

Naufal Randi Aldiansyah  
L200180142  
Modul 8

## File stat.c

The screenshot shows a Windows 10 desktop environment. The primary application is Visual Studio Code (VS Code), which is open to a C source file named `stat.c`. The code is a program that demonstrates the use of the `stat` system call to retrieve file metadata. It includes headers for `stdio.h`, `stdlib.h`, `unistd.h`, and `sys/stat.h`. The `main` function takes an array of arguments and processes them. It uses `stat` to get file information and prints it in a structured format, including file type, permissions, owner/group IDs, block size, device, inode, last access/modification times, file size, number of links, and permissions in octal and symbolic notation. The file explorer on the left shows the project structure, including a `main.c` file. The taskbar at the bottom displays various application icons, including the Start button, File Explorer, and several utility applications.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <sys/stat.h>
5
6  int main(int argc, char*argv[]){
7      struct stat
8          file; int a;
9          if (argc == 2)
10             {
11                 printf("Usage: ./a.out <filename>\n"); exit(1);
12             }
13             if (a = stat(argv[1], &file) == -1)
14             {
15                 perror(argv[1]);
16                 exit(1);
17             }
18             printf("User id: %d\n", file.st_uid);
19             printf("Group id: %d\n", file.st_gid);
20             printf("Block size: %d\n", file.st_blksize);
21             printf("Block allocated: %d\n", file.st_blocks);
22             printf("Mode no. : %d\n", file.st_mode);
23             printf("Last accessed: %s", ctime(&file.st_atime));
24             printf("Last modified: %s", ctime(&file.st_mtime));
25             printf("File size: %d bytes\n", file.st_size);
26             printf("No. of links : %d\n", file.st_nlink);
27             printf("Permission : ");
28             printf(" (%5ldIR(file.st.mode)) ? 'd' : '-');
29             printf(" (%5ldIR(file.st.mode) & S_IRUSR) ? 'r' : '-');
30             printf(" (%5ldIR(file.st.mode) & S_IRUSR) ? 'w' : '-');
31             printf(" (%5ldIR(file.st.mode) & S_IRUSR) ? 'x' : '-');
32             printf(" (%5ldIR(file.st.mode) & S_IRGRP) ? 'r' : '-');
33             printf(" (%5ldIR(file.st.mode) & S_IRGRP) ? 'w' : '-');
34             printf(" (%5ldIR(file.st.mode) & S_IRGRP) ? 'x' : '-');
35             printf(" (%5ldIR(file.st.mode) & S_IROTH) ? 'r' : '-');
36             printf(" (%5ldIR(file.st.mode) & S_IROTH) ? 'w' : '-');
37             printf(" (%5ldIR(file.st.mode) & S_IROTH) ? 'x' : '-');
38             printf("\n");
39             if (file.st_mode & S_IFREG)
40             {
41                 printf("File type : Regular\n");
42             }
43             if (file.st_mode & S_IFDIR)
44             {
45                 printf("File type: Directory\n");
46             }
47         }
48     }

```

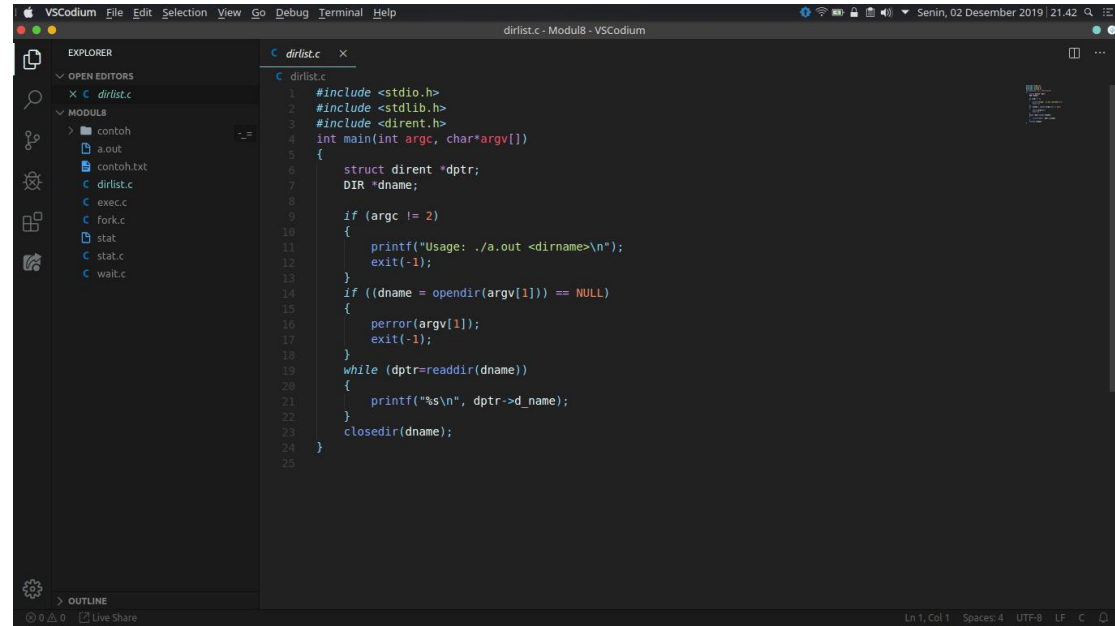
Hasil compile stat.c

```

(base) me@me:~/Downloads/Modul8$ gcc stat.c
Modul8: bash — Konsole
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);
                        ^~
                        |
                        |__blksize_t {aka long int}
                        |
                        |__blksize_t {aka long int}
stat.c:20:31: 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);
                        ^~
                        |
                        |__blkcnt_t {aka long int}
                        |
                        |__blkcnt_t {aka long int}
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);
                        ^~
                        |
                        |__ino_t {aka long unsigned int}
                        |
                        |__ino_t {aka long unsigned int}
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);
                        ^~
                        |
                        |__off_t {aka long int}
                        |
                        |__off_t {aka long int}
stat.c:25:29: warning: format '%d' expects argument of type 'int', but argument 2 has type '__link_t {aka long unsigned int}' [-Wformat=]
printf("No. of links : %d\n", file.st_nlink);
                        ^~
                        |
                        |__link_t {aka long unsigned int}
                        |
                        |__link_t {aka long unsigned int}
(base) me@me:~/Downloads/Modul8$
Modul8: bash

```

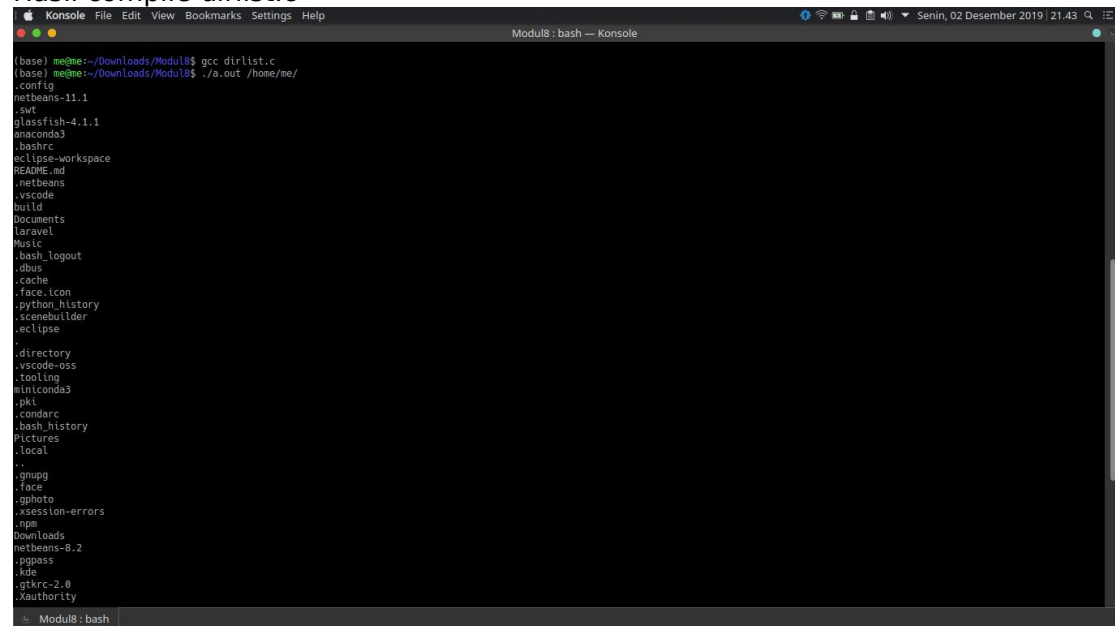
## File dirlist.c



The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left displays a project structure with a 'MODULES' folder containing several files: 'a.out', 'a.out.txt', 'dirlist.c', 'exec.c', 'fork.c', 'stat', 'stat.c', and 'wait.c'. The main editor window is open to 'dirlist.c', showing the following C code:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <dirent.h>
4 int main(int argc, char*argv[])
5 {
6     struct dirent *dptr;
7     DIR *dname;
8
9     if (argc != 2)
10    {
11        printf("Usage: ./a.out <dirname>\n");
12        exit(-1);
13    }
14    if ((dname = opendir(argv[1])) == NULL)
15    {
16        perror(argv[1]);
17        exit(-1);
18    }
19    while (dptr=readdir(dname))
20    {
21        printf("%s\n", dptr->d_name);
22    }
23    closedir(dname);
24 }
25
```

## Hasil compile dirlist.c



The screenshot shows a terminal window titled 'Modul8: bash — Konsole'. It displays the command to compile 'dirlist.c' and the resulting directory listing of the current directory:

```
(base) me@me:~/Downloads/Modul8$ gcc dirlist.c
(base) me@me:~/Downloads/Modul8$ ./a.out /home/me/
.config
.netbeans-11.1
.swt
glassfish-4.1.1
anaconda3
.bashrc
eclipse-workspace
README.md
.netbeans
.vscode
build
Documents
laravel
Music
.bash_logout
.dbus
.cache
.face.icon
.python_history
seabuilder
eclipse
.directory
.vscode-oss
.tooling
miniconda3
.pki
.condarc
.bash_history
Pictures
.local
.gnupg
.face
.gphoto
.xsession-errors
.rgm
Downloads
.netbeans-8.2
.pgpass
.kde
.gtkrc-2.0
.xauthority
```