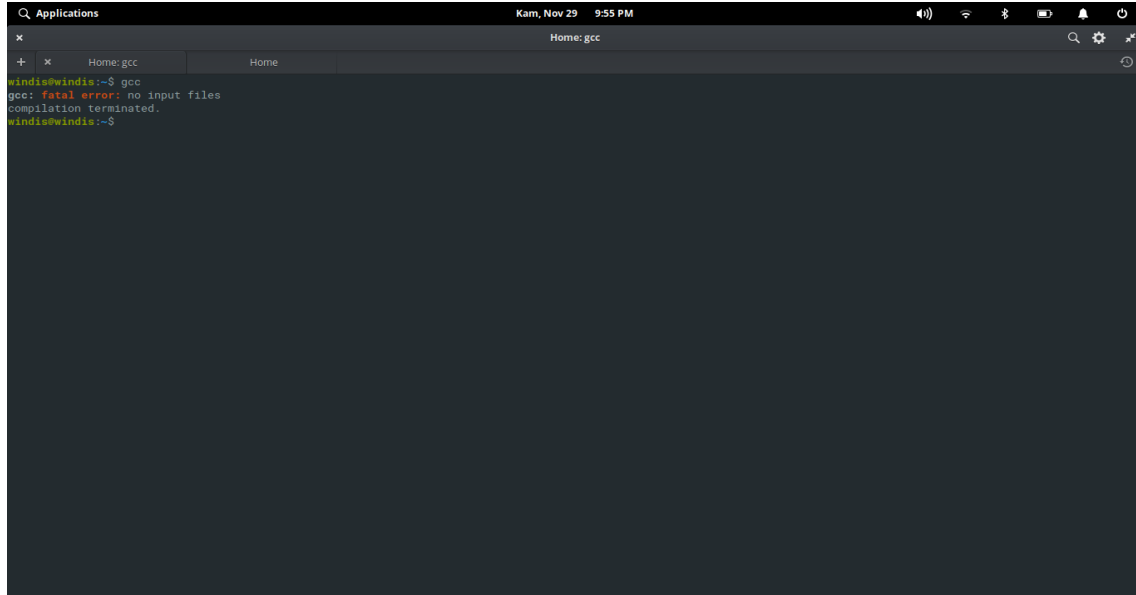


NAMA : WINDI SAPUTRI
KELAS: E
NIM : L200170115

A. Ketik “gcc” pada terminal linux

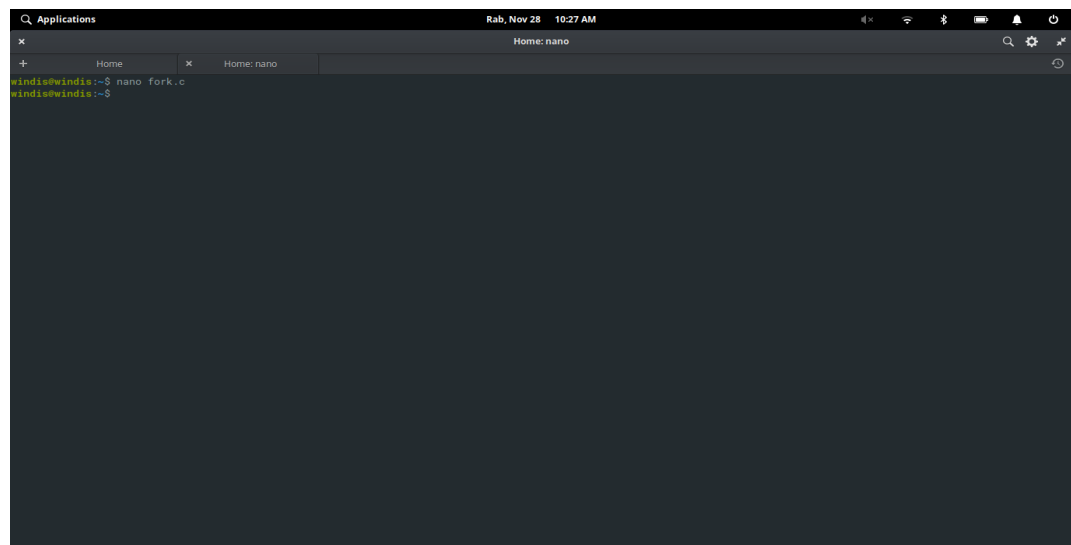


The screenshot shows a terminal window titled "Applications" with a dark background. The terminal prompt is "windi@windi:~\$". The user has entered the command "gcc". The output shows a "fatal error: no input files" and "compilation terminated." followed by a new prompt "windi@windi:~\$".

```
windi@windi:~$ gcc
gcc: fatal error: no input files
compilation terminated.
windi@windi:~$
```

B. Fork.c

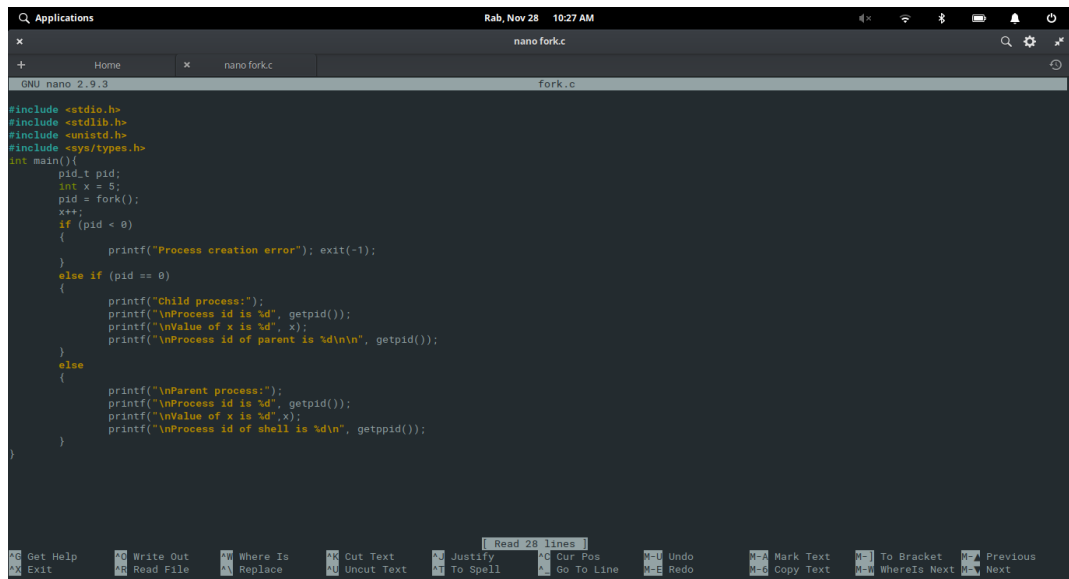
1. Ketik perintah “nano fork.c” untuk membuat file bernama fork



The screenshot shows a terminal window titled "Applications" with a dark background. The terminal prompt is "windi@windi:~\$". The user has entered the command "nano fork.c". The output shows the command being executed and a new prompt "windi@windi:~\$".

```
windi@windi:~$ nano fork.c
windi@windi:~$
```

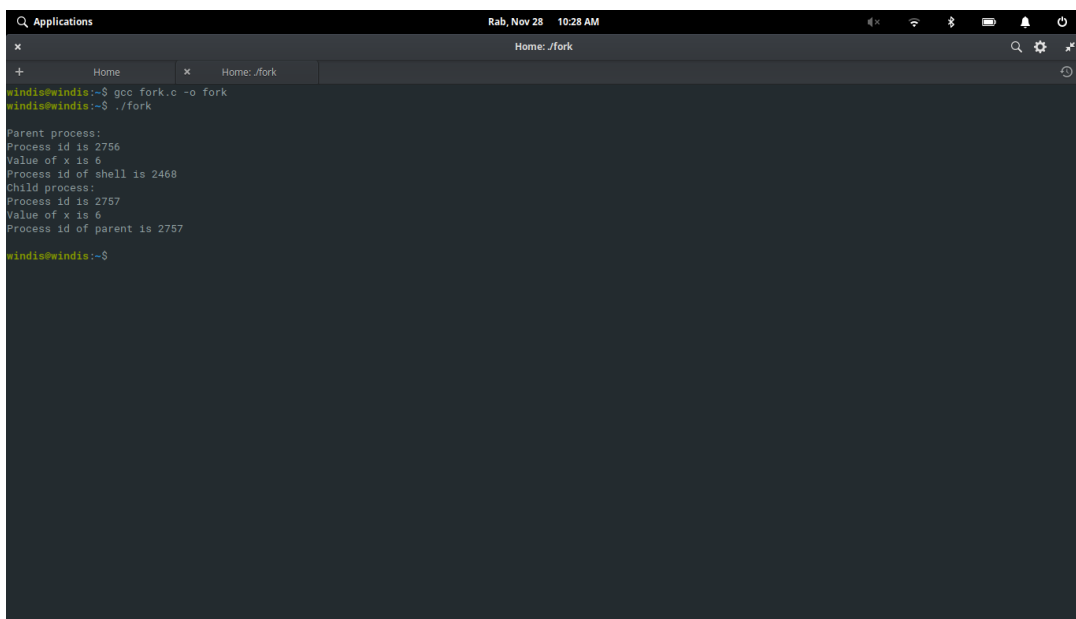
2. Lalu ketik perintah yang terdapat pada modul
3. Tekan ctrl+x untuk menyimpan file, lalu ketik “Y”, lalu enter



```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>

int main(){
    pid_t pid;
    int x = 5;
    pid = fork();
    x++;
    if (pid < 0)
    {
        printf("Process creation error"); exit(-1);
    }
    else if (pid == 0)
    {
        printf("Child process:");
        printf("\nProcess id is %d", getpid());
        printf("\nValue of x is %d", x);
        printf("\nProcess id of parent is %d\n\n", getpid());
    }
    else
    {
        printf("\nParent process:");
        printf("\nProcess id is %d", getpid());
        printf("\nValue of x is %d", x);
        printf("\nProcess id of shell is %d\n", getppid());
    }
}
```

4. Ketik “gcc fork.c -o fork” untuk mengecek apakah codingan sudah benar atau belum.
5. Ketik “./fork” untuk menampilkan outputnya



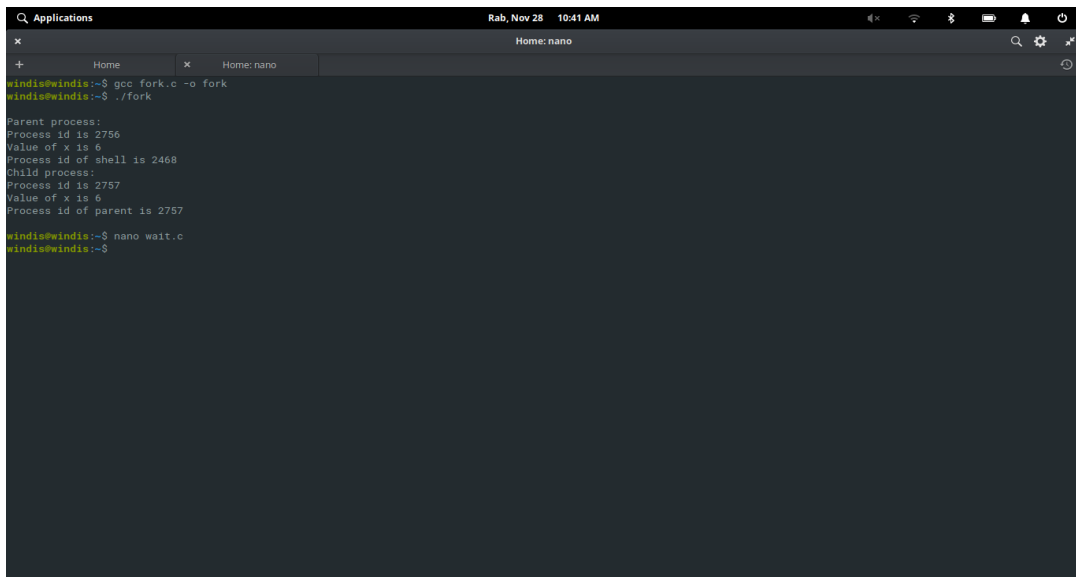
```
windis@windis:~$ gcc fork.c -o fork
windis@windis:~$ ./fork

Parent process:
Process id is 2756
Value of x is 6
Process id of shell is 2468
Child process:
Process id is 2757
Value of x is 6
Process id of parent is 2757

windis@windis:~$
```

C. wait.c

1. Ketik perintah “nano wait.c” untuk membuat file bernama wait

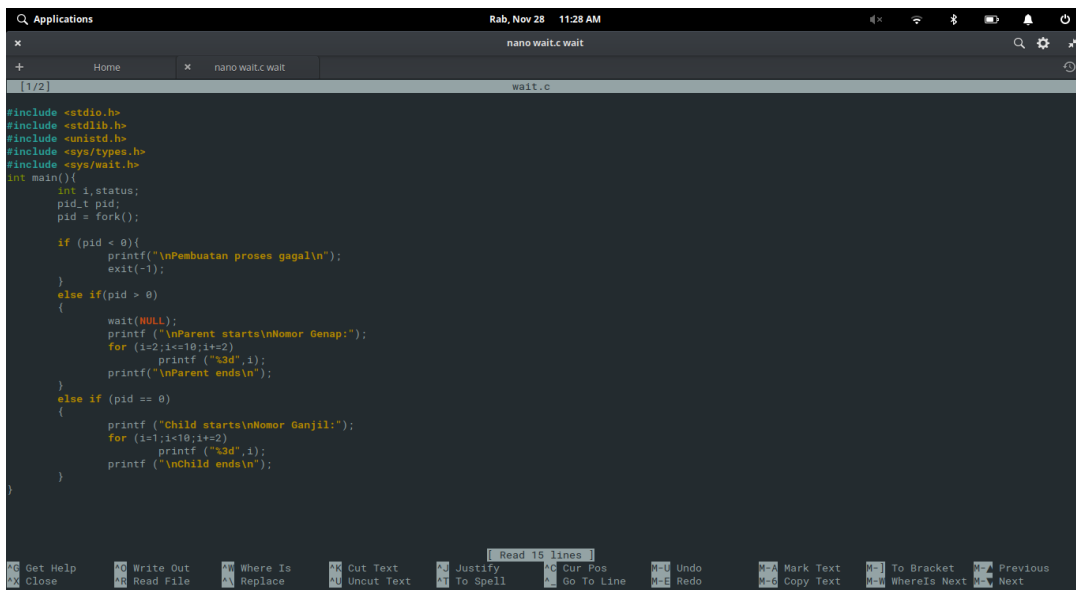


```
windis@windis:~$ gcc fork.c -o fork
windis@windis:~$ ./fork

Parent process:
Process id is 2756
Value of x is 6
Process id of shell is 2468
Child process:
Process id is 2757
Value of x is 6
Process id of parent is 2757

windis@windis:~$ nano wait.c
windis@windis:~$
```

2. Lalu ketik perintah yang terdapat pada modul
3. Tekan ctrl+x untuk menyimpan file, lalu ketik “Y”, lalu enter



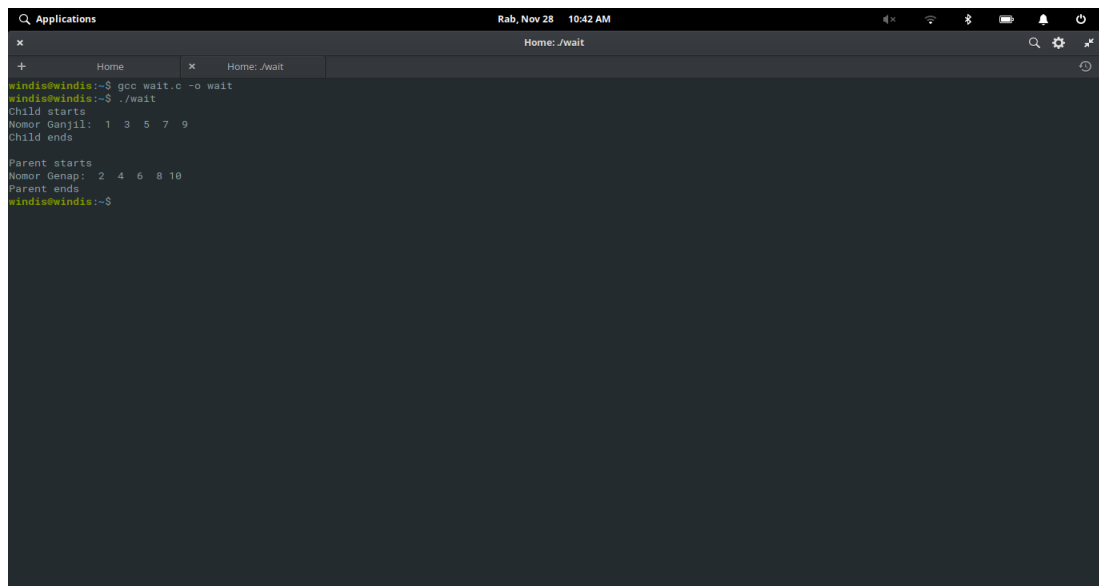
```
windis@windis:~$ nano wait.c
wait.c
[1/2]

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
int main(){
    int i,status;
    pid_t pid;
    pid = fork();

    if (pid < 0){
        printf("\nPembuatan proses gagal\n");
        exit(-1);
    }
    else if(pid > 0)
    {
        wait(NULL);
        printf ("\nParent starts\nNomor Genap:");
        for (i=2;i<=10;i+=2)
            printf ("%3d",i);
        printf("\nParent ends\n");
    }
    else if (pid == 0)
    {
        printf ("Child starts\nNomor Ganjil:");
        for (i=1;i<=10;i+=2)
            printf ("%3d",i);
        printf ("\nChild ends\n");
    }
}

[ Read 15 lines ]
Get Help  Write Out  Where Is  Cut Text  Justify  Cur Pos  Undo  Mark Text  To Bracket  Previous
Close  Read File  Replace  Uncut Text  To Spell  Go To Line  Redo  Copy Text  WhereIs Next  Next
```

4. Ketik “gcc wait.c -o wait” untuk mengecek apakah codingan sudah benar atau belum.
5. Ketik “./wait” untuk menampilkan outputnya

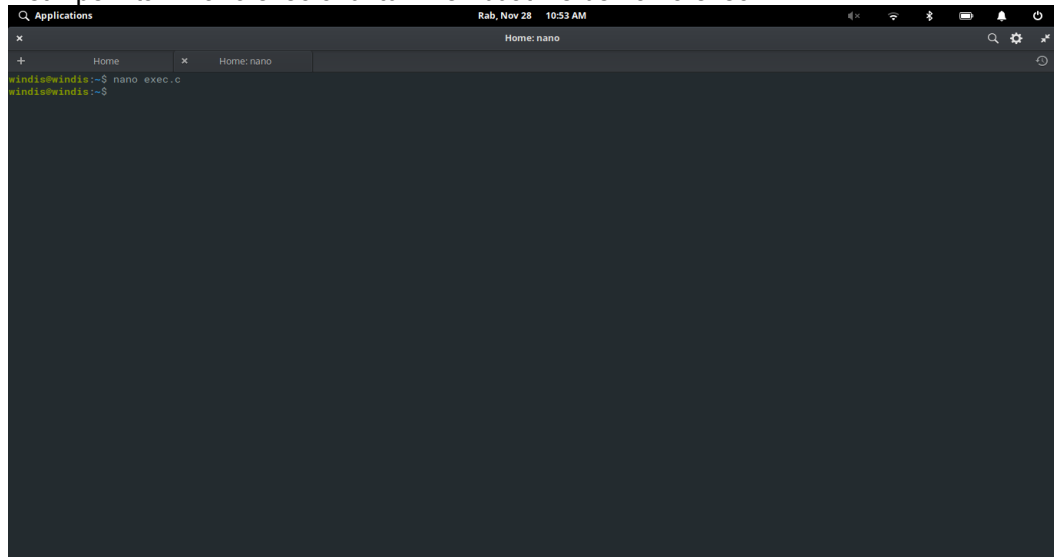


```
Applications Rab, Nov 28 10:42 AM
* Home * Home: ./wait
+ Home: ./wait
windiswindis:~$ gcc wait.c -o wait
windiswindis:~$ ./wait
Child starts
Nomor Ganjil: 1 3 5 7 9
Child ends

Parent starts
Nomor Genap: 2 4 6 8 10
Parent ends
windiswindis:~$
```

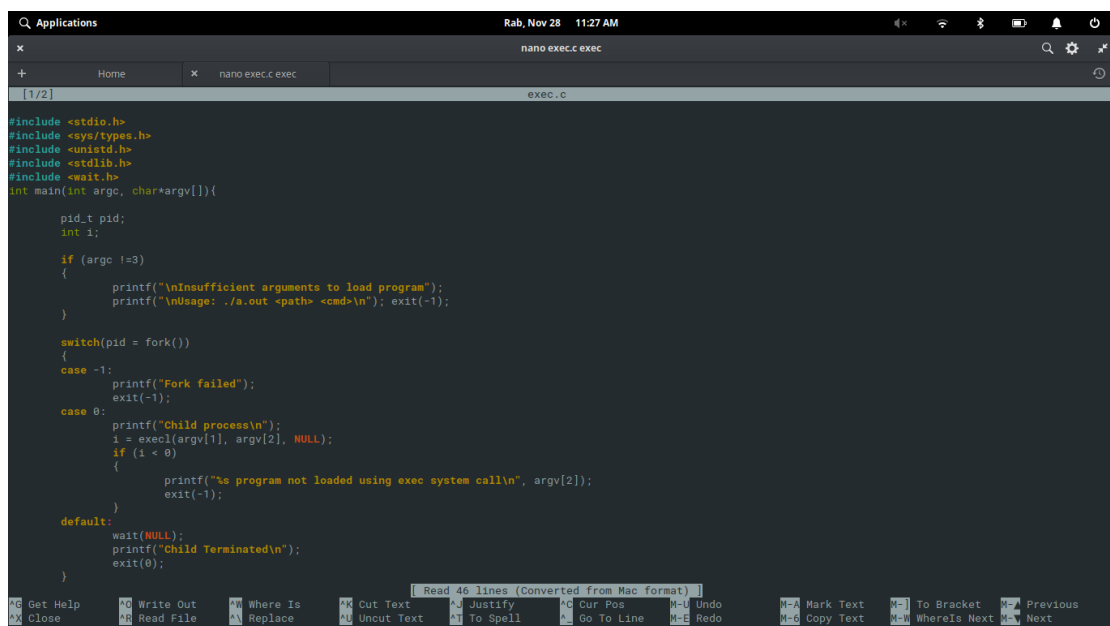
D. exec.c

1. Ketik perintah “nano exec.c” untuk membuat file bernama exec



The screenshot shows a terminal window with the prompt 'windis@windis:~\$'. The command 'nano exec.c' has been entered, and the prompt is now 'windis@windis:~\$' again, indicating the command was executed. The terminal window title is 'Home: nano'.

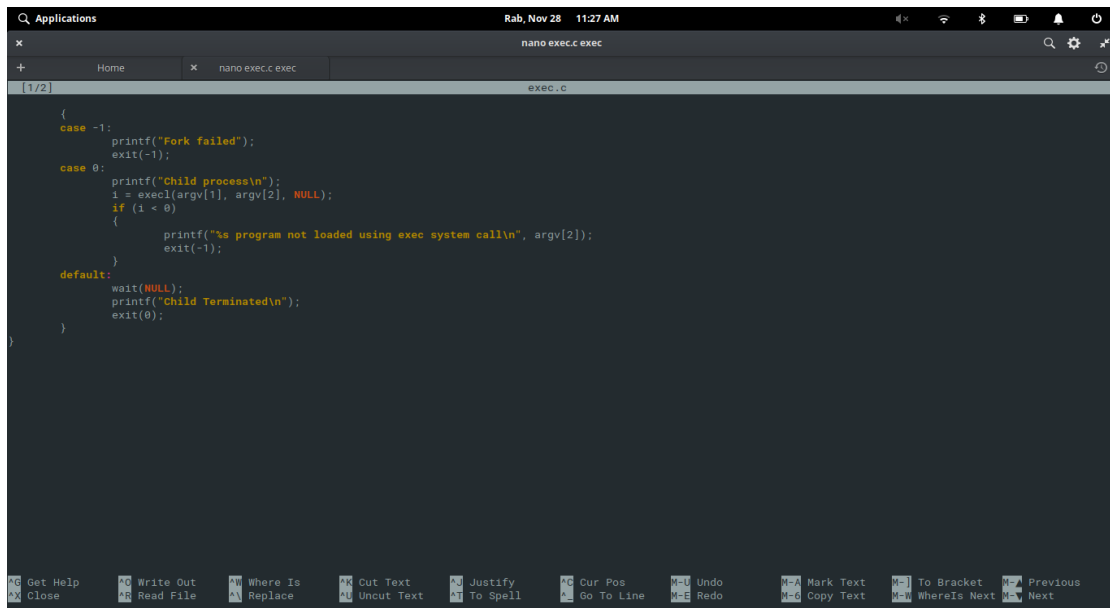
2. Lalu ketik perintah yang terdapat pada modul
3. Tekan ctrl+x untuk menyimpan file, lalu ketik “Y”, lalu enter



The screenshot shows the nano editor window with the file 'exec.c' open. The code is as follows:

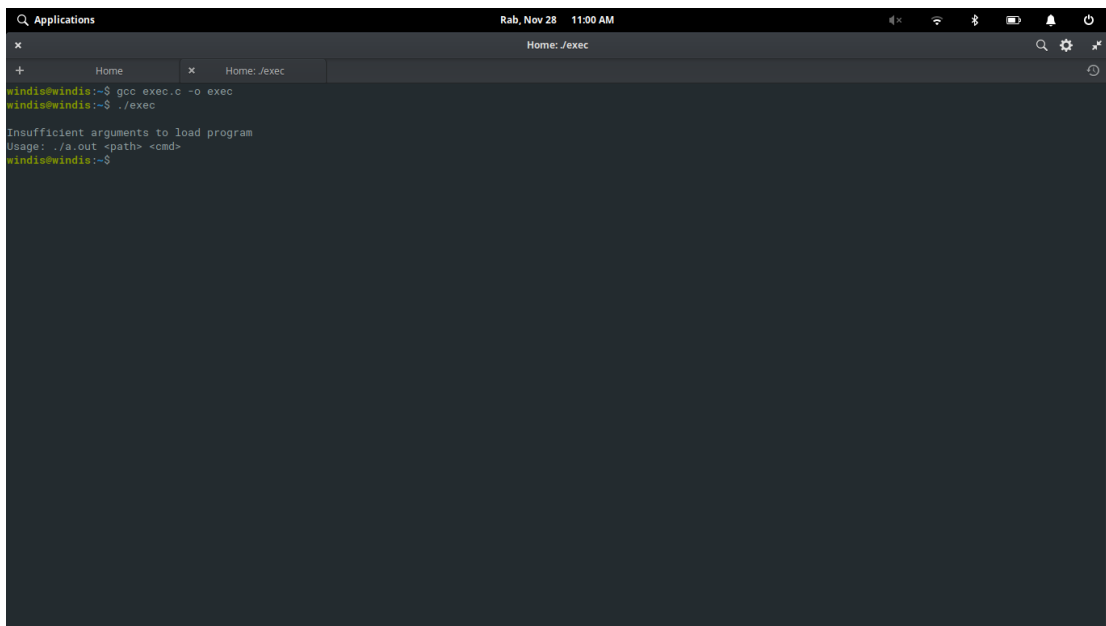
```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>
#include <wait.h>
int main(int argc, char*argv[]){
    pid_t pid;
    int i;
    if (argc !=3)
    {
        printf("\nInsufficient arguments to load program");
        printf("\nUsage: ./a.out <path> <cmd>\n"); exit(-1);
    }
    switch(pid = fork())
    {
        case -1:
            printf("Fork failed");
            exit(-1);
        case 0:
            printf("Child process\n");
            i = execl(argv[1], argv[2], NULL);
            if (i < 0)
            {
                printf("%s program not loaded using exec system call\n", argv[2]);
                exit(-1);
            }
        default:
            wait(NULL);
            printf("Child Terminated\n");
            exit(0);
    }
}
```

The bottom of the window shows a status bar with various keyboard shortcuts and a message: 'Read 46 lines (Converted from Mac format)'.



```
1/2 exec.c
{
    case -1:
        printf("Fork failed");
        exit(-1);
    case 0:
        printf("Child process\n");
        i = execl(argv[1], argv[2], NULL);
        if (i < 0)
        {
            printf("%s program not loaded using exec system call\n", argv[2]);
            exit(-1);
        }
    default:
        wait(NULL);
        printf("Child Terminated\n");
        exit(0);
}
```

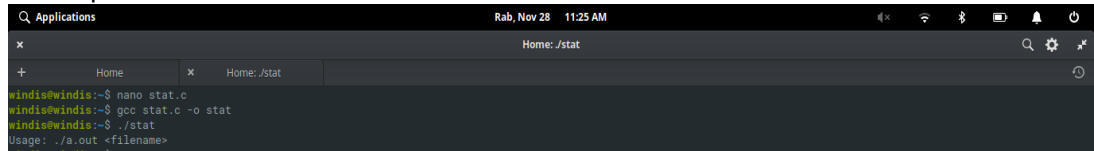
4. Ketik “gcc exec.c -o exec” untuk mengecek apakah codingan sudah benar atau belum.
5. Ketik “./exec” untuk menampilkan outputnya



```
windis@windis:~$ gcc exec.c -o exec
windis@windis:~$ ./exec
Insufficient arguments to load program
Usage: ./a.out <path> <cmd>
windis@windis:~$
```

E. Stat.c

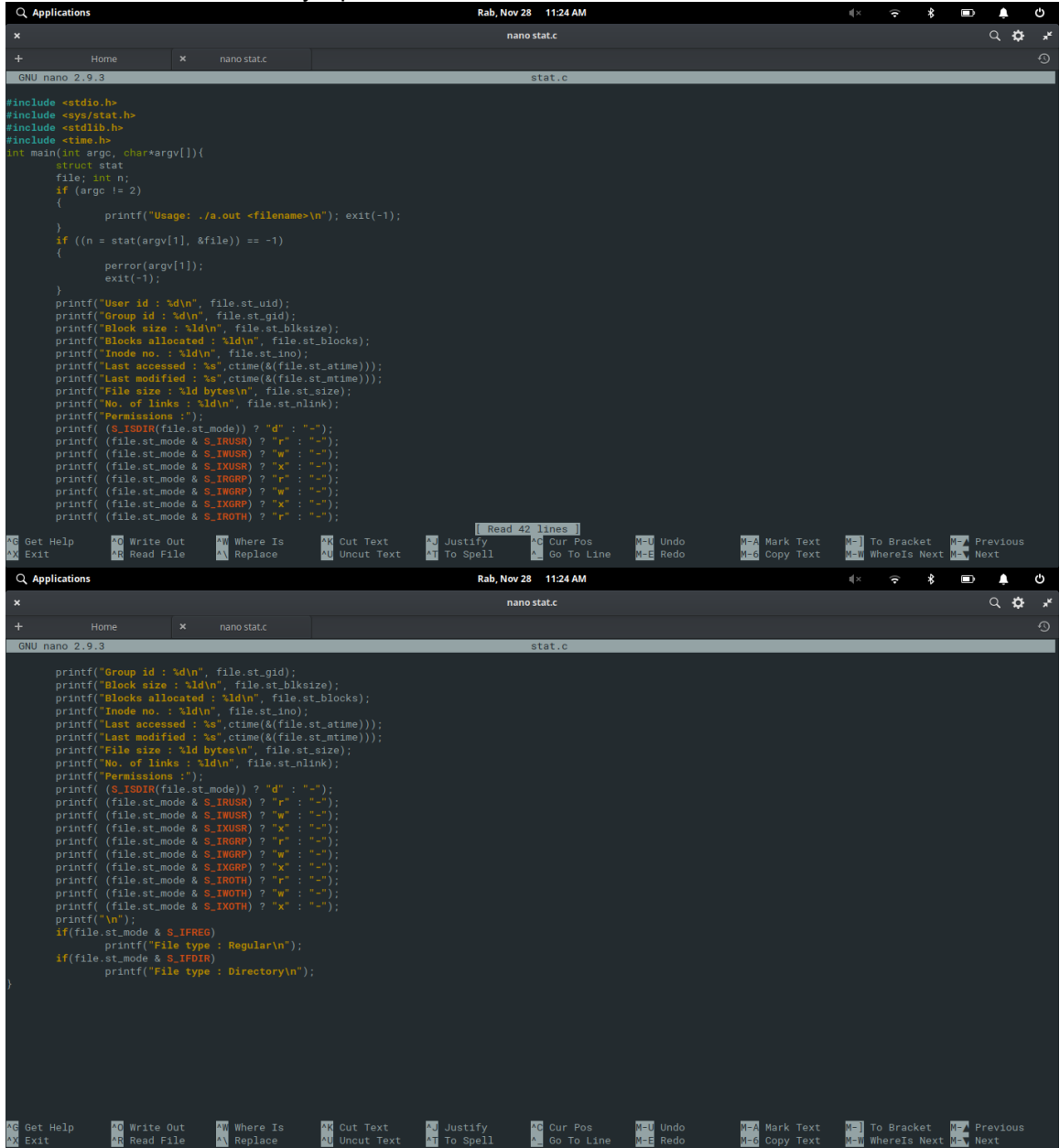
1. Ketik perintah “nano stat.c” untuk membuat file bernama stat



```
windis@windis:~$ nano stat.c
windis@windis:~$ gcc stat.c -o stat
windis@windis:~$ ./stat
Usage: ./a.out <filename>
```

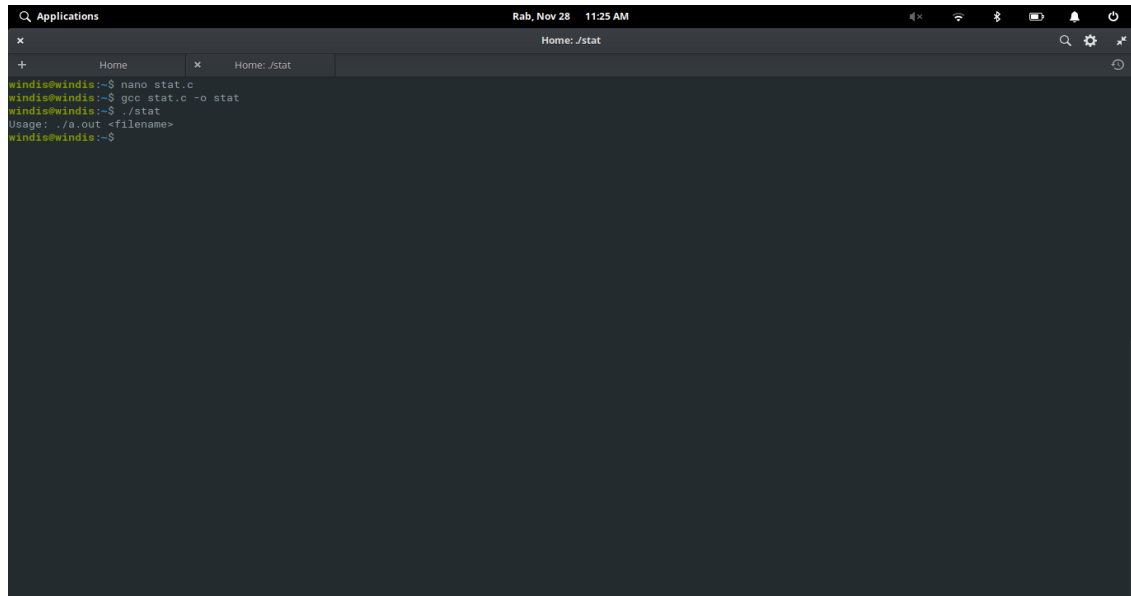
2. Lalu ketik perintah yang terdapat pada modul

3. Tekan ctrl+x untuk menyimpan file, lalu ketik “Y”, lalu enter



```
GNU nano 2.9.3 stat.c
#include <stdio.h>
#include <sys/stat.h>
#include <stdlib.h>
#include <time.h>
int main(int argc, char*argv[]){
    struct stat
    file; int n;
    if (argc != 2)
    {
        printf("Usage: ./a.out <filename>\n"); exit(-1);
    }
    if ((n = stat(argv[1], &file)) == -1)
    {
        perror(argv[1]);
        exit(-1);
    }
    printf("User id : %d\n", file.st_uid);
    printf("Group id : %d\n", file.st_gid);
    printf("Block size : %ld\n", file.st_blksize);
    printf("Blocks allocated : %ld\n", file.st_blocks);
    printf("Inode no. : %ld\n", file.st_ino);
    printf("Last accessed : %s", ctime(&(file.st_atime)));
    printf("Last modified : %s", ctime(&(file.st_mtime)));
    printf("File size : %ld bytes\n", file.st_size);
    printf("No. of links : %ld\n", file.st_nlink);
    printf("Permissions :");
    printf( (S_ISDIR(file.st_mode)) ? "d" : "-");
    printf( (file.st_mode & S_IRUSR) ? "r" : "-");
    printf( (file.st_mode & S_IWUSR) ? "w" : "-");
    printf( (file.st_mode & S_IXUSR) ? "x" : "-");
    printf( (file.st_mode & S_IRGRP) ? "r" : "-");
    printf( (file.st_mode & S_IWGRP) ? "w" : "-");
    printf( (file.st_mode & S_IXGRP) ? "x" : "-");
    printf( (file.st_mode & S_IROTH) ? "r" : "-");
    printf( (file.st_mode & S_IWOTH) ? "w" : "-");
    printf( (file.st_mode & S_IXOTH) ? "x" : "-");
    printf("\n");
    if(file.st_mode & S_IFREG)
        printf("File type : Regular\n");
    if(file.st_mode & S_IFDIR)
        printf("File type : Directory\n");
}
```

4. Ketik “gcc stat.c -o stat” untuk mengecek apakah codingan sudah benar atau belum.
5. Ketik “./stat” untuk menampilkan outputnya

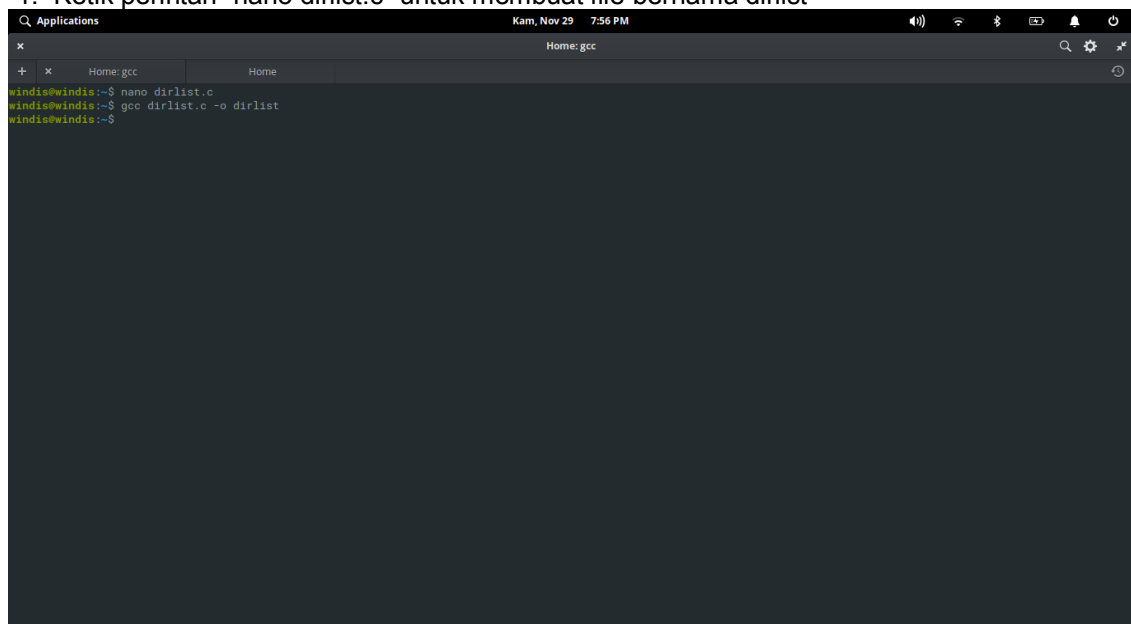


A terminal window titled "Applications" with a dark background. The window shows the following commands and output:

```
windis@windis:~$ nano stat.c
windis@windis:~$ gcc stat.c -o stat
windis@windis:~$ ./stat
Usage: ./a.out <filename>
windis@windis:~$
```

F. Dirlist.c

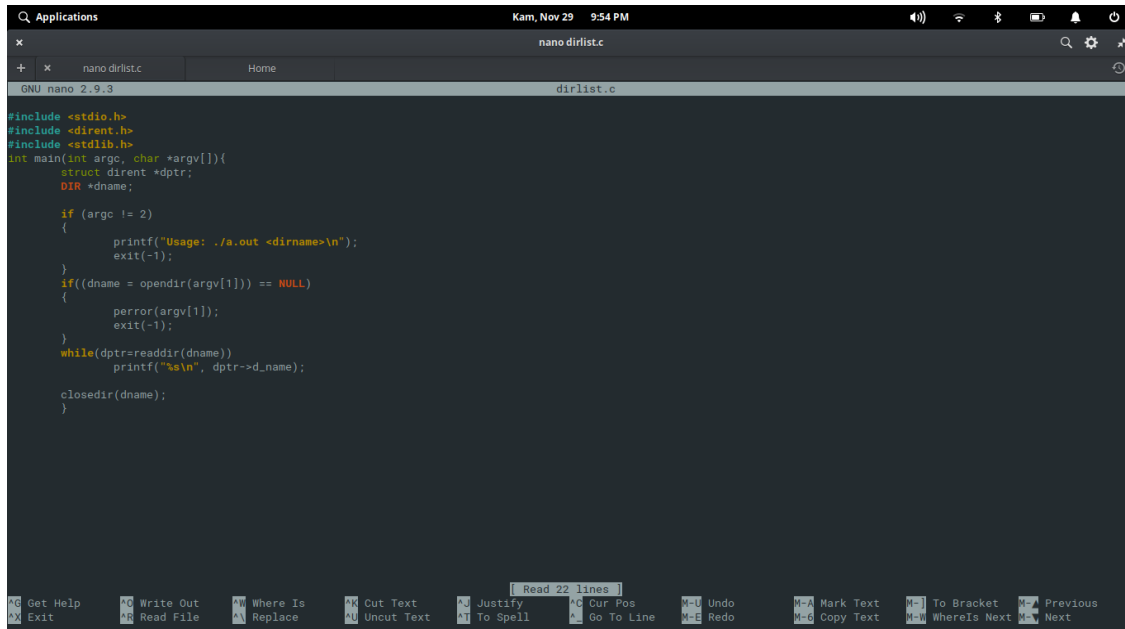
1. Ketik perintah “nano dirlist.c” untuk membuat file bernama dirlist



A terminal window titled "Applications" with a dark background. The window shows the following commands and output:

```
windis@windis:~$ nano dirlist.c
windis@windis:~$ gcc dirlist.c -o dirlist
windis@windis:~$
```


2. Lalu ketik perintah yang terdapat pada modul
3. Tekan ctrl+x untuk menyimpan file, lalu ketik “Y”, lalu enter

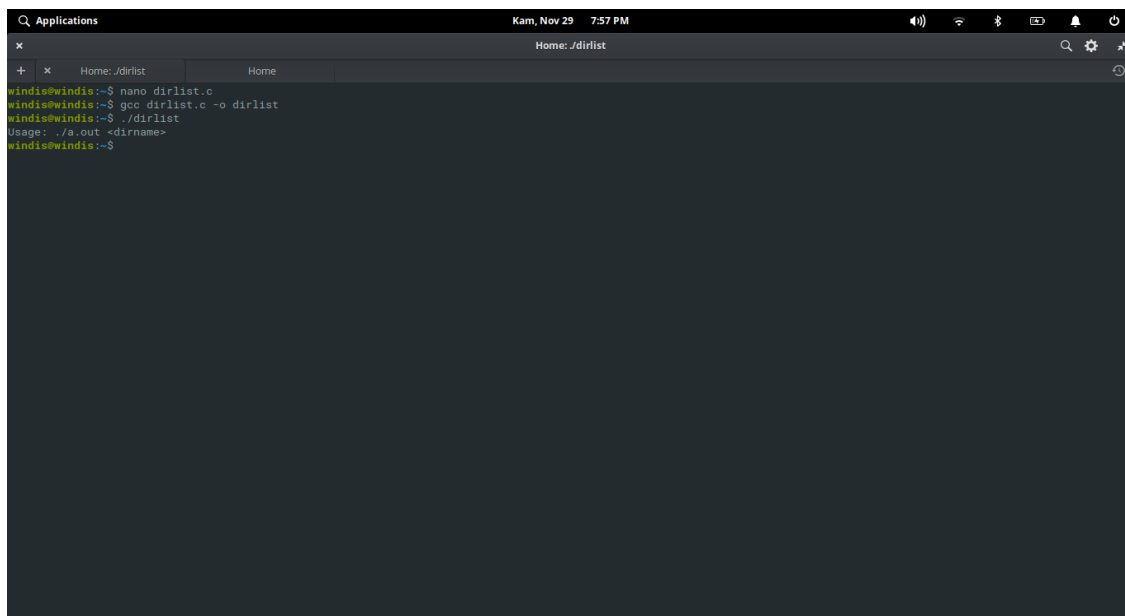


```
GNU nano 2.9.3 dirlist.c
#include <stdio.h>
#include <dirent.h>
#include <stdlib.h>
int main(int argc, char *argv[]){
    struct dirent *dptr;
    DIR *dname;

    if (argc != 2)
    {
        printf("Usage: ./a.out <dirname>\n");
        exit(-1);
    }
    if((dname = opendir(argv[1])) == NULL)
    {
        perror(argv[1]);
        exit(-1);
    }
    while(dptr=readdir(dname))
        printf("%s\n", dptr->d_name);

    closedir(dname);
}
```

4. Ketik “gcc dirlist.c -o dirlist” untuk mengecek apakah codingan sudah benar atau belum.
5. Ketik “./dirlist” untuk menampilkan outputnya



```
windis@windis:~$ nano dirlist.c
windis@windis:~$ gcc dirlist.c -o dirlist
windis@windis:~$ ./dirlist
Usage: ./a.out <dirname>
windis@windis:~$
```