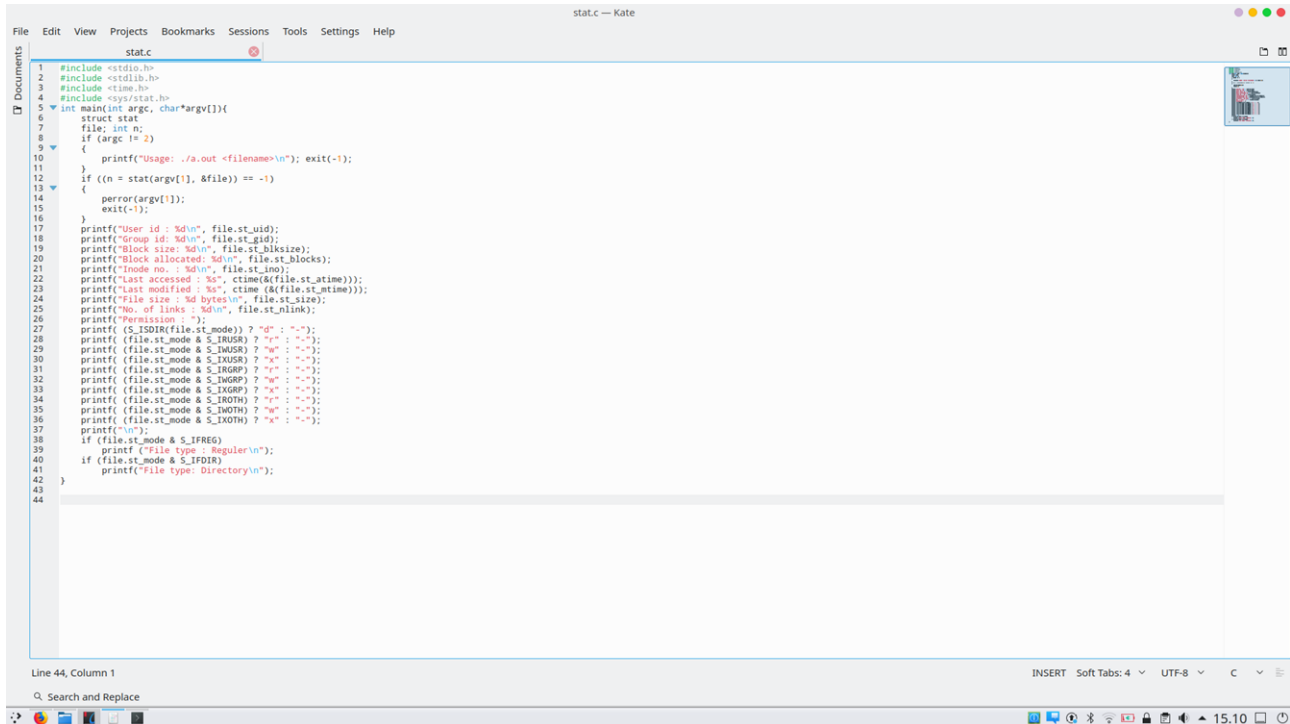


Nama : Berlian Vidia Puspa
NIM : L200180107

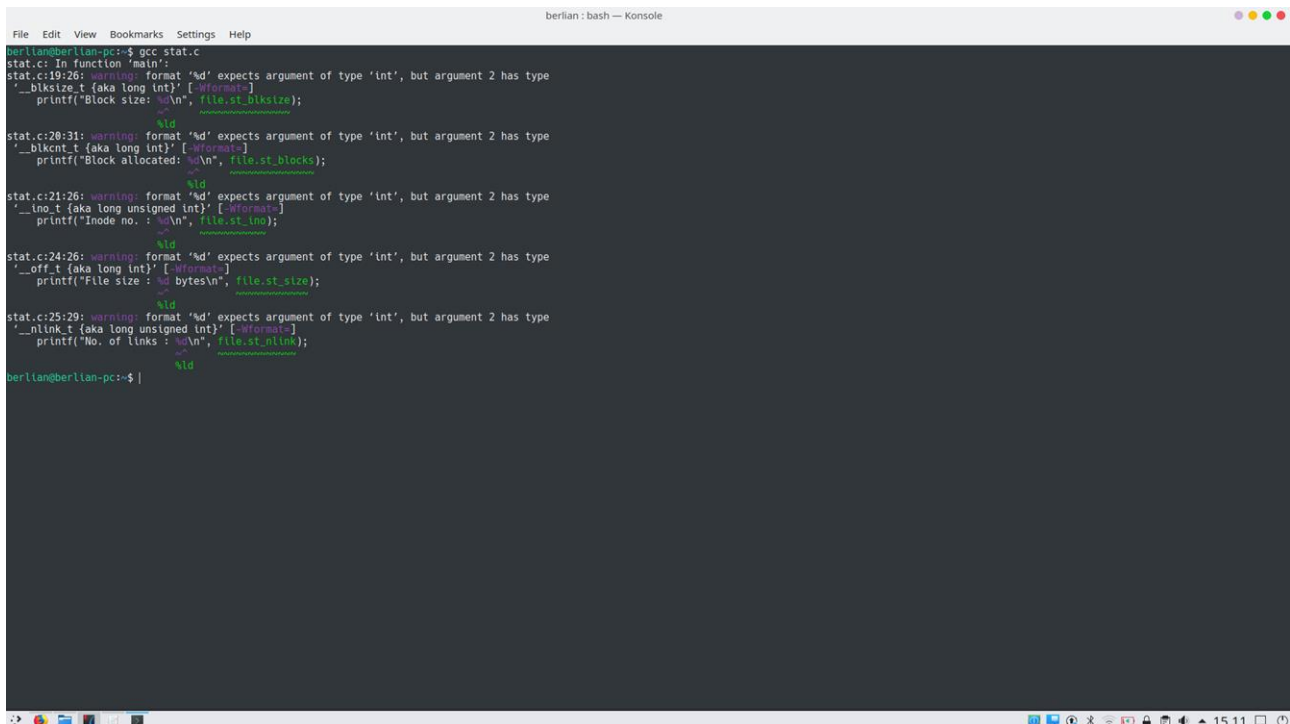
Modul 8 Praktikum Sistem Operasi

4.



The screenshot shows a code editor window titled "stat.c - Kate". The code is a C program that uses the `stat` system call to display file information. It includes headers for `stdio.h`, `stdlib.h`, `time.h`, and `sys/stat.h`. The `main` function takes an argument `argc` and a character array `argv`. It checks if `argc` is 2, and if so, it prints the usage. Otherwise, it calls `stat` on `argv[1]` and prints various file attributes including user ID, group ID, block size, blocks, inode number, last accessed time, file size, number of links, and permissions. It also prints the file type (regular or directory) based on the `S_IFREG` and `S_IFDIR` flags.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4 #include <sys/stat.h>
5 int main(int argc, char*argv[]){
6     struct stat
7     file; int n;
8     if (argc != 2)
9     {
10         printf("Usage: ./a.out <filename>\n"); exit(-1);
11     }
12     if ((n = stat(argv[1], &file)) == -1)
13     {
14         perror(argv[1]);
15         exit(-1);
16     }
17     printf("User id : %d\n", file.st_uid);
18     printf("Group id : %d\n", file.st_gid);
19     printf("Block size: %d\n", file.st_blksize);
20     printf("Block allocated: %d\n", file.st_blocks);
21     printf("Inode no. : %d\n", file.st_ino);
22     printf("Last accessed : %s", ctime(&file.st_atime));
23     printf("Last modified : %s", ctime(&file.st_mtime));
24     printf("File size : %d bytes\n", file.st_size);
25     printf("No. of links : %d\n", file.st_nlink);
26     printf("Permission : ");
27     printf(" (%d\n", file.st_mode);
28     printf(" (%d\n", file.st_mode & S_IRUSR);
29     printf(" (%d\n", file.st_mode & S_IWUSR);
30     printf(" (%d\n", file.st_mode & S_IXUSR);
31     printf(" (%d\n", file.st_mode & S_IRGRP);
32     printf(" (%d\n", file.st_mode & S_IWGRP);
33     printf(" (%d\n", file.st_mode & S_IXGRP);
34     printf(" (%d\n", file.st_mode & S_IROTH);
35     printf(" (%d\n", file.st_mode & S_IWOTH);
36     printf(" (%d\n", file.st_mode & S_IXOTH);
37     printf("\n");
38     if (file.st_mode & S_IFREG)
39         printf("File type : Regular\n");
40     if (file.st_mode & S_IFDIR)
41         printf("File type : Directory\n");
42 }
43
44
```



The screenshot shows a terminal window titled "berlian : bash - Konsole". It displays the compilation and execution of the `stat.c` program. The user runs `gcc stat.c`, which produces several warnings about format specifiers. The user then runs `./a.out`, which displays the file information for the current directory.

```
berlian@berlian-pc:~$ gcc stat.c
stat.c: In function 'main':
stat.c:19:26: warning: format '%d' expects argument of type 'int', but argument 2 has type
'_blksize_t {aka long int}' [-Wformat=]
printf("Block size: %d\n", file.st_blksize);
^
stat.c:20:26: warning: format '%d' expects argument of type 'int', but argument 2 has type
'_blkcnt_t {aka long int}' [-Wformat=]
printf("Block allocated: %d\n", file.st_blocks);
^
stat.c:21:26: warning: format '%d' expects argument of type 'int', but argument 2 has type
'_ino_t {aka long unsigned int}' [-Wformat=]
printf("Inode no. : %d\n", file.st_ino);
^
stat.c:24:26: warning: format '%d' expects argument of type 'int', but argument 2 has type
'_off_t {aka long int}' [-Wformat=]
printf("File size : %d bytes\n", file.st_size);
^
stat.c:25:29: warning: format '%d' expects argument of type 'int', but argument 2 has type
'_nlink_t {aka long unsigned int}' [-Wformat=]
printf("No. of links : %d\n", file.st_nlink);
^
berlian@berlian-pc:~$
```

5.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <dirent.h>
4 int main(int argc, char*argv[]){
5     struct dirent *dptr;
6     DIR *dname;
7
8     if (argc != 2)
9     {
10         printf("Usage: ./a.out <dirname>\n");
11         exit(-1);
12     }
13     if ((dname = opendir(argv[1])) == NULL)
14     {
15         perror(argv[1]);
16         exit(-1);
17     }
18     while (dptr=readdir(dname))
19         printf("%s\n", dptr->d_name);
20     closedir(dname);
21 }
22
23
```

```
berlian@berlian-pc:~$ gcc dirlist.c
berlian@berlian-pc:~$ ./a.out /home/berlian
.recently-used
.mozilla
.local
.bash_history
.Xauthority
.python_history
.bashrc
.snap
.NetBeansProjects
.Java
.netbeans-8.2
.hedgewars
.Templates
.gtkrc-2.0
.face.icon
.VirtualBox VMs
.stat.c
.xvnc
.Public
..
.Downloads
.cache
.nbl
.face
.Music
.Pictures
.Ade
.netbeans
.gnupg
.Workspaces
.gconf
.gimp-2.8
.directory
.Desktop
.fonts
.Videos
.xsession-errors
dirlist.c
.sudo_as_admin_successful
.Documents
.pkt
.fonts.conf
.qr-scanner-oprec-fostlums
.thumbnails
.profile
a.out
.bash_logout
.dbus
.gnome
.config
berlian@berlian-pc:~$
```