**LEMBAR KERJA 8**

**PRAKTIKKUM SISTEM OPERASI**

|  |  |
| --- | --- |
| **Nama : Jesika Anjarwati**  **NIM : L200210258**  **Kelas : E**  **Dosen pengampu : Heru Setiya Nugraha, S.T, M.Kom** | **Tanggal praktikum : Selasa, 6 Desember 2022**  **Nilai :** |

***LangkahKerja***

Gunakan aplikasi ‘nano’ atau ‘vi’ atau teks editor yang lain untuk mengedit kodeprogramberikut, Selanjutnya untuk melakukan kompilasi dapat dilakukan dengan perintahberikut:

$gcc‘nama\_file.c’

Jika tidak ada kesalahan maka akan dihasilkan sebuah program bernama ‘a.out’, danuntukmenjalankanprogram tersebut dapatdilakukan dengancaraberikut:

$ ./a.out

Jika pada PC anda tidak tersedia compiler ‘gcc’ dapat digunakan fasilitas onlinecompileryangdisediakanoleh linkberikut:

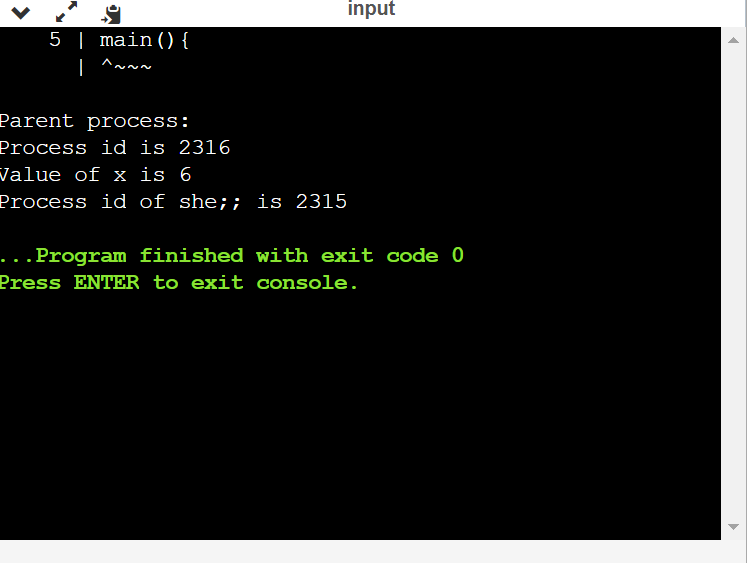
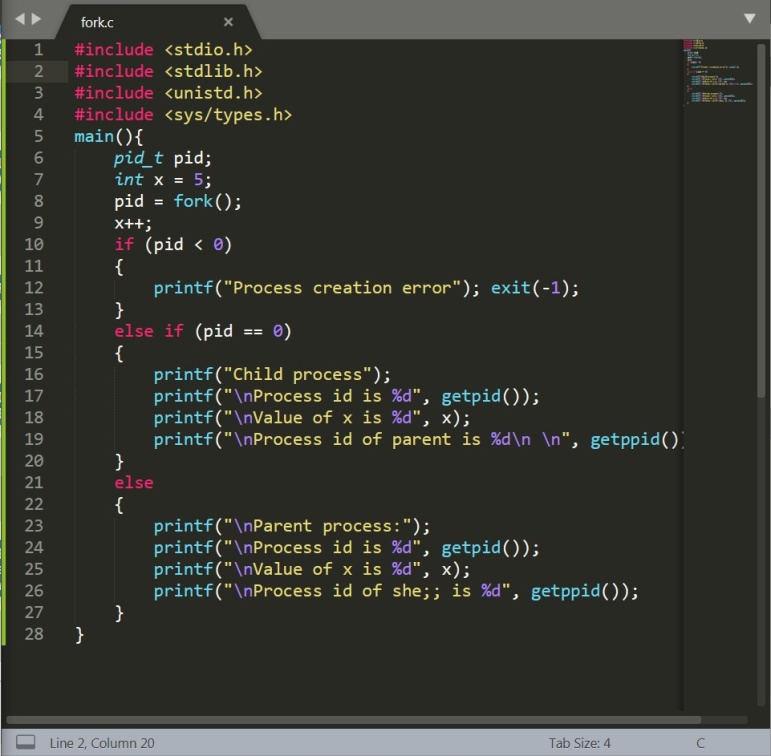
<http://www.tutorialspoint.com/compile_c_online.php>

# PRAKTIKUM1(Fork.c)

1. Membuatsebuah‘childprocess’(prosesbaru)denganmenggunakansystemcall

‘fork’.Mambuatprogramdenganalgoritmasebagaiberikut:

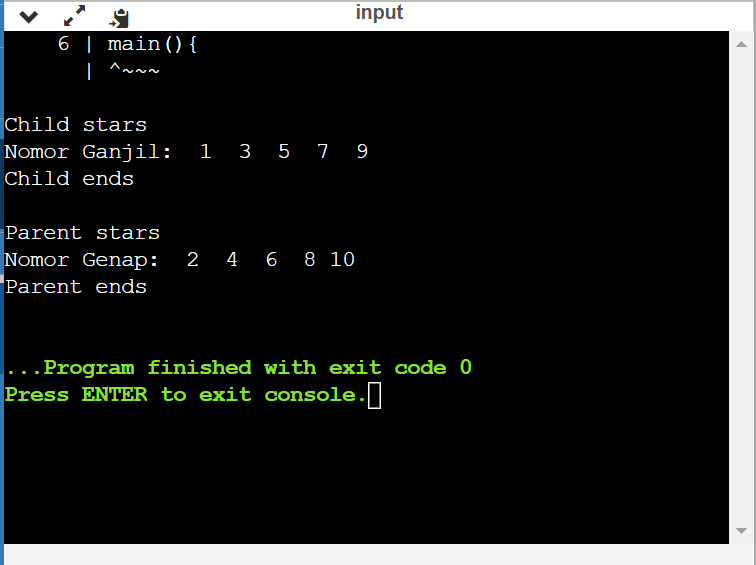
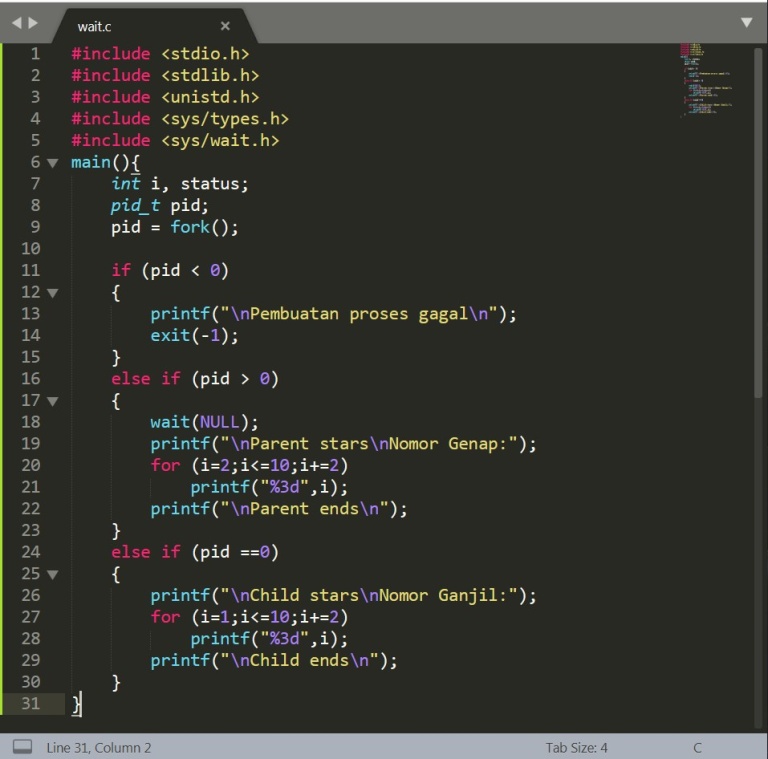
(contohprogramdiberikanpadabagian berikutnya).

* 1. Deklarasi sebuah variabel x yang akan diakses bersama antara child proses dan parentproses.
  2. Membuatsebuahchildprosesmenggunakansystemcallfork.
  3. Jika return value bernilai -1, tampilkan teks ‘Pembuatan proses GAGAL’, dilanjutkandengankeluarprogramdenganperintahsystemcall‘exit’.
  4. Jikareturnvaluesamadengan0(NOL),Tampilkanteks‘Child Process’,tampilkanID proses dari child proses menggunakan perintah system call ‘getpid’,tampilkannilaix,dantampilkanIDprosesparentdenganperintahsystemcall‘getppid’.
  5. Untuknilaireturnvalueyanglainnya,tampilkanteks‘Parentprocess’,tampilkanID dari parent proses menggunakan perintah system call getpid, tampilkan nilai x, dantampilkanID dariproses shell menggunakanperintah system callgetppid.
  6. Stop
     + Kode Program dan Outputnya :

# PRAKTIKUM2(Wait.c)

1. Menghentikansementara(block)prosesparentsampaidenganproseschildselesai,menggunkaanperintahsystemcall‘wait’.

Membuatprogramdenganalgoritmasebagaiberikut,contohprogramdiberikanpadabagianberikutnya..

* 1. Membuatsebuahchildprosesmenggunakansytemcall‘fork’.
  2. Jika return value bernilai -1, selanjutnya tampilkan teks ‘pembuatan proses gagal’,dankelaurprogramdenganmenggunakanperintahsystemcall‘exit’.
  3. Jika return value berupa angka positif ( > 0),‘pause’ hentikan sementara‘parent’prosestunggu sampai child proses berakhir dengan menggunakan perintah system call‘wait’.Tampilkanteks‘Parentstarts’,selanjutnyatampilkannomorgenapmulaidari0s/d10,terakhirtampilkanteks‘Parentend’.
  4. Jikareturnvaluebernilai0(NOL),tampilkanteks‘Childstart’,tampilkannomorganjilmulaidari0s/d10,selanjutnyatampilkanteks‘childends’
  5. Stop
     + Kode Program dan Outputnya :

# PRAKTIKUM3(Exec.c)

1. Loadingprogramyangdapatdieksekusidalamsebuah‘child’prosesmenggunakanperintahsystemcall‘exec’.Membuatprogramdenganalgoritmasebagaiberikut :

(contohprogramdiberikan padabagianberikutnya).

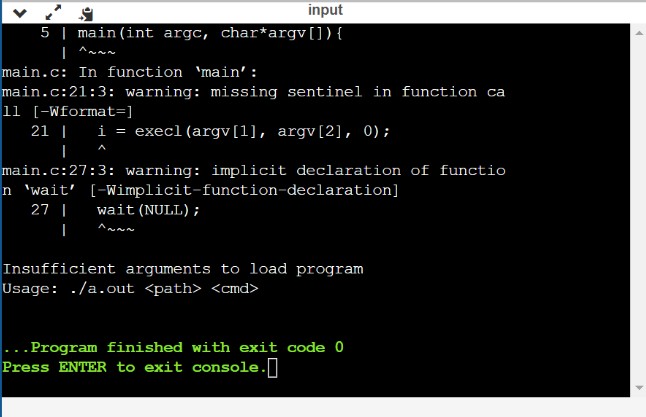
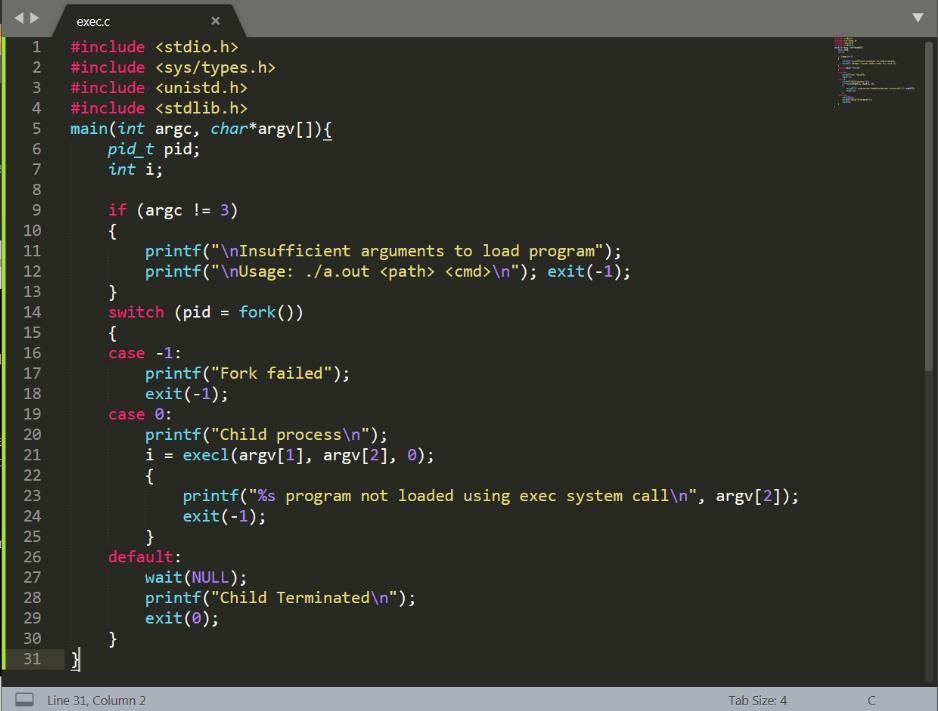
* 1. Jikaterdapat 3argumendalamcommand-lineberhenti (stop).
  2. Membuatchildprosesdenganperintahsystemcall‘fork’
  3. Jikareturnvalueadalah-1,selanjuntyatampilkanteks‘PembuatanprosesGagal’,dankeluarprogram dengan perintah system call exit.
  4. Jika return value >0 (positif), selanjutnya hentikan parent-proses sementara hinggachild-proses berakhir dengan menggunakan perintah system call wait. Tampilkanteks‘Childberakhir’,danhentikanparent-proses.
  5. Jika return value sama dengan 0 (NOL), selanjutnya tampilkan teks ‘Child starts’,load program dari lokasi yang diberikan dalam ‘path’ ke dalam child-proses,menggunakan perintah system call ‘exec’. Jika return value dari perintah ‘exec’adalahbilangannegatif,tampilkanerroryangterjadidanstop.Hentikanchild-proses.
  6. Stop

Contohcaramengkompilasi danmenjalankan program

$ gcc exec.c

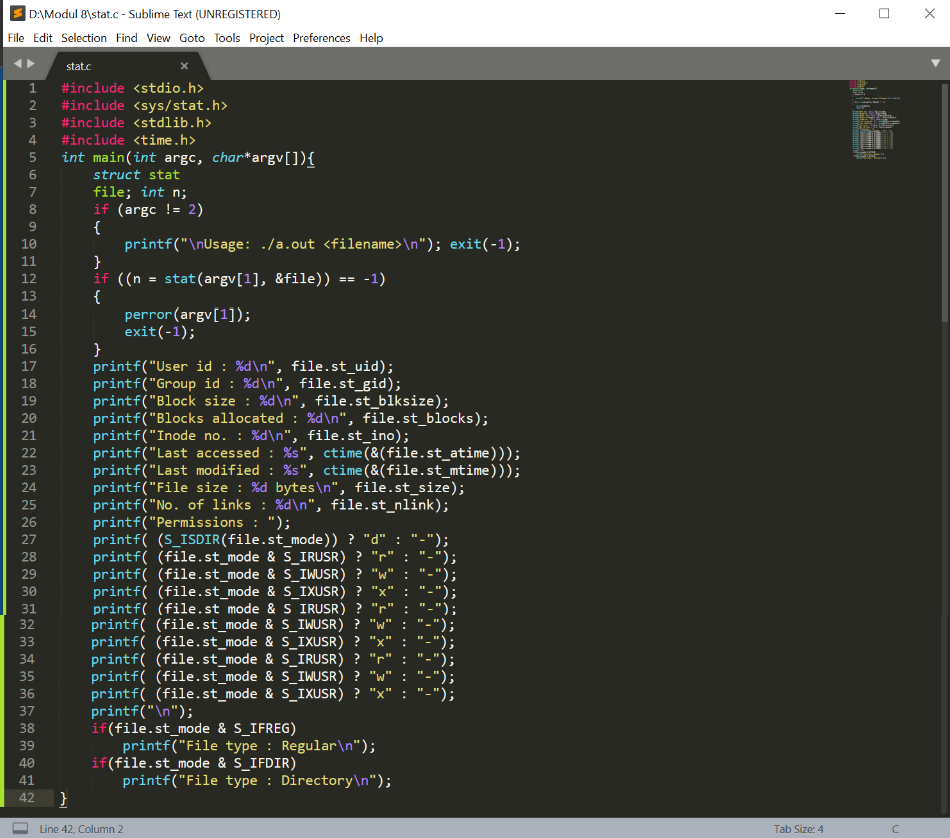
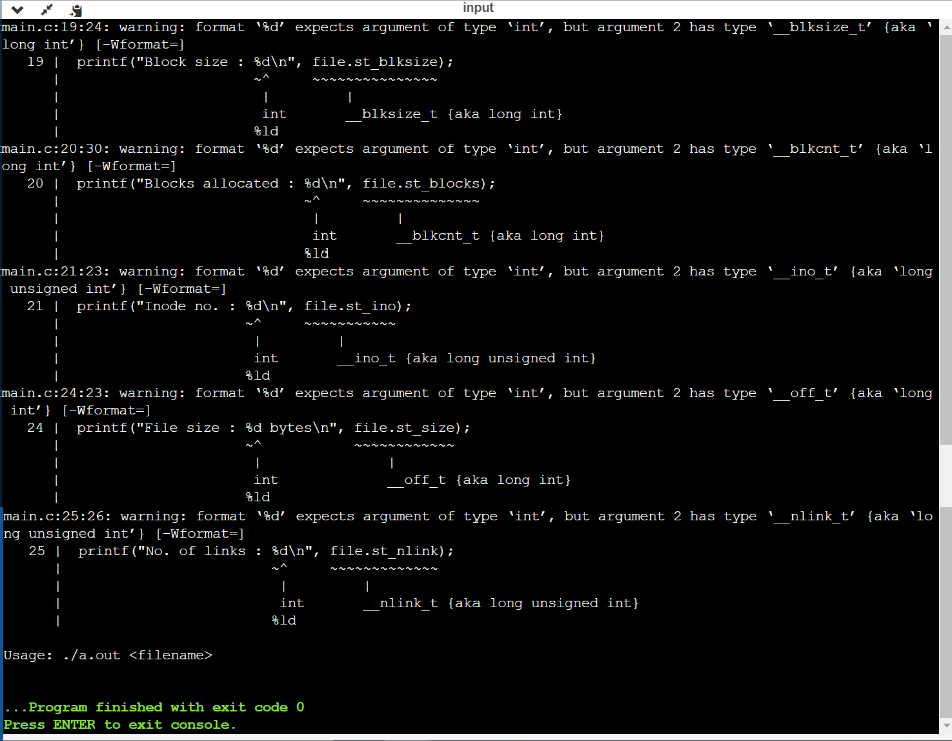
$ ./a.out /bin/ls ls

* + - Kode Program dan Outputnya :



# PRAKTIKUM4(Stat.c)

1. Menampilkanstatusfilemenggunakanperintahsystemcall‘stat’Membuatprogramdenganalgoritma sebagaiberikut(contoh codeadadibagianberikutnya):
   1. Gunakan‘namafile’yangdiberikanmelaluiargumentdalamperintahcommand-line.
   2. Jika‘nama-file’tidakadamakastopdisini(keluarprogram)
   3. Panggilsystemcall‘stat’pada‘nama-file’tersebutyangakanmengembalikansebuahstruktur.
   4. Tampilkaninformasimengenaist\_uid,st\_blksize,st\_block,st\_size,st\_nlink,etc.
   5. Ubahwaktu dalamst\_time, st\_mtimedengan menggunakan fungsictime.
   6. Bandingkan st\_modedengan konstantamodesepertiS\_IRUSR,S\_IWGRP, S\_IXOTHdantampilkaninformasimengenai‘file-permissions’.
   7. Stop
      * Kode Program dan Outputnya :



# PRAKTIKUM5(Dirlist.c)

1. Menampilkanisidirektorimenggunakanperintahsystemcall‘readdir’

Membuatprogramdenganalgoritmasebagaiberikut(contohcodeadadibagianberikutnya):

* 1. Gunakan‘nama-direktori’yangdiberikansebagaiargumenpadacommand-line.
  2. Jikadirektori tidakditemukan stop,keluarprogram
  3. Bukadirektorimenggunakanperintahsystemcall‘opendir’yangakanmenghasilkansebuahstruktur.
  4. Bacadirektorimenggunakanperintahsystemcall‘readdir’yangjugaakanmenghasilkanstruktur data.
  5. Tampilkand\_name(namadirekrori)
  6. Akhiripembacaadirektoridenganperintahsystemcall‘closedir’.
  7. Stop
     + Kode Program dan Outputnya :

