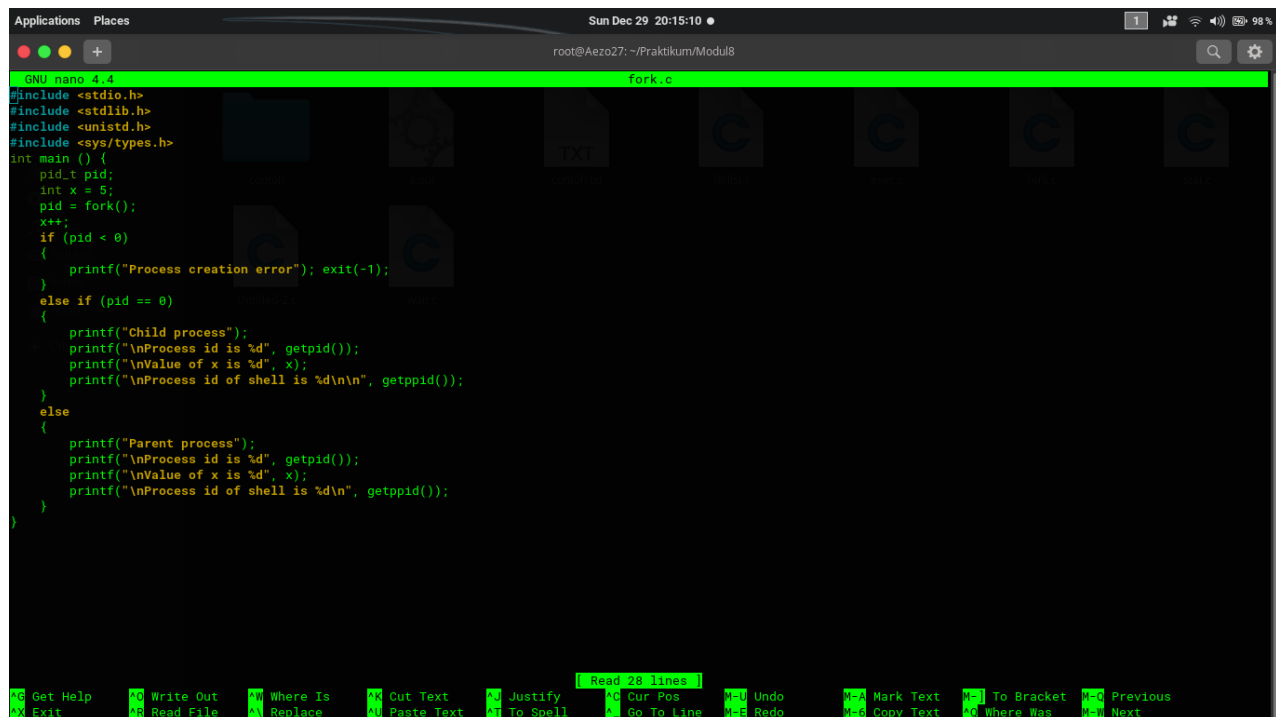


**Nama : Diah Fitri Ramadhani**

**NIM : L200180106**

## MODUL 8 (KEGIATAN 1 – 3)

Fork.c



The screenshot shows a terminal window with the nano text editor open. The editor is displaying the source code for a C program named 'fork.c'. The code uses the 'fork()' system call to create a child process. It includes headers for stdio, stdlib, unistd, and sys/types. The main function declares a pid\_t variable, initializes x to 5, and calls fork(). It then uses conditional logic to handle errors, child process execution, and parent process execution, printing process IDs and the value of x. The terminal window has a dark theme and a status bar at the bottom showing various keyboard shortcuts.

```
GNU nano 4.4 fork.c
#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 shell is %d\n", getppid());
    }
    else
    {
        printf("Parent process");
        printf("\nProcess id is %d", getpid());
        printf("\nValue of x is %d", x);
        printf("\nProcess id of shell is %d\n", getppid());
    }
}
```

Read 28 lines

Get Help Write Out Where Is Cut Text Justify Cur Pos Undo Mark Text To Bracket Previous  
Exit Read File Replace Paste Text To Spell Go To Line Redo Copy Text Where Was Next

```
Applications  Places  Sun Dec 29 20:16:44 ● 1 98%
root@Aezo27: ~/Praktikum/Modul8

root@Aezo27:~/Praktikum/Modul8# nano fork.c
root@Aezo27:~/Praktikum/Modul8# gcc fork.c
root@Aezo27:~/Praktikum/Modul8# ./a.out

Parent process
Process id is 3466
Value of x is 6
Process id of shell is 3449
Child process
Process id is 3467
Value of x is 6
Process id of shell is 3466

root@Aezo27:~/Praktikum/Modul8#
```

## Wait.c

```
Applications  Places  Sun Dec 29 20:17:36 ● 1 98%
root@Aezo27: ~/Praktikum/Modul8

GNU nano 4.4 wait.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.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 ("\nChild starts\nNomor Ganjil:");
        for (i = 1; i < 10; i+=2)
        {
            printf("%3d",i);
        }
        printf("\nChild ends\n");
    }
}

[ Read 35 lines ]
⌘ Get Help  ⌘ Write Out  ⌘ Where Is  ⌘ Cut Text  ⌘ Justify  ⌘ Cur Pos  ⌘ Undo  ⌘ Mark Text  ⌘ To Bracket  ⌘ Previous
⌘ Exit      ⌘ Read File  ⌘ Replace   ⌘ Paste Text ⌘ To Spell  ⌘ Go To Line ⌘ Redo   ⌘ Copy Text  ⌘ Where Was  ⌘ Next
```

```
Applications  Places  Sun Dec 29 20:17:51
root@Aezo27: ~/Praktikum/Modul8

root@Aezo27:~/Praktikum/Modul8# nano wait.c
root@Aezo27:~/Praktikum/Modul8# gcc wait.c
wait.c: In function 'main':
wait.c:17:9: warning: implicit declaration of function 'wait'; did you mean 'main'? [-Wimplicit-function-declaration]
    wait(NULL);
    ^~~~~
    main
root@Aezo27:~/Praktikum/Modul8# ./a.out

Child starts
Nomor Ganjil: 1 3 5 7 9
Child ends

Parent starts
Nomor Genap: 2 4 6 8
Parent ends
root@Aezo27:~/Praktikum/Modul8#
```

## Exec.c

```
Applications  Places  Sun Dec 29 20:10:13
root@Aezo27: ~/Praktikum/Modul8

GNU nano 4.4  exec.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.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], 0);
            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);
    }
}

[ Read 35 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  ^U Undo  ^M Mark Text  ^C] To Bracket  ^_ Previous
^X Exit      ^R Read File  ^N Replace   ^P Paste Text ^T To Spell  ^L Go To Line ^E Redo    ^H Copy Text  ^C Where Was ^_ Next
```

```
Applications  Places  Sun Dec 29 20:14:32  1  98%
root@Aezo27: ~/Praktikum/Modul8

root@Aezo27:~/Praktikum/Modul8# nano exec.c
root@Aezo27:~/Praktikum/Modul8# gcc exec.c
exec.c: In function 'main':
exec.c:31:9: warning: implicit declaration of function 'wait'; did you mean 'main'? [-Wimplicit-function-declaration]
     wait(NULL);
     ^~~~~
     main
root@Aezo27:~/Praktikum/Modul8# ./a.out /bin/ls -ls
Child process
a.out  contoh  contoh.txt  dirlist.c  exec.c  fork.c  stat.c  Untitled-2.c  wait.c
Child Terminated
root@Aezo27:~/Praktikum/Modul8#
```