

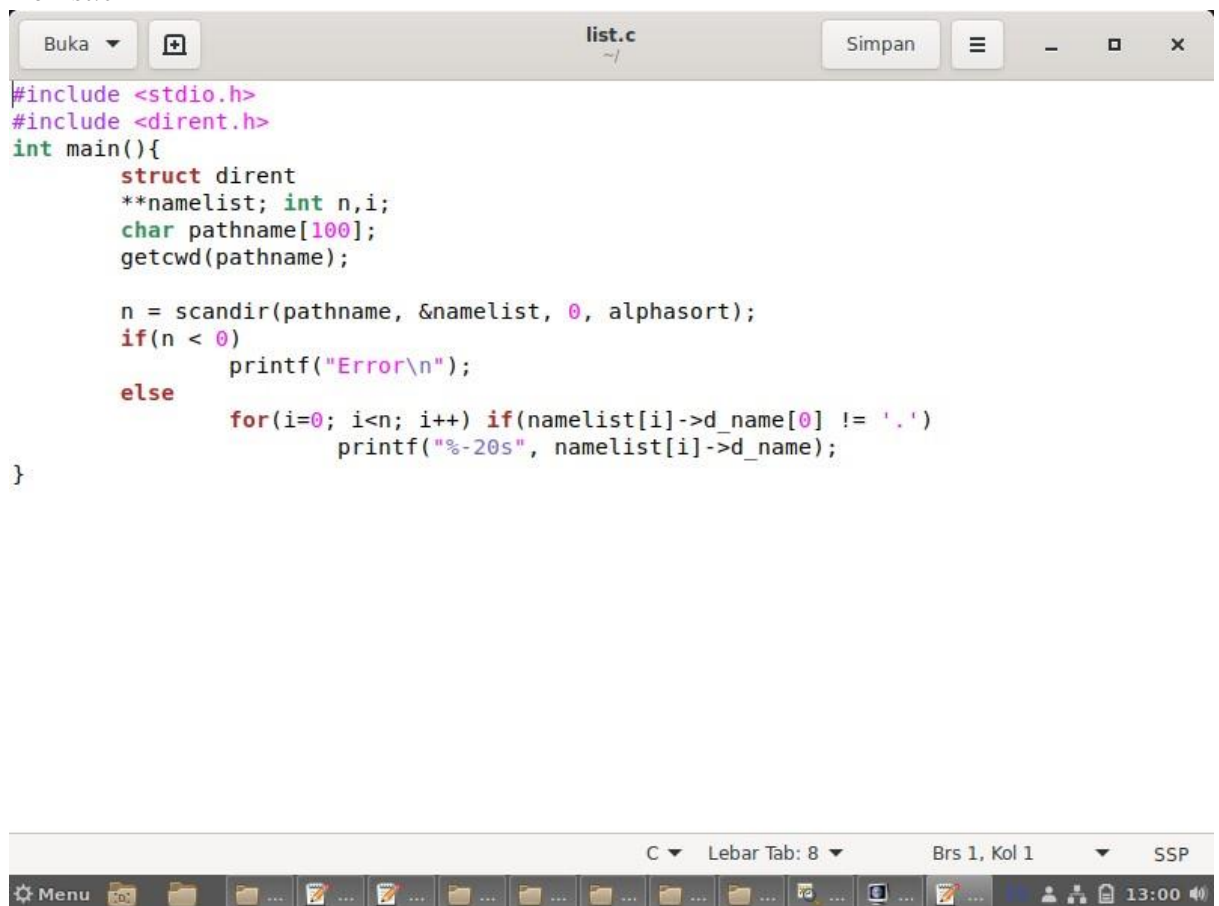
Nama : Karina Muslimah

NIM : L200180138

Kelas : C

Modul 10

1. File list.c



```
#include <stdio.h>
#include <dirent.h>
int 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);
}
```

The image shows a code editor window titled 'list.c' with a menu bar containing 'Buka', '+', and 'Simpan'. The code is a C program that lists files in the current directory using the `scandir` function. The code is as follows:

```
#include <stdio.h>
#include <dirent.h>
int 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);
}
```

Below the code editor, the Windows taskbar is visible, showing the 'Menu' button, several open application icons, and the system clock displaying '13:00'.

2. File

```
mygrep.c
Buka + mygrep.c ~/ Simpan ≡ _ □ ×

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int 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)
        {
            write(i)
            {
                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;
                }
            }
        }
    }
}
```

C ▾ Lebar Tab: 8 ▾ Brs 1, Kol 1 ▾ SSP

Menu 13:03

Buka + mygrep.c ~/ Simpan ≡ _ □ ×

C ▾ Lebar Tab: 8 ▾ Brs 1, Kol 1 ▾ SSP

Menu 13:03

3. File

copy.c

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/stat.h>
#define SIZE 1024
int main(int argc, char *argv[] ){
    int src, dst, nread;
    char buf[SIZE];
    if (argc != 3)
    {
        printf("Usage: gcc copy.c -o copy\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);
}
```

copy.c

```
char buf[SIZE];
if (argc != 3)
{
    printf("Usage: gcc copy.c -o copy\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);
}
```

4. File

```
del.c
# include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
int 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]);
}
```

C Lebar Tab: 8 Brs 1, Kol 1 SSP

5. Running program

```
mlterm
acer@debian:~$ nano list.c
acer@debian:~$ gcc list.c -o list
list.c: In function 'main':
list.c:7:2: warning: implicit declaration of function 'getcwd'; did you mean g
etw'? [-Wimplicit-function-declaration]
    getcwd(pathname);
    ^
getw
acer@debian:~$ nano mygrep.c
acer@debian:~$ gcc mygrep.c -o mygrep
acer@debian:~$ nano copy.c
acer@debian:~$ gcc copy.c -o copy
copy.c: In function 'main':
copy.c:12:10: warning: unknown escape sequence: '\c'
    printf("Usage: .\copy <filename> <newfile> \n");
    ^
copy.c:25:17: warning: implicit declaration of function 'read'; did you mean f
read'? [-Wimplicit-function-declaration]
    while((nread = read(src, buf, SIZE)) > 0)
                  ^
fread
copy.c:27:7: warning: implicit declaration of function 'write'; did you mean f
write'? [-Wimplicit-function-declaration]
    if (write(dst, buf, nread) == -1)
```



```
mlterm
while((nread = read(src, buf, SIZE)) > 0)
    ^
    fread
copy.c:27:7: warning: implicit declaration of function 'write'; did you mean f
write'? [-Wimplicit-function-declaration]
    if (write(dst, buf, nread) == -1)
        ^
        fwrite
copy.c:33:2: warning: implicit declaration of function 'close'; did you mean p
close'? [-Wimplicit-function-declaration]
    close(src);
    ^
    pclose
acer@debian:~$ nano del.c
acer@debian:~$ gcc del.c -o del
del.c: In function 'main':
del.c:15:3: warning: implicit declaration of function 'close'; did you mean 'p
close'? [-Wimplicit-function-declaration]
    close(fd);
    ^
    pclose
del.c:16:3: warning: implicit declaration of function 'unlink'; did you mean u
nix'? [-Wimplicit-function-declaration]
    unlink(argv[1]);
```

```
mlterm
    close(fd);
    ^
    pclose
del.c:16:3: warning: implicit declaration of function 'unlink'; did you mean u
nix'? [-Wimplicit-function-declaration]
    unlink(argv[1]);
    ^
    unix
acer@debian:~$ ./list
Error
acer@debian:~$ ls
a.out    del.c    fappend  fread    list     mygrep.c  Templat
copy     Destop   fappend.c fread.c   list.c   nano.save  Unduhan
copy.c   dirlist.c fcreate  Gambar   Musik    Publik     Video
del      Dokumen fcreate.c l200180122.txt mygrep   stat.c
acer@debian:~$ ./list
Error
acer@debian:~$ ./mygrep anisa l200180135.txt
l200180135.txt is not exist
acer@debian:~$ nano l200180135.txt
acer@debian:~$ ./mygrep anisa l200180135.txt
nama = anisa
acer@debian:~$ ./copy l200180135.txt anisa.txt
acer@debian:~$ ls
```

```
mIterm
copy.c  dirlist.c  fcreate  Gambar  Musik  Publik  Video
del     Dokumen  fcreate.c  l200180122.txt  mygrep  stat.c
acer@debian:~$ ./list
Error
acer@debian:~$ ./mygrep anisa l200180135.txt
l200180135.txt is not exist
acer@debian:~$ nano l200180135.txt
acer@debian:~$ ./mygrep anisa l200180135.txt
nama = anisa
acer@debian:~$ ./copy l200180135.txt anisa.txt
acer@debian:~$ ls
anisa.txt  del.c      fappend.c  Gambar  Musik  stat.c
a.out     Destop    fcreate    l200180122.txt  mygrep  Templat
copy      dirlist.c fcreate.c  l200180135.txt  mygrep.c Unduhan
copy.c    Dokumen  fread      list      nano.save Video
del       fappend  fread.c    list.c      Publik
acer@debian:~$ ./del anisa.txt
acer@debian:~$ ls
a.out     Destop    fcreate    l200180122.txt  mygrep  Templat
copy      dirlist.c fcreate.c  l200180135.txt  mygrep.c Unduhan
copy.c    Dokumen  fread      list      nano.save Video
del       fappend  fread.c    list.c      Publik
del.c     fappend.c Gambar  Musik  stat.c
acer@debian:~$
```