PEMBUATAN SHELL

1. Fungsi Main

```c
int main(){
   char *input;
    char **args;
    char *pwd; // *char menunjuk pwd sekarang
    char buff[100]; //Buff pwd
    while(1){
        pwd=getcwd(buff, sizeof(buff)); // Mengecek pwd terkini
        printf(KBLU "%s#" KYEL "%s%s" KGRN " > $ " RESET,getenv("USER"),getenv("GDMSESSION"         ),pwd); //Template inputan user

        signal(SIGINT, signal_handler); //Untuk signal ctrl-c
        signal(SIGTSTP, signal_handler); //Untuk signal ctrl-z

        input=inputs();
        args=splitToToken(input);

        if(strcmp(args[0],"Exit")==0 || strcmp(args[0],"exit")==0){
            break;
        }

        if(!isBackgroundProcess(args)){
            process(args);
        }else{
            int index_apersand= isBackgroundProcess(args);
            args[index_apersand]=NULL;
            backgroundprocess(args);
        }


    }
    free(input);
    free(args);
    return 0;
}
```

2. Fungsi yang menerima Input User

```c
char* inputs(){ // Fungsi untuk menerima inputan user
    int Max=MaxInput;
        char* input=malloc(sizeof(char)*Max);
    getline(&input, &Max, stdin);
    return input;
}
```

3. Fungsi yang memecah input user

```c
void backgroundprocess(char**args){ //Fungsi membuat proses dr inputan user
    pid_t process,sid;
    int status;
    process=fork();
    if(process>0){
        return ;
    }else{
        umask(0);
sid = setsid();
       if (strcmp(args[0],"cd")==0){
            chdir(args[1]); //arg[1] menunjukan direktori setelah user mengetik cd
        }else{
            execvp(args[0], args); // Eksekusi Inputan user
        }

    }
}
```

4. Fungsi yang mengeksekusi inputan yang telah dipecah

```c
void process(char**args){ //Fungsi membuat proses dr inputan user
    pid_t process,wpid;
    int status;
    process=fork();
    if(process==0){
        if (strcmp(args[0],"cd")==0){
            chdir(args[1]); //arg[1] menunjukan direktori setelah user mengetik cd
        }else{
            execvp(args[0], args); // Eksekusi Inputan user
        }

    }else{
        do {
            wpid = waitpid(process, &status, WUNTRACED); // MEnunggu proses anak
        } while (!WIFEXITED(status) && !WIFSIGNALED(status));
    }
}
```

5.Mengecek apakah inputan meminta proses background taau tidak

```c
int isBackgroundProcess(char**args){
    int i=0;
    int Max=MaxInput;
    for(i=0;i<MaxInput;i++){
        if(args[i]==NULL){
            return 0;
        }else if(strcmp(args[i],"&")==0){
            return i;
        }
    }
}
```

6. Mengeksekusi inputan user secara background proses

```c
char **splitToToken(char* input){ // Fungsi untuk memecah inputan user
    int Max=MaxInput;
    int index=0;
    char *token; // Inputan utuh
    char **tokens=malloc(sizeof(char)*Max); // tokens untuk menampung pecahan2 token
    token = strtok (input,"\n \t\r\a");
    while(token!=NULL){
        tokens[index]=token;
        index++;
        token=strtok(NULL,"\n \t\r\a");
    }
    return tokens;
}
```

7. Menangani sinyal ctrl-c dan ctrl-z

```c
void signal_handler(int signo) // Fungsi penangkap sinyal
{
  if (signo == SIGTSTP){
        printf("\n");
        main();
        signal(SIGTSTP, SIG_IGN); //Untuk signal ctrl-z
    }
 else if (signo == SIGINT){
        printf("\n");
        main();
        signal(SIGINT, SIG_IGN); //Untuk signal ctrl-c
    }
}
```

8. Contoh penggunaan perintah ps

```
jeffry@nasri-HP-Pavilion-g4-Notebook-PC ~ $ gcc -o shell shell.c
jeffry@nasri-HP-Pavilion-g4-Notebook-PC ~ $ ./shell
jeffry#cinnamon/home/jeffry > $ ps
  PID TTY          TIME CMD
 3258 pts/1    00:00:00 bash
 4199 pts/1    00:00:00 shell
 4200 pts/1    00:00:00 ps
159371272jeffry#cinnamon/home/jeffry > $
```

9. Contoh penggunaan perintah cd

```
jeffry#cinnamon/home/jeffry > $ cd Documents
jeffry#cinnamon/home/jeffry/Documents > $ ls
Asistensi1    out.txt        signal.c          strtok           tugasignal
bash-3.2      Praktikum1     sinyalctrl-z.c    strtok.c         tugassignal.c
Bersih2.psd   praktikum1.sh  sinyalinterupt    tambah.sh        Untitled-1.psd
bin           Praktikum2     sinyalinterupt.c  test
fork          SEsilab1       strstr            thread
fork.c        signal         strstr.c          THreadcreating.c
jeffry#cinnamon/home/jeffry/Documents > $ exit
jeffry#cinnamon/home/jeffry > $ exit
jeffry@nasri-HP-Pavilion-g4-Notebook-PC ~ $
```

PENGHITUNG JUMLAH BILANGAN PRIMA

1. Fungsi Main, dimana pembuatan thread dilakukan sebanyak jumlah input kita, dan tiap thread akan mengecek apakah bilangan tersebut prima atau bukan

```
void main () {
        int input,i=1,j=1,banyak_bilprim =0;
        printf("Input some number : ");scanf("%d",&input);
        pthread_t t1[input];
        void* bilprim;
        while(i <= input){
        pthread_create(&(t1[i]), NULL, find_prime, (void*)i);
                i++;
        }
        while(j <= input){
        pthread_join(t1[j], &bilprim);
                if(bilprim!=0){
                        printf("%d ",(int)bilprim);
                        banyak_bilprim++;
        }
        j++;
        }

        printf("\nTerdapat %d bilangan prima \n",banyak_bilprim);

}
```

2. Fungsi pengecekan bilangan prima, jika prima akan return ke nilai primanya. Jika BUKAN prima kan return 0

```
#include<stdio.h>
#include <stdlib.h>
#include <pthread.h>
void *find_prime(void *args) {
  int maks= (int)args;
  int count;
        if(maks==2){
                return (void *)maks;
        }
        for(count=2;count<maks;count++){
                if(maks%count==0){
                        break;
                }else if(count==(maks-1)){
                        return (void*)maks;
                }else{
                        continue;
                }
        }
        pthread_exit(NULL); // Kalau tidak ada ini, jika maks-nya 2 maka return nya 2 kali
}
```

3. Contoh program dijalankan

```
jeffry@nasri-HP-Pavilion-g4-Notebook-PC ~ $ ./Find_Prime_and_Copy_File
Input some number : 8
2 3 5 7
Terdapat 4 bilangan prima
jeffry@nasri-HP-Pavilion-g4-Notebook-PC ~ $ ./Find_Prime_and_Copy_File
Input some number : 11
2 3 5 7 11
Terdapat 5 bilangan prima
jeffry@nasri-HP-Pavilion-g4-Notebook-PC ~ $
```