

LAPORAN PRAKTIKUM SISTEM OPERASI



Oleh:
Ardi Hergustiyan
L200210241
Kelas E

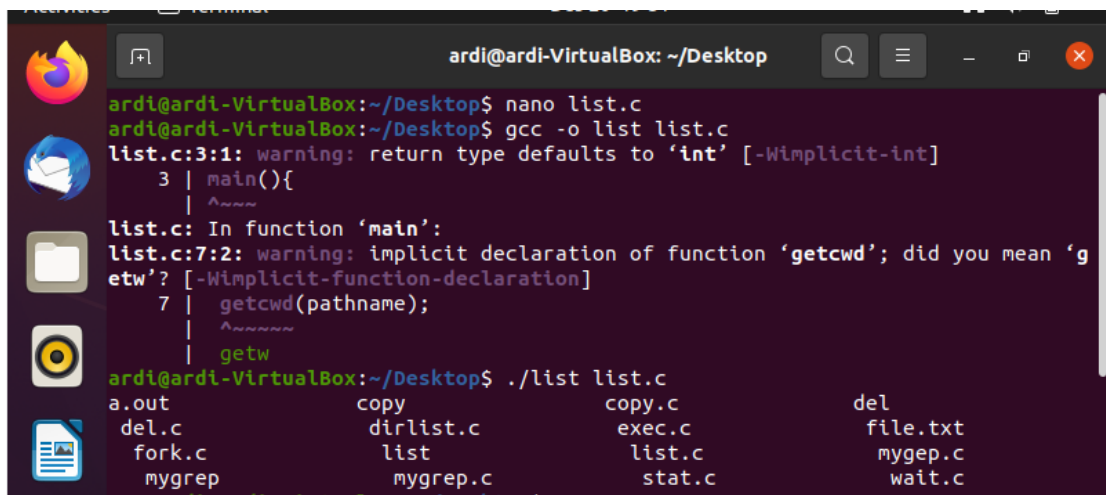
**UNIVERSITAS MUHAMMADIYAH SURAKARTA
TAHUN AJARAN 2021/2022**

Lembar Kerja Praktikum

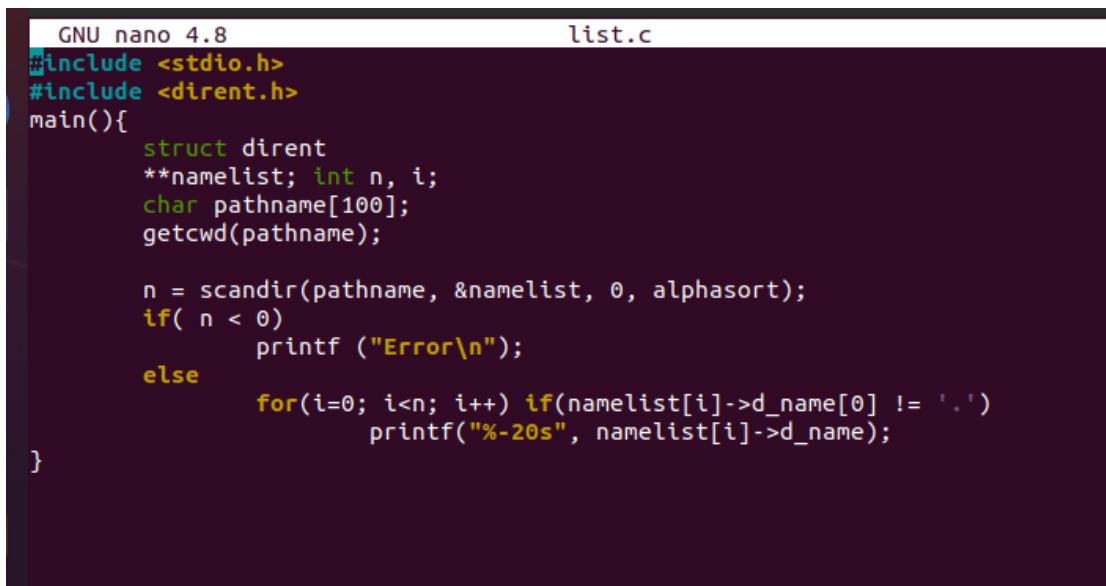
NIM	: L200210241	Nilai Praktek :
Nama	: Ardi Hergustiyan	
Nama Asisten	: -	Tanda Tangan :
	-	
Tanggal Praktikum	: 12/12/2022	

Modul 10

1. List.c untuk menampilkan file yang ada



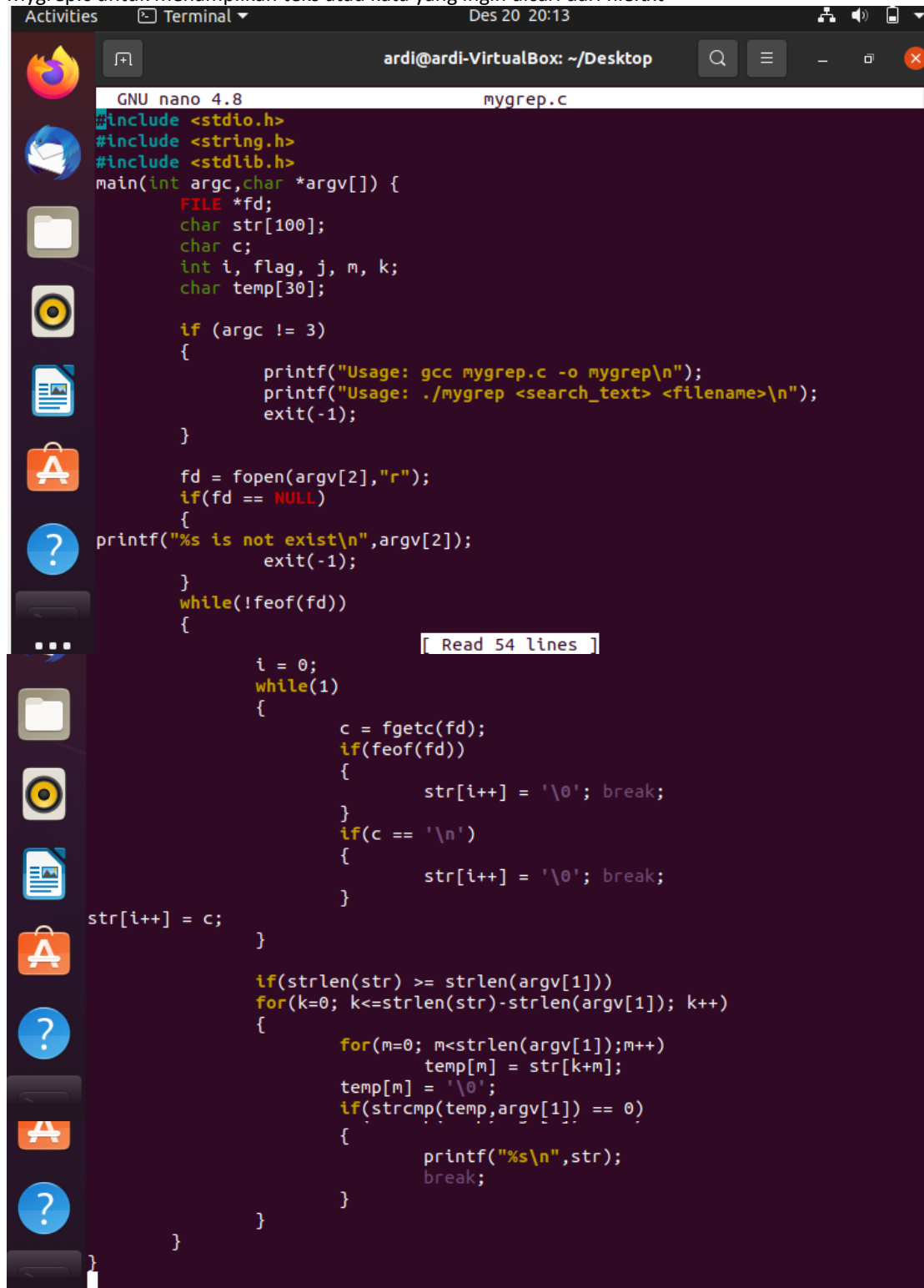
```
ardi@ardi-VirtualBox: ~/Desktop
ardi@ardi-VirtualBox:~/Desktop$ nano list.c
ardi@ardi-VirtualBox:~/Desktop$ gcc -o list list.c
list.c:3:1: warning: return type defaults to 'int' [-Wimplicit-int]
3 | main(){
  | ^~~~~~
list.c: In function 'main':
list.c:7:2: warning: implicit declaration of function 'getcwd'; did you mean 'getw'? [-Wimplicit-function-declaration]
7 |  getcwd(pathname);
  |  ^~~~~~
ardi@ardi-VirtualBox:~/Desktop$ ./list list.c
a.out          copy          copy.c        del
del.c          dirlist.c     exec.c        file.txt
fork.c         list          list.c        mygep.c
mygrep         mygrep.c     stat.c        wait.c
```



```
GNU nano 4.8 list.c
#include <stdio.h>
#include <dirent.h>
main(){
    struct dirent
    **namelist; int n, i;
    char pathname[100];
    getcwd(pathname);

    n = scandir(pathname, &namelist, 0, alphasort);
    if( n < 0)
        printf ("Error\n");
    else
        for(i=0; i<n; i++) if(namelist[i]->d_name[0] != '.')
            printf("%-20s", namelist[i]->d_name);
}
```

2. Mygrep.c untuk menampilkan teks atau kata yang ingin dicari dari file.txt



The screenshot shows a terminal window titled "ardi@ardi-VirtualBox: ~/Desktop" with a date and time of "Des 20 20:13". The terminal is running the GNU nano 4.8 editor, editing a file named "mygrep.c". The code is as follows:

```
GNU nano 4.8 mygrep.c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
main(int argc, char *argv[]) {
    FILE *fd;
    char str[100];
    char c;
    int i, flag, j, m, k;
    char temp[30];

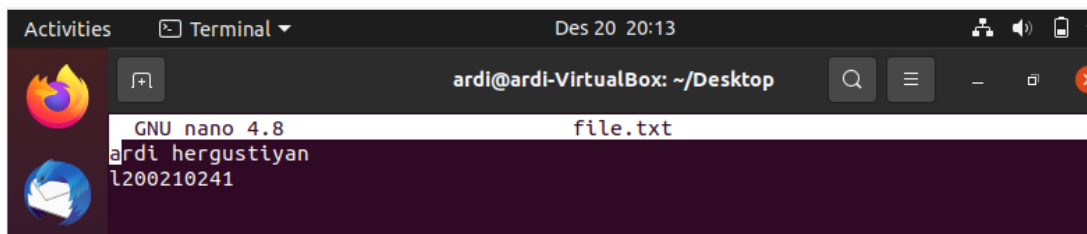
    if (argc != 3)
    {
        printf("Usage: gcc mygrep.c -o mygrep\n");
        printf("Usage: ./mygrep <search_text> <filename>\n");
        exit(-1);
    }

    fd = fopen(argv[2], "r");
    if (fd == NULL)
    {
        printf("%s is not exist\n", argv[2]);
        exit(-1);
    }
    while (!feof(fd))
    {
        i = 0;
        while (1)
        {
            c = fgetc(fd);
            if (feof(fd))
            {
                str[i++] = '\0'; break;
            }
            if (c == '\n')
            {
                str[i++] = '\0'; break;
            }
            str[i++] = c;
        }

        if (strlen(str) >= strlen(argv[1]))
        for (k = 0; k <= strlen(str) - strlen(argv[1]); k++)
        {
            for (m = 0; m < strlen(argv[1]); m++)
                temp[m] = str[k+m];
            temp[m] = '\0';
            if (strcmp(temp, argv[1]) == 0)
            {
                printf("%s\n", str);
                break;
            }
        }
    }
}
```

A status bar at the bottom of the editor indicates "Read 54 lines".

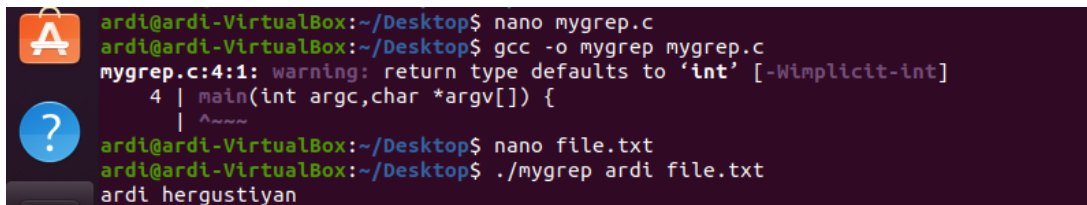
Code mygrep.c



A terminal window titled 'ardi@ardi-VirtualBox: ~/Desktop' showing the contents of a file named 'file.txt' using the nano text editor. The file contains the text 'ardi hergustiyan' followed by 'l200210241' on the next line.

```
GNU nano 4.8 file.txt
ardi hergustiyan
l200210241
```

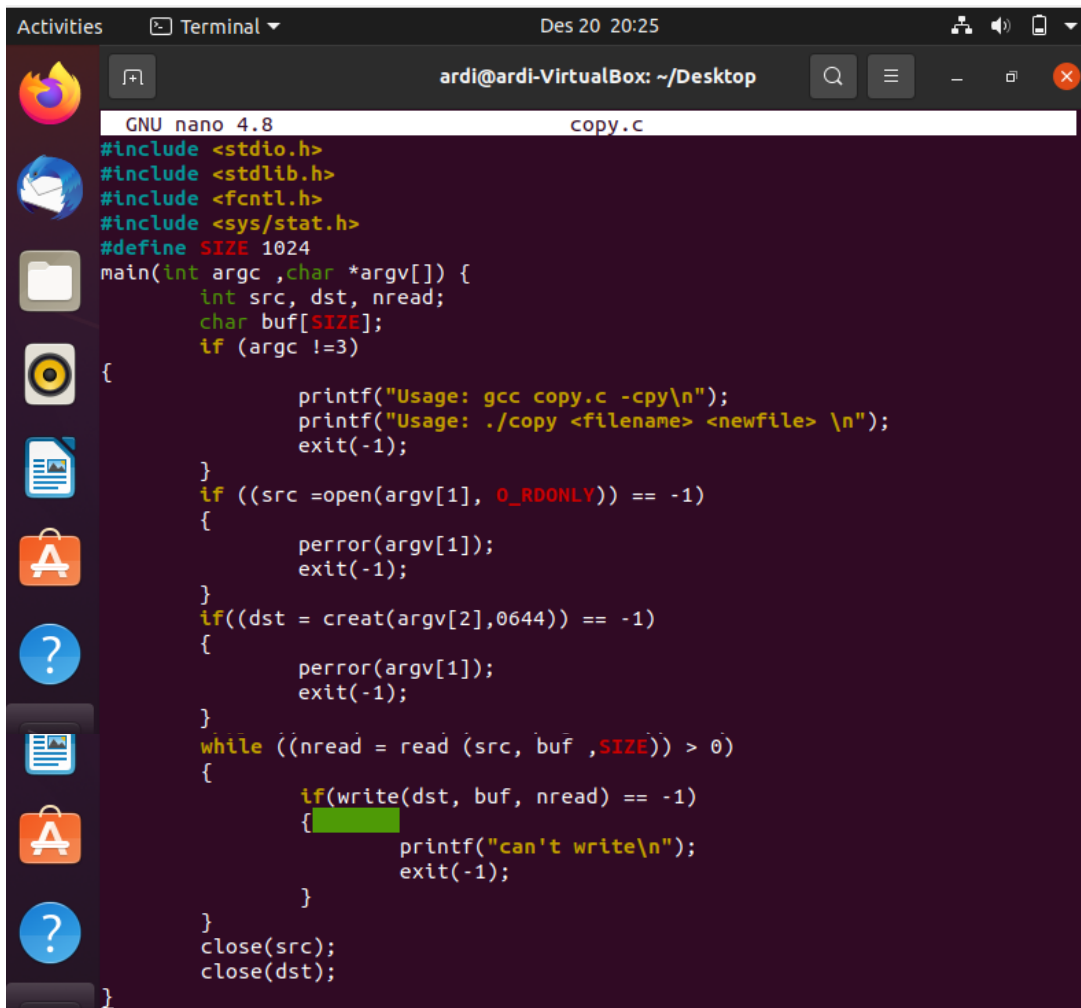
Isi teks dari file.txt



A terminal window showing the compilation and execution of a program named 'mygrep.c'. The user runs 'nano mygrep.c', then 'gcc -o mygrep mygrep.c', which shows a warning about the return type defaulting to 'int'. Then the user runs 'nano file.txt' and finally './mygrep ardi file.txt', which outputs 'ardi hergustiyan'.

```
ardi@ardi-VirtualBox:~/Desktop$ nano mygrep.c
ardi@ardi-VirtualBox:~/Desktop$ gcc -o mygrep mygrep.c
mygrep.c:4:1: warning: return type defaults to 'int' [-Wimplicit-int]
    4 | main(int argc,char *argv[]) {
      | ^
ardi@ardi-VirtualBox:~/Desktop$ nano file.txt
ardi@ardi-VirtualBox:~/Desktop$ ./mygrep ardi file.txt
ardi hergustiyan
```

3. Copy.c digunakan untuk mengcopy isi dari isi file.txt



A terminal window showing the source code of a program named 'copy.c' using the nano text editor. The code includes headers for stdio, stdlib, fcntl, and sys/stat. It defines a constant SIZE of 1024. The main function takes three arguments: argc, argv, and nread. It checks for correct usage, opens the source file, creates the destination file, and copies the contents using read and write functions. It includes error handling for file operations.

```
GNU nano 4.8 copy.c
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/stat.h>
#define SIZE 1024
main(int argc ,char *argv[]) {
    int src, dst, nread;
    char buf[SIZE];
    if (argc !=3)
    {
        printf("Usage: gcc copy.c -cpy\n");
        printf("Usage: ./copy <filename> <newfile> \n");
        exit(-1);
    }
    if ((src =open(argv[1], O_RDONLY)) == -1)
    {
        perror(argv[1]);
        exit(-1);
    }
    if((dst = creat(argv[2],0644)) == -1)
    {
        perror(argv[1]);
        exit(-1);
    }
    while ((nread = read (src, buf ,SIZE)) > 0)
    {
        if(write(dst, buf, nread) == -1)
        {
            printf("can't write\n");
            exit(-1);
        }
    }
    close(src);
    close(dst);
}
```

Program copy.c

```
Activities Terminal Des 20 20:29
ardi@ardi-VirtualBox: ~/Desktop
ardi hergustiyan
ardi@ardi-VirtualBox:~/Desktop$ nano copy.c
ardi@ardi-VirtualBox:~/Desktop$ gcc -o copy copy.c
copy.c:6:1: warning: return type defaults to 'int' [-Wimplicit-int]
6 | main(int argc, char *argv[]) {
  |
copy.c: In function 'main':
copy.c:25:18: warning: implicit declaration of function 'read'; did you mean 'f
read'? [-Wimplicit-function-declaration]
25 | while ((nread = read (src, buf, SIZE)) > 0)
  |                  ^~~~~
copy.c:27:6: warning: implicit declaration of function 'write'; did you mean 'f
write'? [-Wimplicit-function-declaration]
27 | if(write(dst, buf, nread) == -1)
  |    ^~~~~
copy.c:33:2: warning: implicit declaration of function 'close'; did you mean 'p
close'? [-Wimplicit-function-declaration]
33 | close(src);
  |    ^~~~~
pclose
```

```
ardi@ardi-VirtualBox:~/Desktop$ ./copy file.txt file2.txt
ardi@ardi-VirtualBox:~/Desktop$ dir
a.out del exec.c fork.c mygep.c stat.c
copy del.c file2.txt list mygrep wait.c
copy.c dirlist.c file.txt list.c mygrep.c
ardi@ardi-VirtualBox:~/Desktop$ nano file2.txt
ardi@ardi-VirtualBox:~/Desktop$ nano del.c
```

Proses mengCopy file.txt ke file2.txt dan melihat isi direktori apakah file2.txt sudah ada

```
Activities Terminal Des 20 20:29
ardi@ardi-VirtualBox: ~/Desktop
GNU nano 4.8 file2.txt
ardi hergustiyan
l200210241
```

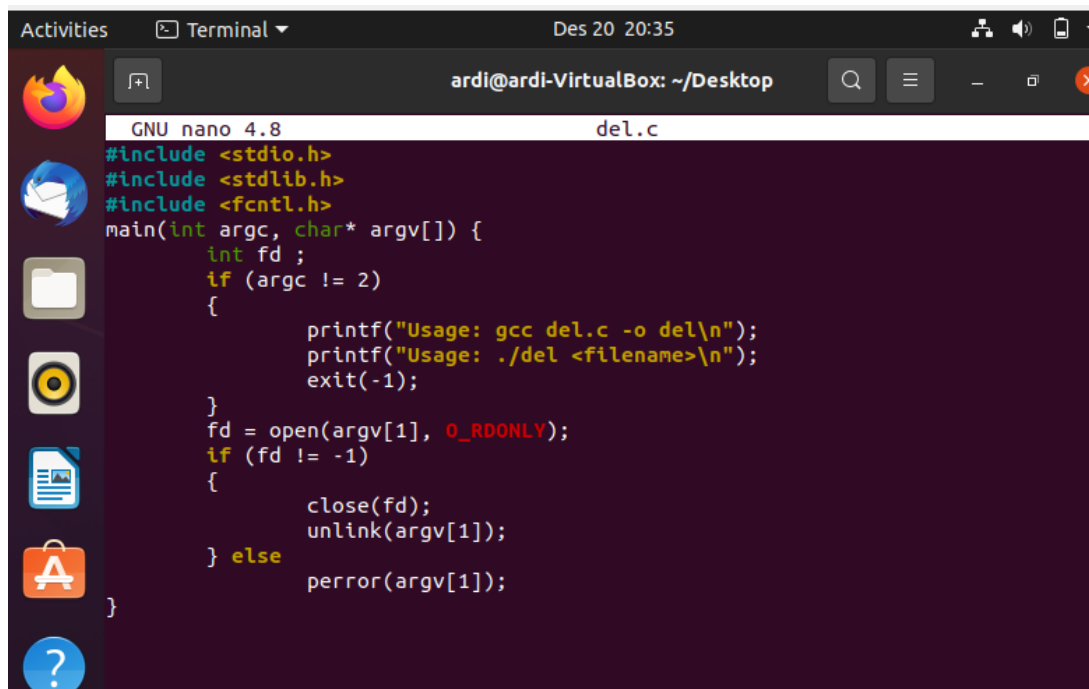
Isi file2.txt sudah mencopy file.txt

4. Del.c untuk menghapus file

```
Activities Terminal Des 20 20:34
ardi@ardi-VirtualBox: ~/Desktop
ardi@ardi-VirtualBox:~/Desktop$ nano del.c
ardi@ardi-VirtualBox:~/Desktop$ gcc -o del del.c
del.c:4:1: warning: return type defaults to 'int' [-Wimplicit-int]
4 | main(int argc, char* argv[]) {
  |
del.c: In function 'main':
del.c:15:3: warning: implicit declaration of function 'close'; did you mean 'pc
lose'? [-Wimplicit-function-declaration]
15 | close(fd);
  |    ^~~~~
pclose
del.c:16:3: warning: implicit declaration of function 'unlink' [-Wimplicit-func
tion-declaration]
16 | unlink(argv[1]);
  |    ^~~~~
```

```
ardi@ardi-VirtualBox:~/Desktop$ ./del file.txt
ardi@ardi-VirtualBox:~/Desktop$ dir
a.out  copy.c  del.c    exec.c   fork.c   list.c   mygrep   stat.c
copy   del     dirlist.c file2.txt list     mygep.c mygrep.c wait.c
```

file.txt sudah terdelete dari direktori



The screenshot shows a terminal window titled "ardi@ardi-VirtualBox: ~/Desktop" with a timestamp of "Des 20 20:35". The window contains the GNU nano 4.8 editor displaying the source code for del.c. The code includes headers for stdio.h, stdlib.h, and fcntl.h. The main function takes argc and argv as arguments. It checks if argc is not equal to 2, and if so, it prints usage information and exits with a status of -1. If argc is 2, it opens the file specified in argv[1] in read-only mode. If the file is successfully opened, it closes the file descriptor and unlinks the file. If the file cannot be opened, it prints an error message.

```
GNU nano 4.8 del.c
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
main(int argc, char* argv[]) {
    int fd ;
    if (argc != 2)
    {
        printf("Usage: gcc del.c -o del\n");
        printf("Usage: ./del <filename>\n");
        exit(-1);
    }
    fd = open(argv[1], O_RDONLY);
    if (fd != -1)
    {
        close(fd);
        unlink(argv[1]);
    } else
        perror(argv[1]);
}
```

Code del.c