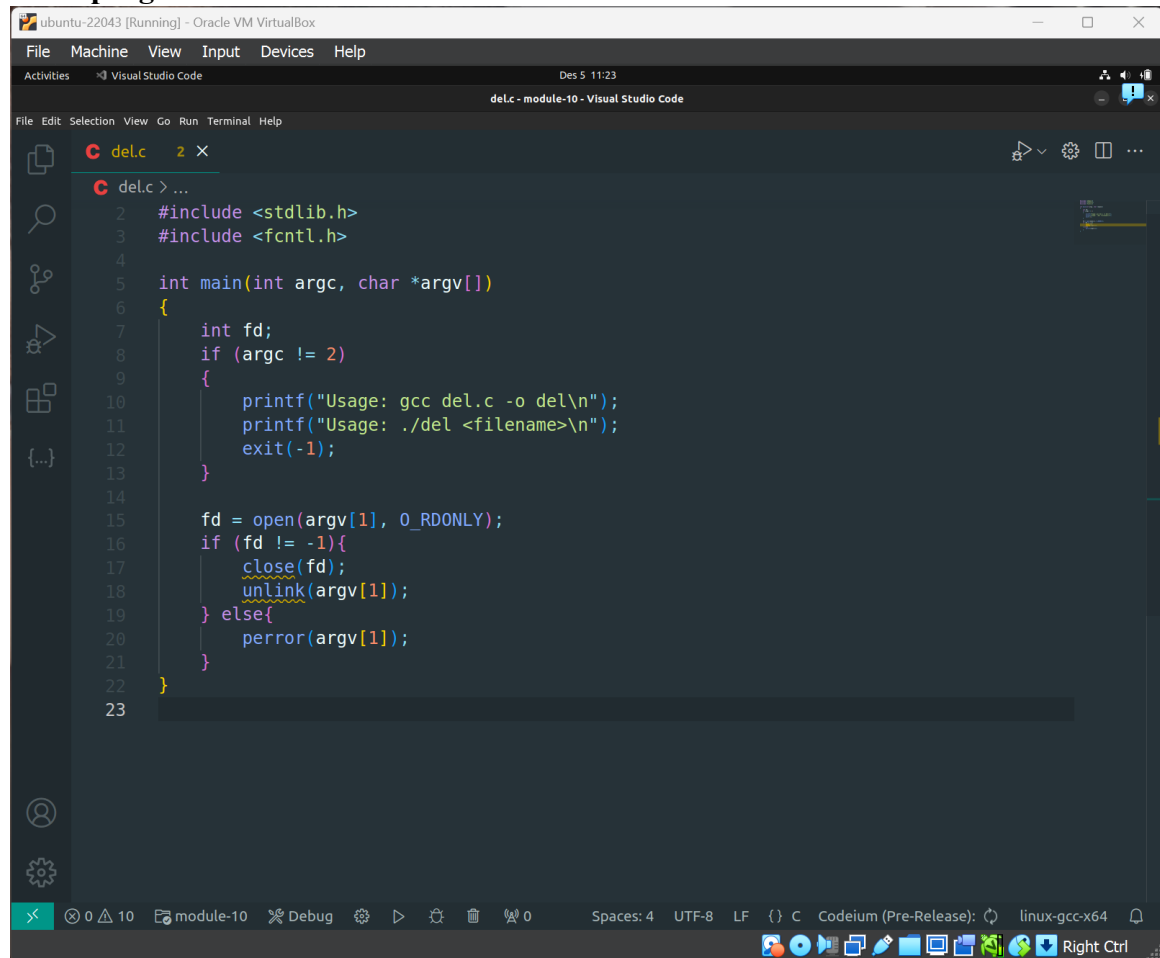
```
module-10 ls
build copy copy.c del.c grep grep.c list list.c
module-10 gcc copy.c -o copy
copy.c: In function 'main':
copy.c:27:21: warning: implicit declaration of function 'read'; did you mean 'fread'? [-Wimplicit-function-declaration]
    27 |         while ((nread = read(src, buf, SIZE)) > 0){
        |                       ^~~~~
copy.c:28:13: warning: implicit declaration of function 'write'; did you mean 'fwrite'? [-Wimplicit-function-declaration]
    28 |         if (write(dst, buf, nread) == -1){
        |             ^~~~~
copy.c:34:5: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplicit-function-declaration]
    34 |         close(src);
        |         ^~~~~
module-10 ./copy copy.c newcopy.c
module-10 ls
build copy copy.c del.c grep grep.c list list.c newcopy.c
module-10
```

Program untuk mensimulasi perintah 'rm'.

Membuat kode program dengan algorithm sebagai berikut:

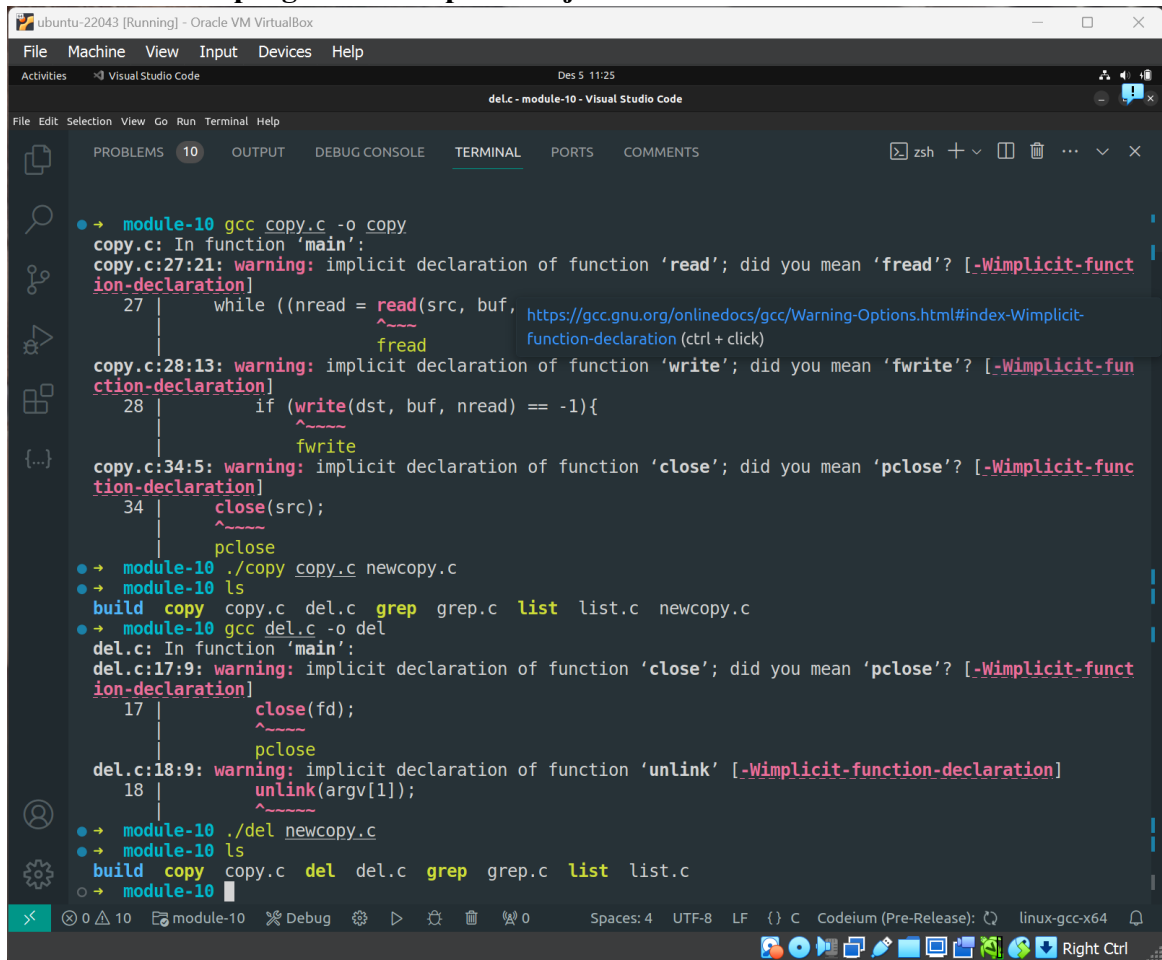
1. Gunakan nama file yang diberikan dalam argumen command line
2. Buka file dalam mode 'read-only' menggunakan perintah 'read'
3. Jika file tidak ditemukan, stop keluar program
4. Tutup file menggunakan perintah 'close'
5. Menghapus file menggunakan perintah 'unlink'
6. Stop

Kode program:



```
del.c 2 X
del.c > ...
2  #include <stdlib.h>
3  #include <fcntl.h>
4
5  int main(int argc, char *argv[])
6  {
7      int fd;
8      if (argc != 2)
9      {
10         printf("Usage: gcc del.c -o del\n");
11         printf("Usage: ./del <filename>\n");
12         exit(-1);
13     }
14
15     fd = open(argv[1], O_RDONLY);
16     if (fd != -1){
17         close(fd);
18         unlink(argv[1]);
19     } else{
20         perror(argv[1]);
21     }
22 }
23
```

Terminal ketika program dicompile & dijalankan:



```
module-10 gcc copy.c -o copy
copy.c: In function 'main':
copy.c:27:21: warning: implicit declaration of function 'read'; did you mean 'fread'? [-Wimplicit-function-declaration]
   27 |         while ((nread = read(src, buf,
      |                        ^~~~~
      |                        fread
copy.c:28:13: warning: implicit declaration of function 'write'; did you mean 'fwrite'? [-Wimplicit-function-declaration]
   28 |         if (write(dst, buf, nread) == -1){
      |             ^~~~~
      |             fwrite
copy.c:34:5: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplicit-function-declaration]
   34 |         close(src);
      |         ^~~~~
      |         pclose
module-10 ./copy copy.c newcopy.c
module-10 ls
build copy copy.c del.c grep grep.c list list.c newcopy.c
module-10 gcc del.c -o del
del.c: In function 'main':
del.c:17:9: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplicit-function-declaration]
   17 |         close(fd);
      |         ^~~~~
      |         pclose
del.c:18:9: warning: implicit declaration of function 'unlink' [-Wimplicit-function-declaration]
   18 |         unlink(argv[1]);
      |         ^~~~~
module-10 ./del newcopy.c
module-10 ls
build copy copy.c del del.c grep grep.c list list.c
module-10
```