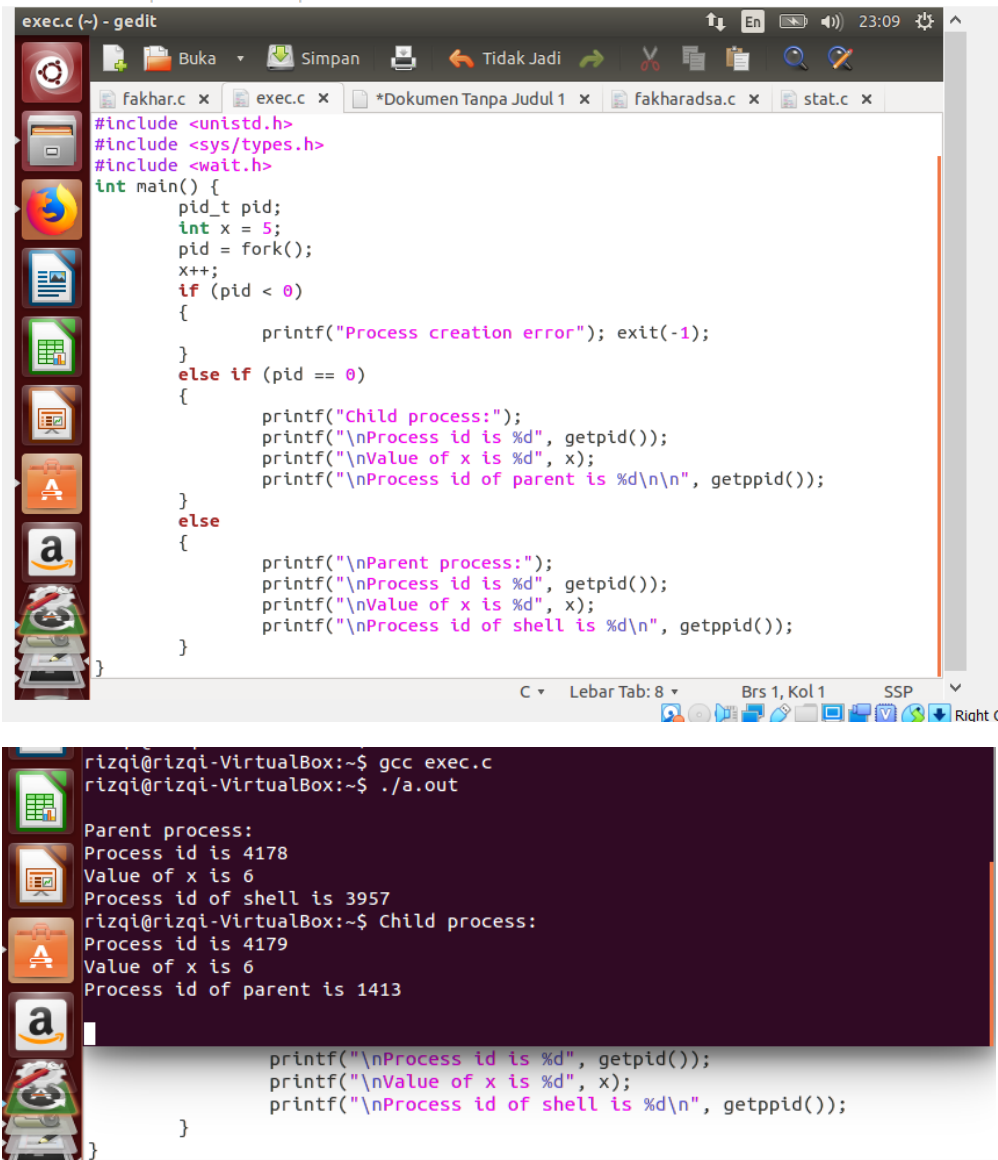


Muhammad Rizqi Erdyansyah

L200180171

3. exec.c



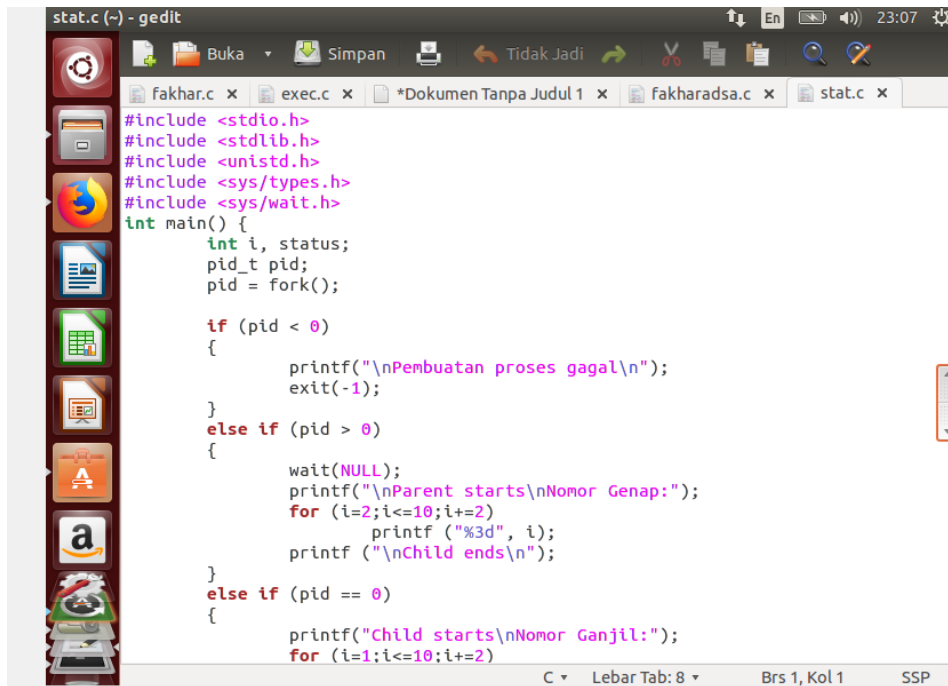
The image shows a C program named 'exec.c' and its execution output. The program is written in a text editor and then compiled and run in a terminal.

```
#include <unistd.h>
#include <sys/types.h>
#include <wait.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", getppid());
    }
    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());
    }
}
```

The terminal output shows the execution of the program. The parent process prints its ID (4178), the value of x (6), and the ID of the shell (3957). The child process prints its ID (4179), the value of x (6), and the ID of the parent (1413).

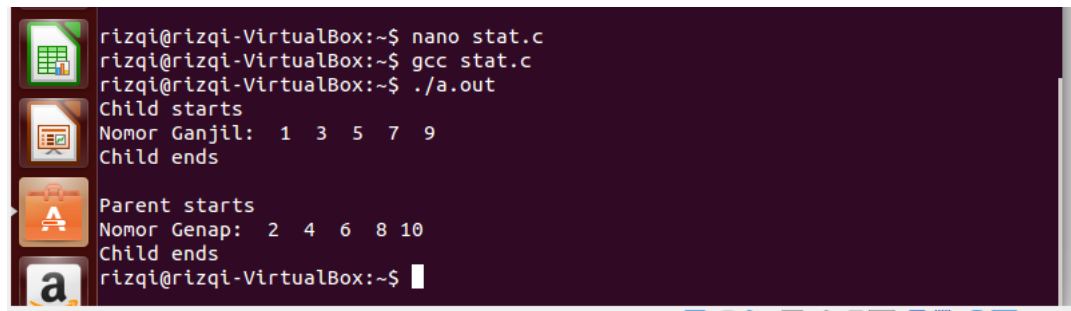
```
rizqi@rizqi-VirtualBox:~$ gcc exec.c
rizqi@rizqi-VirtualBox:~$ ./a.out
Parent process:
Process id is 4178
Value of x is 6
Process id of shell is 3957
rizqi@rizqi-VirtualBox:~$ Child process:
Process id is 4179
Value of x is 6
Process id of parent is 1413
```

4. stat.c



```
stat.c (~) - gedit
#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();

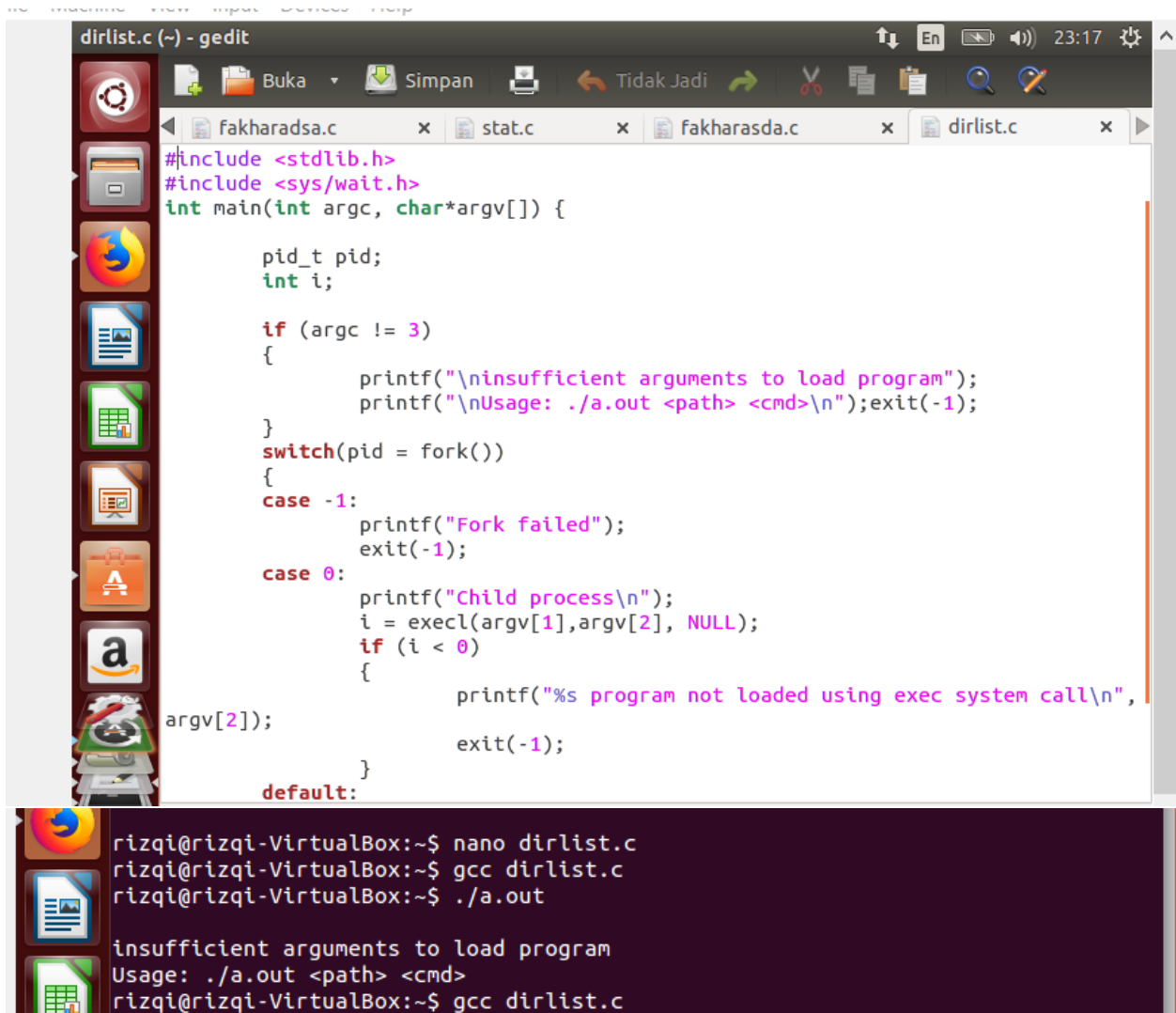
    if (pid < 0)
    {
        printf("\nPembuatan proses gagal\n");
        exit(-1);
    }
    else if (pid > 0)
    {
        wait(NULL);
        printf("\nParent starts\nNomor Genap:");
        for (i=2;i<=10;i+=2)
            printf ("%3d", i);
        printf ("\nChild ends\n");
    }
    else if (pid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



```
rizqi@rizqi-VirtualBox:~$ nano stat.c
rizqi@rizqi-VirtualBox:~$ gcc stat.c
rizqi@rizqi-VirtualBox:~$ ./a.out
Child starts
Nomor Ganjil:  1  3  5  7  9
Child ends

Parent starts
Nomor Genap:   2  4  6  8 10
Child ends
rizqi@rizqi-VirtualBox:~$
```

5. dirlist.c



The image shows a Linux desktop environment. The top part of the image displays a gedit text editor window titled 'dirlist.c (~) - gedit'. The editor contains the following C code:

```
#include <stdlib.h>
#include <sys/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:
    }
```

The bottom part of the image shows a terminal window with the following commands and output:

```
rizqi@rizqi-VirtualBox:~$ nano dirlist.c
rizqi@rizqi-VirtualBox:~$ gcc dirlist.c
rizqi@rizqi-VirtualBox:~$ ./a.out

insufficient arguments to load program
Usage: ./a.out <path> <cmd>
rizqi@rizqi-VirtualBox:~$ gcc dirlist.c
```