

LAPORAN PRAKTIKUM SISTEM OPERASI

PRAKTIKUM 04 – SHELL

Nama Kelompok :

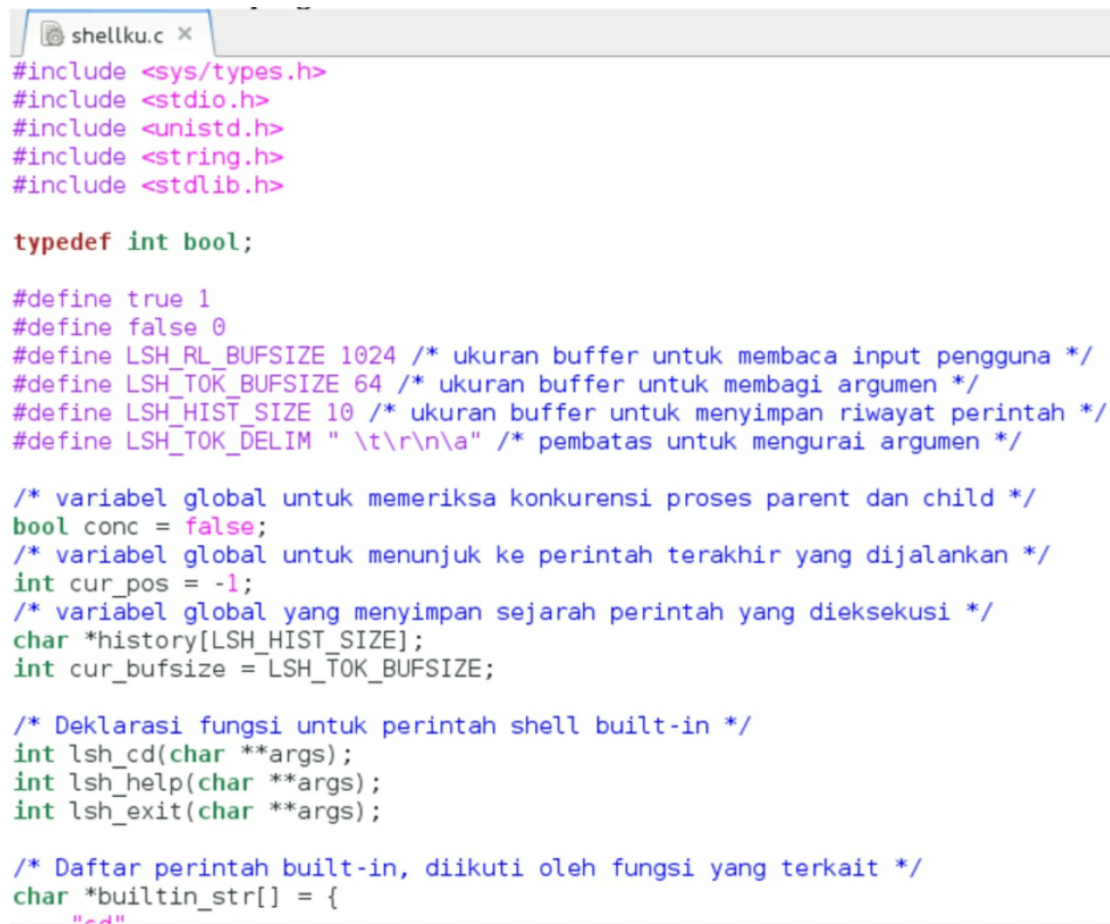
Kholis Nabil Abdullah (17312101)

Candra Yuda (17312005)

Penjelasan Shell

Shell adalah salah satu jenis program asli dari sistem operasi (program yang terpisah dari inti sistem operasi) yang menyediakan komunikasi langsung antara pengguna dan sistem operasi.

Screenshoot kode program



```
shellku.c x
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <stdlib.h>

typedef int bool;

#define true 1
#define false 0
#define LSH_RL_BUFSIZE 1024 /* ukuran buffer untuk membaca input pengguna */
#define LSH_TOK_BUFSIZE 64 /* ukuran buffer untuk membagi argumen */
#define LSH_HIST_SIZE 10 /* ukuran buffer untuk menyimpan riwayat perintah */
#define LSH_TOK_DELIM " \t\r\n\a" /* pembatas untuk mengurai argumen */

/* variabel global untuk memeriksa konkurensi proses parent dan child */
bool conc = false;
/* variabel global untuk menunjuk ke perintah terakhir yang dijalankan */
int cur_pos = -1;
/* variabel global yang menyimpan sejarah perintah yang dieksekusi */
char *history[LSH_HIST_SIZE];
int cur_bufsize = LSH_TOK_BUFSIZE;

/* Deklarasi fungsi untuk perintah shell built-in */
int lsh_cd(char **args);
int lsh_help(char **args);
int lsh_exit(char **args);

/* Daftar perintah built-in, diikuti oleh fungsi yang terkait */
char *builtin_str[] = {
    "cd"
```

```
shellku.c x
/* Daftar perintah built-in, diikuti oleh fungsi yang terkait */
char *builtin_str[] = {
    "cd",
    "help",
    "exit"
};

int (*builtin_func[]) (char **) = {
    &lsh_cd,
    &lsh_help,
    &lsh_exit
};

int lsh_num_builtins(){
    return sizeof(builtin_str) / sizeof(char *);
}

/* Implementasi fungsi built-in */
int lsh_cd(char **args)
{
    if(args[1] == NULL){
        fprintf(stderr, "lsh: expected argument to \"cd\"\n");
    }else{
        if(chdir(args[1]) != 0){
            perror("lsh");
        }
    }

    return 1;
}
```

shellku.c ×

```
int lsh_help(char **args)
{
    int i;
    printf(" ===== \n");
    printf(" Kholis nabil (17312101) \n Chandra yuda (17312005)\n");
    printf(" ===== \n");
    printf("Ketik nama dan argumen program, dan tekan enter.\n");
    printf("Tambahkan \"%s\" setelah argumen untuk konkurensi antara proses parent dan child.\n");
    printf("Berikut ini yang dibangun:\n");

    for(i = 0; i < lsh_num_builtins(); i++){
        printf(" %s\n", builtin_str[i]);
    }

    printf("Gunakan perintah man untuk informasi tentang program lain.\n");
    return 1;
}

int lsh_exit(char **args)
{
    return 0;
}

/* Luncurkan program */
int lsh_launch(char **args)
{
    pid_t pid, wpid;
    int status;

    pid = fork();|
```

```
root@kholis:~/Desktop

File Edit View Search Terminal Help

root@kholis:~/ cd Dekstop/
root@kholis:~/ cd Dekstop# gcc -o myshell shellku.c
root@kholis:~/ cd Dekstop# ./myshell
>help
=====
Kholis nabil <17312101>
Chandra yuda <17312005>
=====
Ketik nama dan argumen program, dan tekan enter.
Tambahkan "&" setelah argumen untuk konkurensi antara proses
Berikut ini yang dibangun:
cd
help
exit
Gunakan perintah man untuk informasi tentang program lain.
```

Penjelasan kode program

Program shell tersebut, saya meniru code program aman_shell.c yang terdapat di scele. Tetapi saya sedikit merubah pemberitahuan (komentar) yang terdapat pada code program aman shell tersebut agar mudah di pahami oleh pembaca dan tentunya saya si pembuat Laporan ini.

Spesifikasi program (program dapat menjalankan perintah apa saja)

Program yang dapat di jalankan shell sama seperti program yang dapat di jalankan pada terminal, tetapi shell hanya bisa menjalankan program seperti mkdir, ls, rmdir hanya perintah-perintah biasa.