netlab

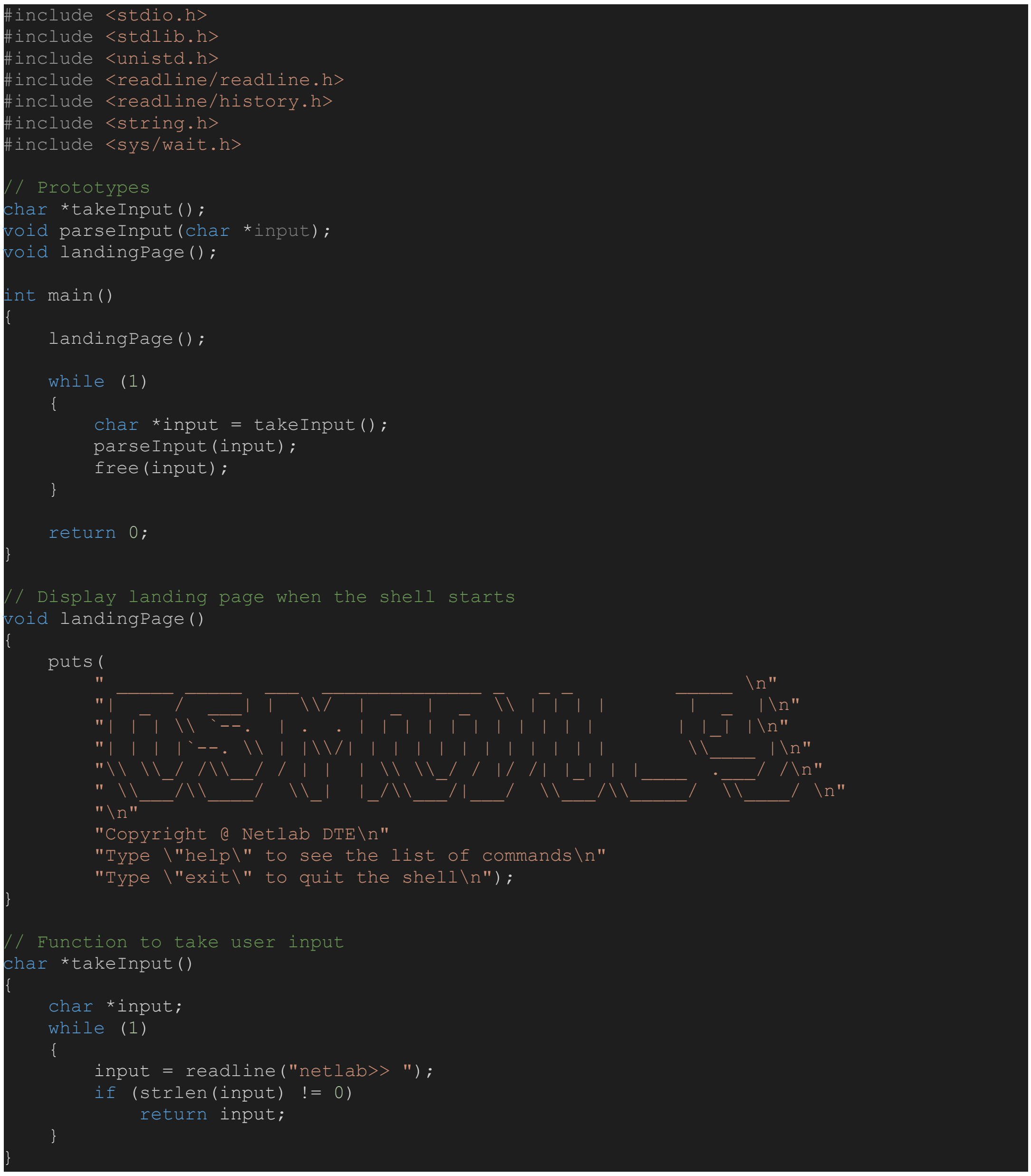
Mochtar Riady Plaza Quantum, Department of Electrical
Engineering, 3rd Floor, Universitas Indonesia, Depok 16424

PRAKTIKUM
SISTEM OPERASI

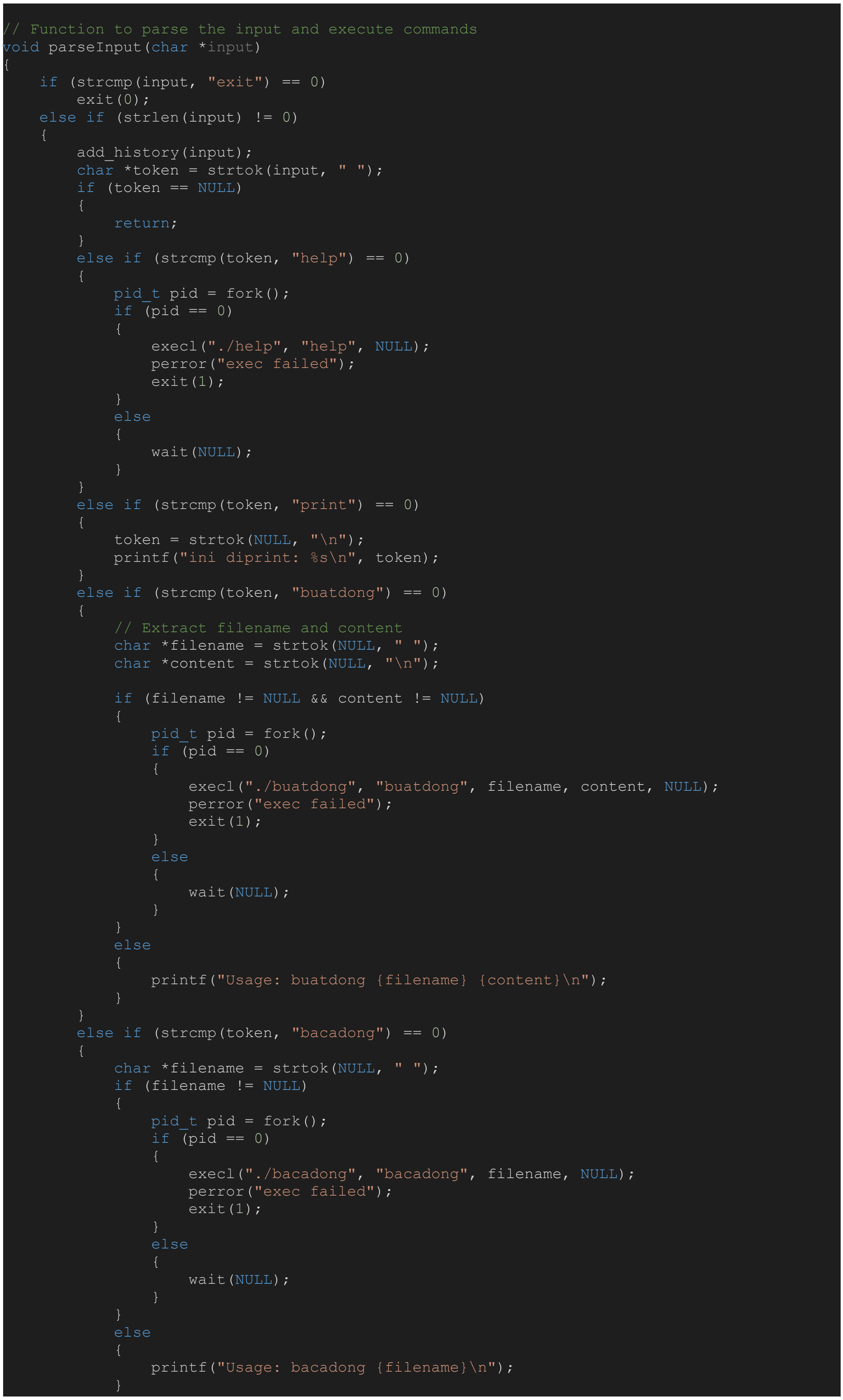
Nama	R. Aisha Syauqi Ramadhani	**No. Modul**	9
NPM	2306250554	**Tipe**	CS

Fitur Utama:

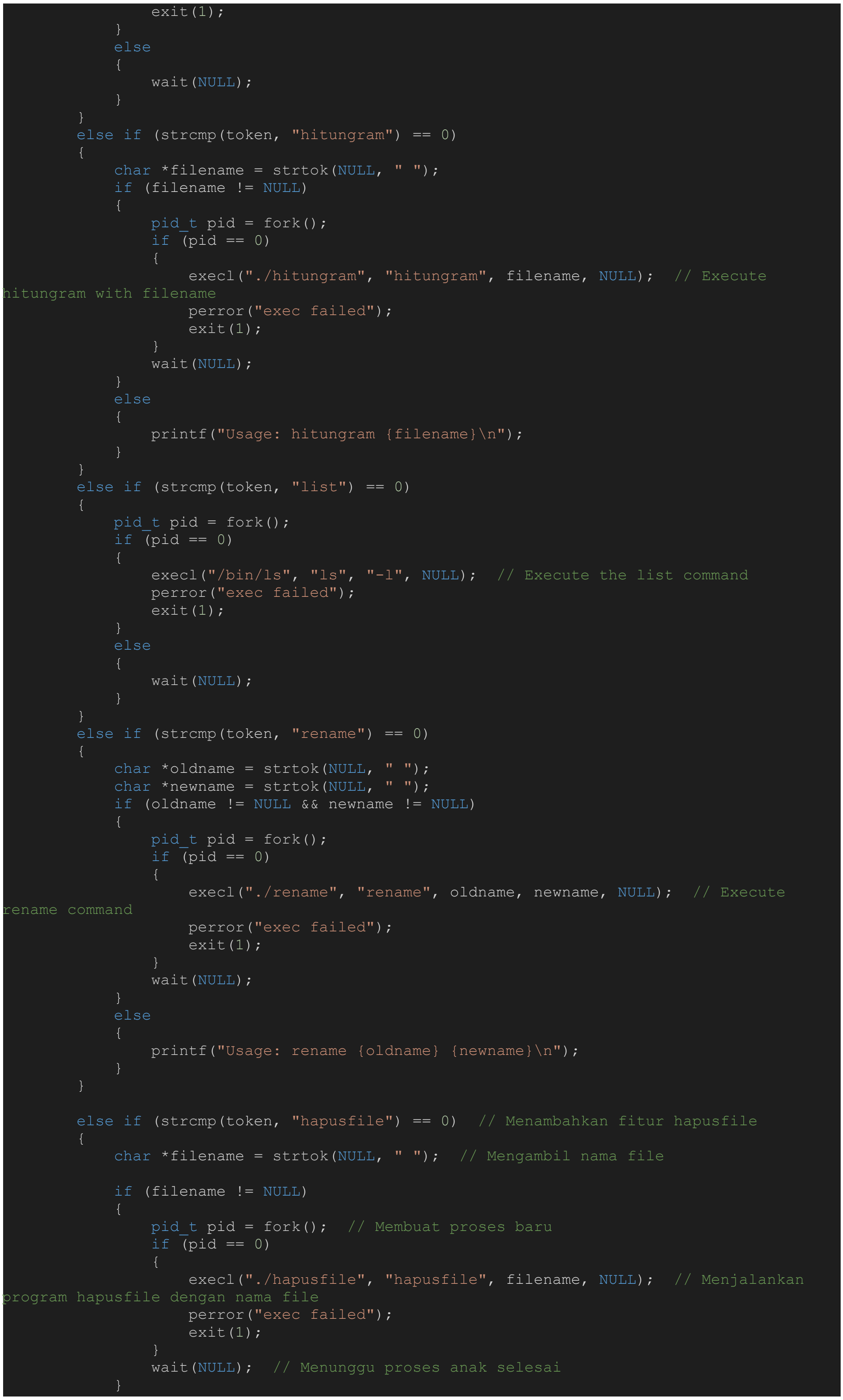
1. **Dokumentasi dan penjelasan**

File	Code
Shell.c (inc fitur tambahan)	

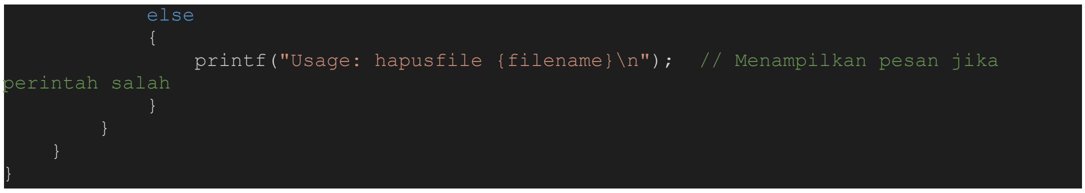
```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <readline/readline.h>
#include <readline/history.h>
#include <string.h>
#include <sys/wait.h>

// Prototypes
char *takeInput();
void parseInput(char *input);
void landingPage();

int main()
{
    landingPage();

    while (1)
    {
        char *input = takeInput();
        parseInput(input);
        free(input);
    }

    return 0;
}

// Display landing page when the shell starts
void landingPage()
{
    puts(
        "  _____ _____  ___ ______________ _  _ _        _____ \n"
        "|  _  /  ___| |  \\/  |  _  |  _  \\ | | | |       |  _  |\n"
        "| | | \\ `--.  | .  . | | | | | | | | | | |      | |_| |\n"
        "| | | |`--. \\ | |\\/| | | | | | | | | | | |       \\____ |\n"
        "\\ \\_/ /\\__/ / | |  | \\ \\_/ / |/ /| |_| | |____   .___/ /\n"
        " \\___/\\____/  \\_|  |_/\\___/|___/  \\___/\\_____/   \\____/ \n"
        "\n"
        "Copyright @ Netlab DTE\n"
        "Type \"help\" to see the list of commands\n"
        "Type \"exit\" to quit the shell\n");
}

// Function to take user input
char *takeInput()
{
    char *input;
    while (1)
    {
        input = readline("netlab>> ");
        if (strlen(input) != 0)
            return input;
    }
}
```



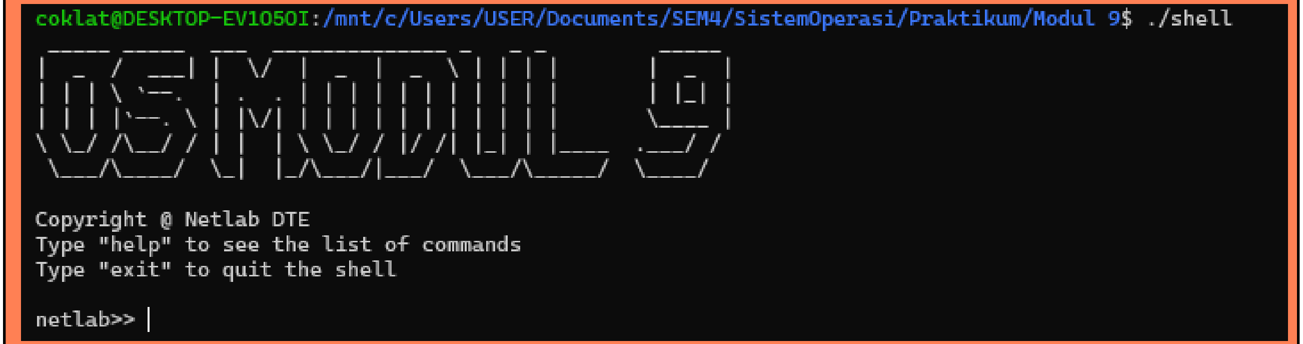
```c
// Function to parse the input and execute commands
void parseInput(char *input)
{
    if (strcmp(input, "exit") == 0)
        exit(0);
    else if (strlen(input) != 0)
    {
        add_history(input);
        char *token = strtok(input, " ");
        if (token == NULL)
        {
            return;
        }
        else if (strcmp(token, "help") == 0)
        {
            pid_t pid = fork();
            if (pid == 0)
            {
                execl("./help", "help", NULL);
                perror("exec failed");
                exit(1);
            }
            else
            {
                wait(NULL);
            }
        }
        else if (strcmp(token, "print") == 0)
        {
            token = strtok(NULL, "\n");
            printf("ini diprint: %s\n", token);
        }
        else if (strcmp(token, "buatdong") == 0)
        {
            // Extract filename and content
            char *filename = strtok(NULL, " ");
            char *content = strtok(NULL, "\n");

            if (filename != NULL && content != NULL)
            {
                pid_t pid = fork();
                if (pid == 0)
                {
                    execl("./buatdong", "buatdong", filename, content, NULL);
                    perror("exec failed");
                    exit(1);
                }
                else
                {
                    wait(NULL);
                }
            }
            else
            {
                printf("Usage: buatdong {filename} {content}\n");
            }
        }
        else if (strcmp(token, "bacadong") == 0)
        {
            char *filename = strtok(NULL, " ");
            if (filename != NULL)
            {
                pid_t pid = fork();
                if (pid == 0)
                {
                    execl("./bacadong", "bacadong", filename, NULL);
                    perror("exec failed");
                    exit(1);
                }
                else
                {
                    wait(NULL);
                }
            }
            else
            {
                printf("Usage: bacadong {filename}\n");
            }
        }
        else if (strcmp(token, "rahasiabanget") == 0)
        {
            char *filename = strtok(NULL, " ");
            char *content = strtok(NULL, "\n");

            pid_t pid = fork();
            if (pid == 0)
            {
                execl("./rahasiabanget", "rahasiabanget", filename, content, NULL);
                perror("exec failed");
                exit(1);
            }
            wait(NULL);
        }

        else if (strcmp(token, "pembacapikiran") == 0) {
            char *filename = strtok(NULL, " ");

            if (filename != NULL) {
                pid_t pid = fork();
                if (pid == 0) {
                    execl("./pembacapikiran", "pembacapikiran", filename, NULL);
                    perror("exec failed");
                    exit(1);
                }
                wait(NULL);
            } else {
                printf("Usage: pembacapikiran {filename}\n");
            }
        }

        else if (strcmp(token, "itungwoi") == 0)
        {
            char *operation = strtok(NULL, " ");
            char *num1_str = strtok(NULL, " ");
            char *num2_str = strtok(NULL, " ");

            pid_t pid = fork();
            if (pid == 0)
            {
                if (operation != NULL && num1_str != NULL && num2_str != NULL) {
                    execl("./itungwoi", "itungwoi", operation, num1_str, num2_str, NULL);
                } else if (operation != NULL) {
                    execl("./itungwoi", "itungwoi", operation, NULL);
                } else {
                    execl("./itungwoi", "itungwoi", NULL);
                }
                perror("exec failed");
                exit(1);
            }
            else
            {
                wait(NULL);
            }
        }

        else if (strcmp(token, "clear") == 0)
        {
            pid_t pid = fork();
            if (pid == 0)
            {
                execl("/bin/clear", "clear", NULL);  // Execute the clear command
                perror("exec failed");
                exit(1);
            }
            else
            {
                wait(NULL);
            }
        }
        else if (strcmp(token, "waktusekarang") == 0)
        {
            pid_t pid = fork();
            if (pid == 0)
            {
                execl("./waktusekarang", "waktusekarang", NULL);  // Execute the waktu
command
                perror("exec failed");
```



```
                exit(1);
            }
            else
            {
                wait(NULL);
            }
        }
        else if (strcmp(token, "hitungram") == 0)
        {
            char *filename = strtok(NULL, " ");
            if (filename != NULL)
            {
                pid_t pid = fork();
                if (pid == 0)
                {
                    execl("./hitungram", "hitungram", filename, NULL);  // Execute
hitungram with filename
                    perror("exec failed");
                    exit(1);
                }
                wait(NULL);
            }
            else
            {
                printf("Usage: hitungram {filename}\n");
            }
        }
        else if (strcmp(token, "list") == 0)
        {
            pid_t pid = fork();
            if (pid == 0)
            {
                execl("/bin/ls", "ls", "-l", NULL);  // Execute the list command
                perror("exec failed");
                exit(1);
            }
            else
            {
                wait(NULL);
            }
        }
        else if (strcmp(token, "rename") == 0)
        {
            char *oldname = strtok(NULL, " ");
            char *newname = strtok(NULL, " ");
            if (oldname != NULL && newname != NULL)
            {
                pid_t pid = fork();
                if (pid == 0)
                {
                    execl("./rename", "rename", oldname, newname, NULL);  // Execute
rename command
                    perror("exec failed");
                    exit(1);
                }
                wait(NULL);
            }
            else
            {
                printf("Usage: rename {oldname} {newname}\n");
            }
        }

        else if (strcmp(token, "hapusfile") == 0)  // Menambahkan fitur hapusfile
        {
            char *filename = strtok(NULL, " ");  // Mengambil nama file

            if (filename != NULL)
            {
                pid_t pid = fork();  // Membuat proses baru
                if (pid == 0)
                {
                    execl("./hapusfile", "hapusfile", filename, NULL);  // Menjalankan
program hapusfile dengan nama file
                    perror("exec failed");
                    exit(1);
                }
                wait(NULL);  // Menunggu proses anak selesai
            }
```

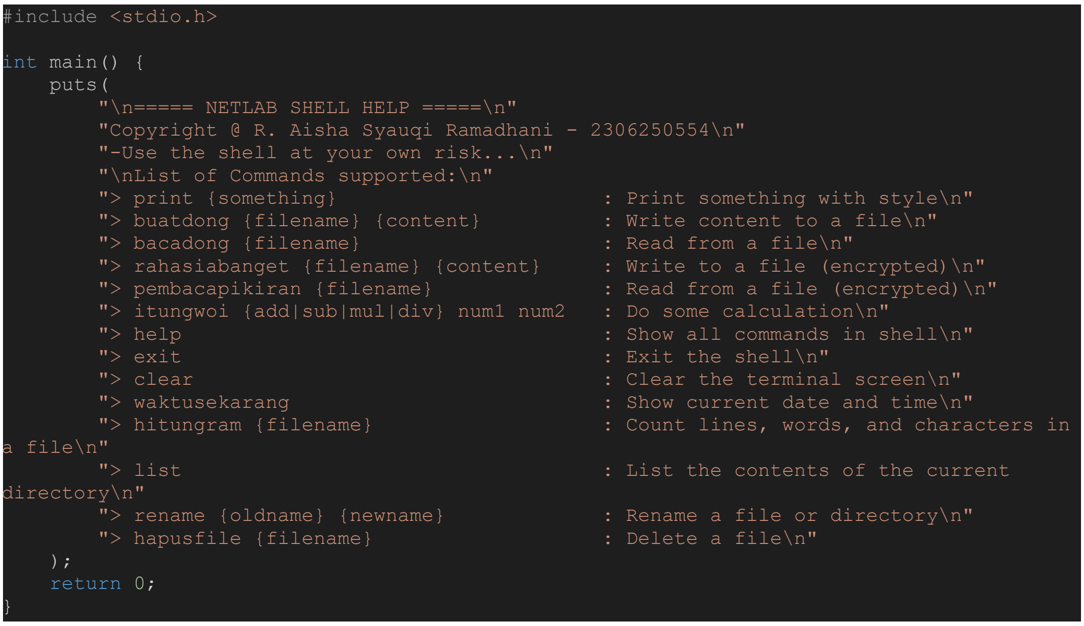


```
            else
            {
                printf("Usage: hapusfile {filename}\n");  // Menampilkan pesan jika
perintah salah
            }
        }
    }
}
```

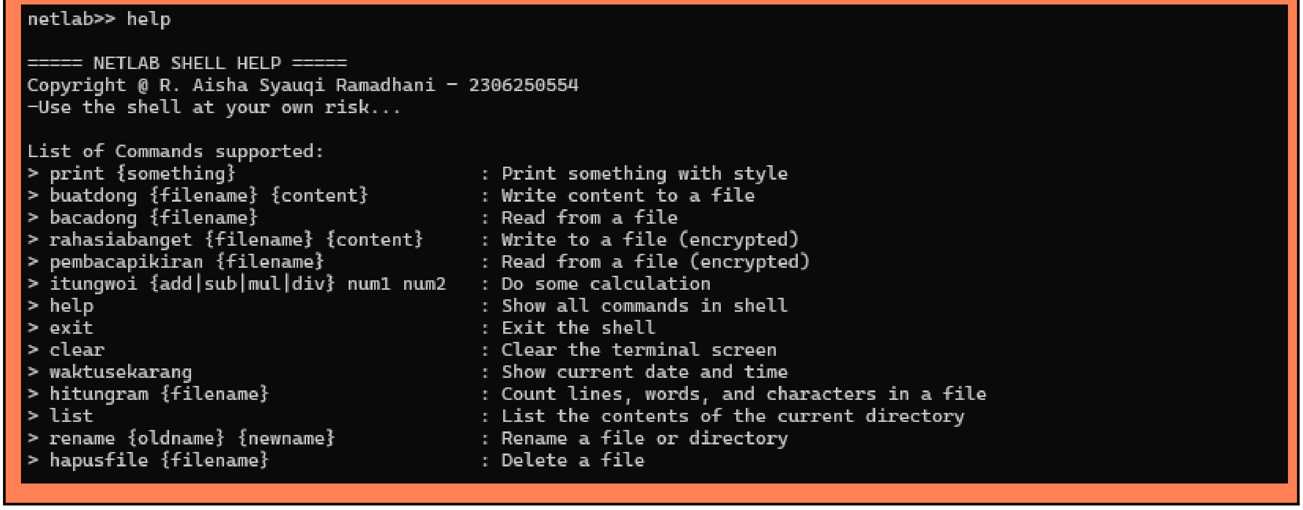


```
coklat@DESKTOP-EV1O5OI:/mnt/c/Users/USER/Documents/SEM4/SistemOperasi/Praktikum/Modul 9$ ./shell
 _____  _____ ___  ___ ___________ _   _ _       _____ 
|  _  |/  ___||  \/  ||  _  |  _  \ | | | |     |  _  |
| | | |\ `--. | .  . || | | | | | | | | | |     | |_| |
| | | | `--. \| |\/| || | | | | | | | | | |     \____ |
\ \_/ //\__/ /| |  | |\ \_/ / |/ /| |_| | |____ .___/ /
 \___/ \____/ \_|  |_/ \___/|___/  \___/\_____/ \____/ 

Copyright @ Netlab DTE
Type "help" to see the list of commands
Type "exit" to quit the shell

netlab>> 
```

printHelp.c



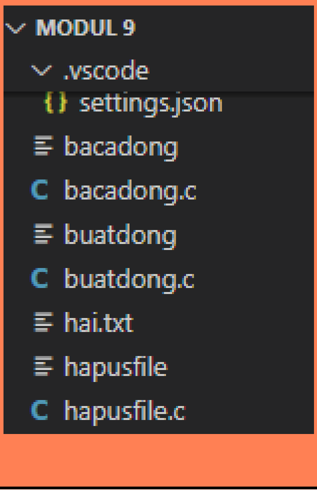
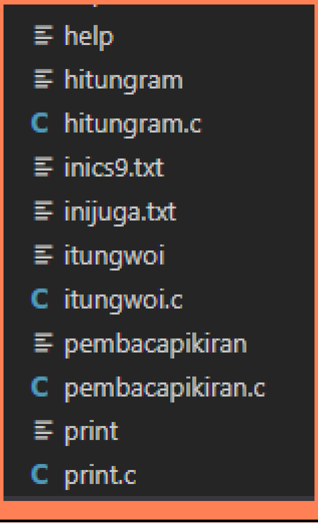
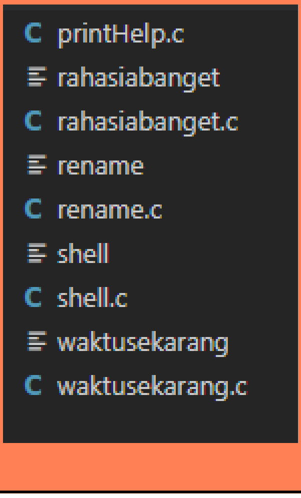
```
#include <stdio.h>

int main() {
    puts(
        "\n===== NETLAB SHELL HELP =====\n"
        "Copyright @ R. Aisha Syauqi Ramadhani - 2306250554\n"
        "-Use the shell at your own risk...\n"
        "\nList of Commands supported:\n"
        "> print {something}                         : Print something with style\n"
        "> buatdong {filename} {content}             : Write content to a file\n"
        "> bacadong {filename}                       : Read from a file\n"
        "> rahasiabanget {filename} {content}        : Write to a file (encrypted)\n"
        "> pembacapikiran {filename}                 : Read from a file (encrypted)\n"
        "> itungwoi {add|sub|mul|div} num1 num2      : Do some calculation\n"
        "> help                                      : Show all commands in shell\n"
        "> exit                                      : Exit the shell\n"
        "> clear                                     : Clear the terminal screen\n"
        "> waktusekarang                             : Show current date and time\n"
        "> hitungram {filename}                      : Count lines, words, and characters in
a file\n"
        "> list                                      : List the contents of the current
directory\n"
        "> rename {oldname} {newname}                : Rename a file or directory\n"
        "> hapusfile {filename}                      : Delete a file\n"
    );
    return 0;
}
```



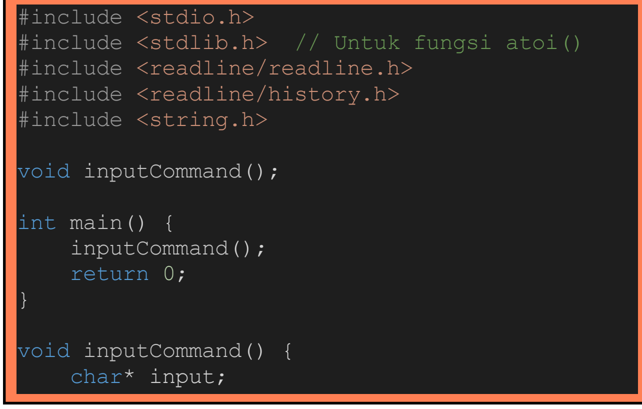
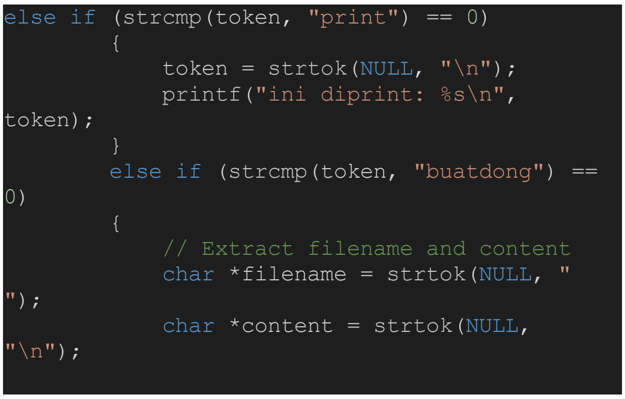
```
netlab>> help

===== NETLAB SHELL HELP =====
Copyright @ R. Aisha Syauqi Ramadhani - 2306250554
-Use the shell at your own risk...

List of Commands supported:
> print {something}                         : Print something with style
> buatdong {filename} {content}             : Write content to a file
> bacadong {filename}                       : Read from a file
> rahasiabanget {filename} {content}        : Write to a file (encrypted)
> pembacapikiran {filename}                 : Read from a file (encrypted)
> itungwoi {add|sub|mul|div} num1 num2      : Do some calculation
> help                                      : Show all commands in shell
> exit                                      : Exit the shell
> clear                                     : Clear the terminal screen
> waktusekarang                             : Show current date and time
> hitungram {filename}                      : Count lines, words, and characters in a file
> list                                      : List the contents of the current directory
> rename {oldname} {newname}                : Rename a file or directory
> hapusfile {filename}                      : Delete a file
```

Kode ini adalah implementasi sebuah shell sederhana yang memungkinkan pengguna untuk memasukkan perintah dan mengeksekusinya. Shell ini dilengkapi dengan beberapa fitur seperti menampilkan landing page, menerima input melalui readline(), dan memproses berbagai perintah seperti help, exit, print, dan beberapa perintah khusus lainnya. Setiap perintah yang dipilih akan dieksekusi oleh proses anak menggunakan fork() dan execl(), serta menunggu proses anak selesai dengan wait(). Selain itu, terdapat juga fitur tambahan untuk menghapus file (hapusfile), melihat daftar file (list), dan menjalankan program eksternal, seperti menghitung waktu (waktusekarang). Program ini juga mendukung penambahan perintah custom yang dapat dipanggil langsung dari shell yang diimplementasikan.

2. **Setiap case shellnya dibuat pada file yang berbeda**

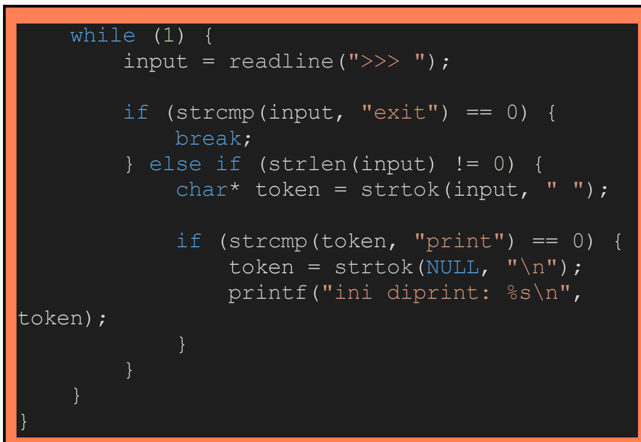
Command		
 MODUL 9 .vscode settings.json bacadong bacadong.c buatdong buatdong.c hai.txt hapusfile hapusfile.c	 help hitungram hitungram.c inics9.txt inijuga.txt itungwoi itungwoi.c pembacapikiran pembacapikiran.c print print.c	 printHelp.c rahasiabanget rahasiabanget.c rename rename.c shell shell.c waktusekarang waktusekarang.c

3. **Print.c**

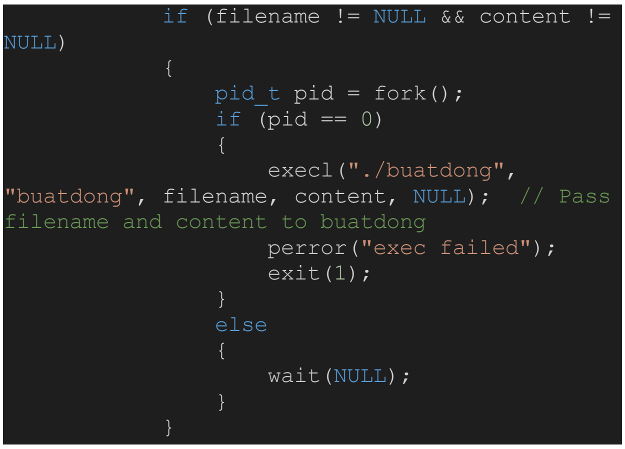
print.c	Code di shell
	

```
#include <stdio.h>
#include <stdlib.h>  // Untuk fungsi atoi()
#include <readline/readline.h>
#include <readline/history.h>
#include <string.h>

void inputCommand();

int main() {
    inputCommand();
    return 0;
}

void inputCommand() {
    char* input;
```

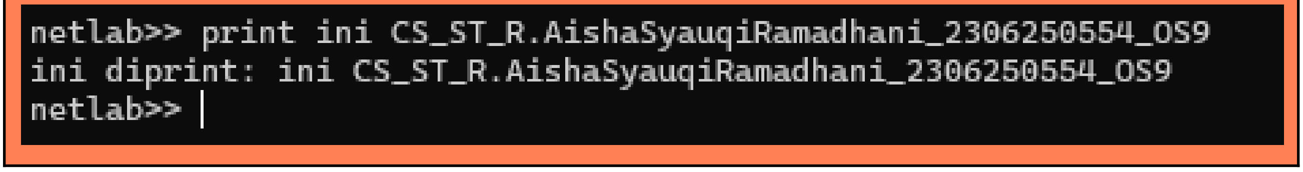
```
else if (strcmp(token, "print") == 0)
        {
            token = strtok(NULL, "\n");
            printf("ini diprint: %s\n",
token);
        }
        else if (strcmp(token, "buatdong") ==
0)
        {
            // Extract filename and content
            char *filename = strtok(NULL, "
");
            char *content = strtok(NULL,
"\n");
```



```
    while (1) {
        input = readline(">>> ");

        if (strcmp(input, "exit") == 0) {
            break;
        } else if (strlen(input) != 0) {
            char* token = strtok(input, " ");

            if (strcmp(token, "print") == 0) {
                token = strtok(NULL, "\n");
                printf("ini diprint: %s\n",
token);
            }
        }
    }
}
```



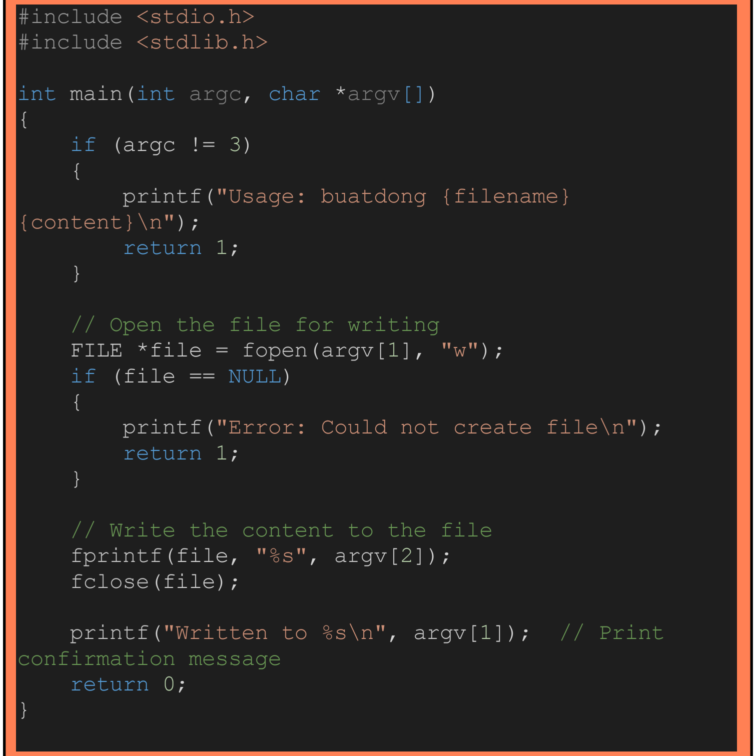
```
            if (filename != NULL && content !=
NULL)
            {
                pid_t pid = fork();
                if (pid == 0)
                {
                    execl("./buatdong",
"buatdong", filename, content, NULL);  // Pass
filename and content to buatdong
                    perror("exec failed");
                    exit(1);
                }
                else
                {
                    wait(NULL);
                }
            }
```



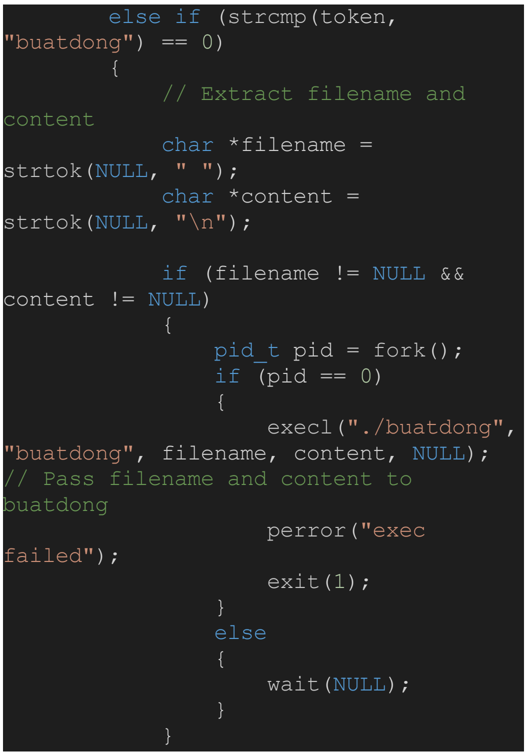
```
netlab>> print ini CS_ST_R.AishaSyauqiRamadhani_2306250554_OS9
ini diprint: ini CS_ST_R.AishaSyauqiRamadhani_2306250554_OS9
netlab>> 
```

4. Membuat dan mengisi file

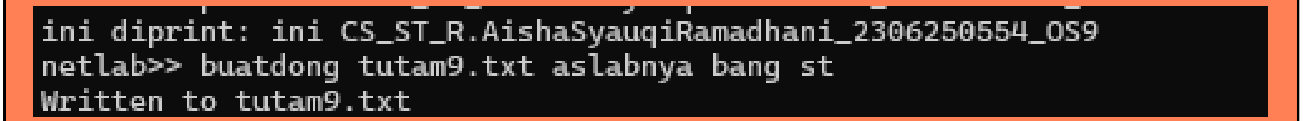
buatdong.c	Code di shell



```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    if (argc != 3)
    {
        printf("Usage: buatdong {filename}
{content}\n");
        return 1;
    }

    // Open the file for writing
    FILE *file = fopen(argv[1], "w");
    if (file == NULL)
    {
        printf("Error: Could not create file\n");
        return 1;
    }

    // Write the content to the file
    fprintf(file, "%s", argv[2]);
    fclose(file);

    printf("Written to %s\n", argv[1]);  // Print
confirmation message
    return 0;
}
```

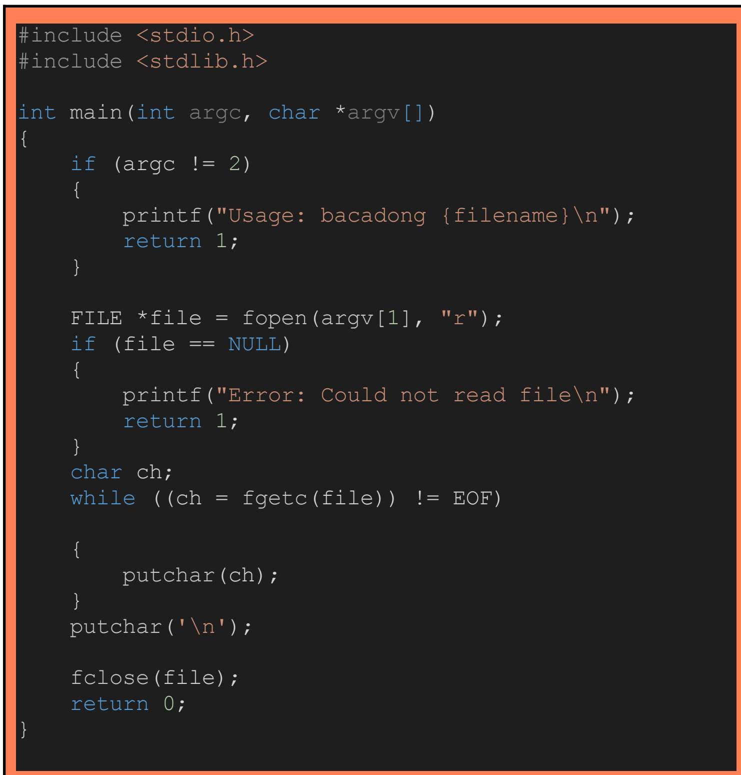


```
        else if (strcmp(token,
"buatdong") == 0)
        {
            // Extract filename and
content
            char *filename =
strtok(NULL, " ");
            char *content =
strtok(NULL, "\n");

            if (filename != NULL &&
content != NULL)
            {
                pid_t pid = fork();
                if (pid == 0)
                {
                    execl("./buatdong",
"buatdong", filename, content, NULL);
// Pass filename and content to
buatdong
                    perror("exec
failed");
                    exit(1);
                }
                else
                {
                    wait(NULL);
                }
            }
```

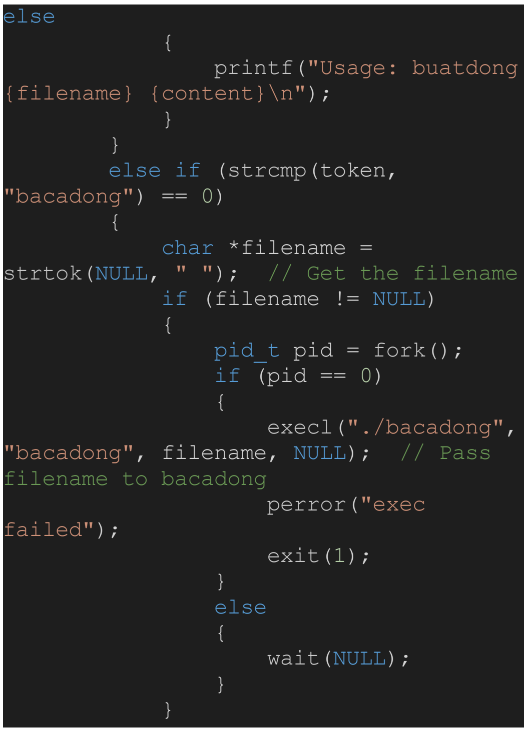


```
ini diprint: ini CS_ST_R.AishaSyauqiRamadhani_2306250554_OS9
netlab>> buatdong tutam9.txt aslabnya bang st
Written to tutam9.txt
```

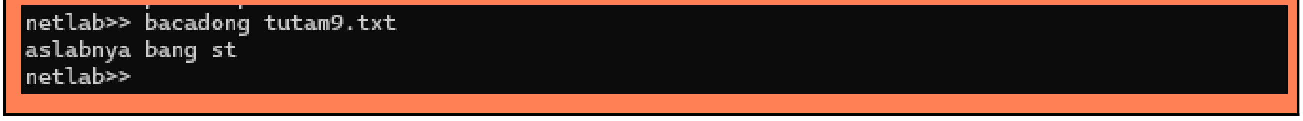
bacadong.c	Code di shell



```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    if (argc != 2)
    {
        printf("Usage: bacadong {filename}\n");
        return 1;
    }

    FILE *file = fopen(argv[1], "r");
    if (file == NULL)
    {
        printf("Error: Could not read file\n");
        return 1;
    }
    char ch;
    while ((ch = fgetc(file)) != EOF)

    {
        putchar(ch);
    }
    putchar('\n');

    fclose(file);
    return 0;
}
```



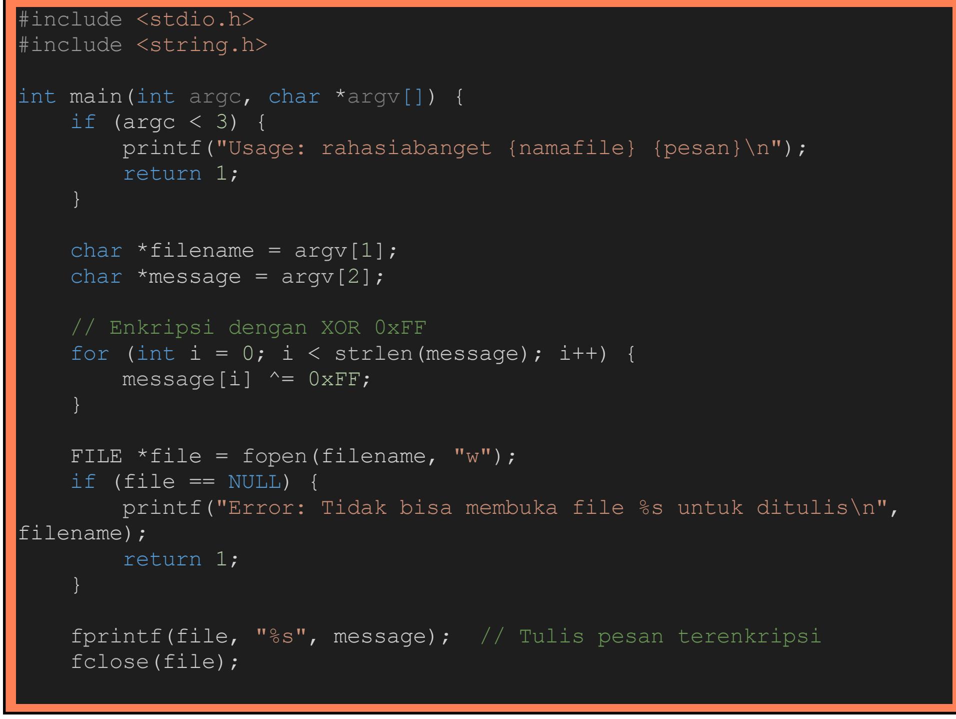
```
else
            {
                printf("Usage: buatdong
{filename} {content}\n");
            }
        }
        else if (strcmp(token,
"bacadong") == 0)
        {
            char *filename =
strtok(NULL, " ");  // Get the filename
            if (filename != NULL)
            {
                pid_t pid = fork();
                if (pid == 0)
                {
                    execl("./bacadong",
"bacadong", filename, NULL);  // Pass
filename to bacadong
                    perror("exec
failed");
                    exit(1);
                }
                else
                {
                    wait(NULL);
                }
            }
```



```
netlab>> bacadong tutam9.txt
aslabnya bang st
netlab>>
```

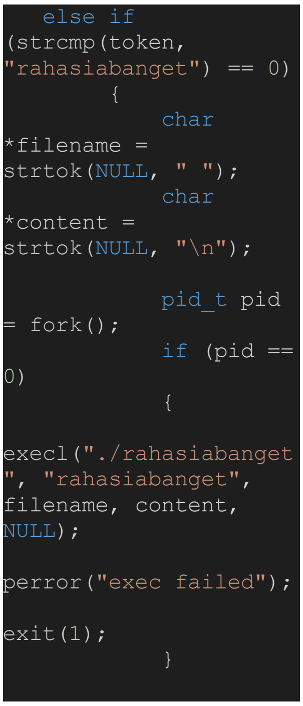
5. Enkripsi sederhana

rahasiabanget.c



```
#include <stdio.h>
#include <string.h>

int main(int argc, char *argv[]) {
    if (argc < 3) {
        printf("Usage: rahasiabanget {namafile} {pesan}\n");
        return 1;
    }

    char *filename = argv[1];
    char *message = argv[2];

    // Enkripsi dengan XOR 0xFF
    for (int i = 0; i < strlen(message); i++) {
        message[i] ^= 0xFF;
    }

    FILE *file = fopen(filename, "w");
    if (file == NULL) {
        printf("Error: Tidak bisa membuka file %s untuk ditulis\n",
filename);
        return 1;
    }

    fprintf(file, "%s", message);  // Tulis pesan terenkripsi
    fclose(file);
```

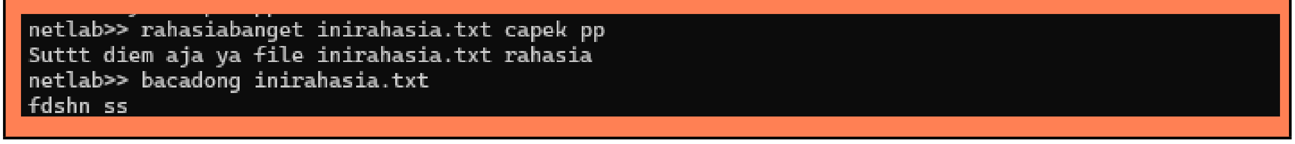
Code di shell



```
   else if
(strcmp(token,
"rahasiabanget") == 0)
        {
            char
*filename =
strtok(NULL, " ");
            char
*content =
strtok(NULL, "\n");

            pid_t pid
= fork();
            if (pid ==
0)
            {

execl("./rahasiabanget
", "rahasiabanget",
filename, content,
NULL);

perror("exec failed");

exit(1);
            }
```

```
    printf("Syut diem aja ya file %s rahasia\n", filename);
    return 0;
}
```

```
wait(NULL);
        }
```



```
netlab>> rahasiabanget inirahasia.txt capek pp
Suttt diem aja ya file inirahasia.txt rahasia
netlab>> bacadong inirahasia.txt
fdshn ss
```

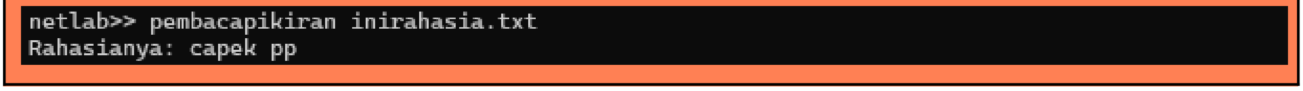
6. Deskripsi

pembacapikiran.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void decrypt(char *text) {
    int i = 0;
    while (text[i] != '\0') {
        if ((text[i] >= 'a' && text[i] <= 'z') ||
(text[i] >= 'A' && text[i] <= 'Z')) {
            if ((text[i] >= 'a' && text[i] <= 'z')) {
                text[i] = ((text[i] - 'a' + 26 - 3) %
26) + 'a';
            } else {
                text[i] = ((text[i] - 'A' + 26 - 3) %
26) + 'A';
            }
        }
        i++;
    }
}

int main(int argc, char *argv[]) {
    if (argc < 2) {
        printf("Usage: pembacapikiran {filename}\n");
        return 1;
    }

    FILE *file = fopen(argv[1], "r");
    if (file == NULL) {
        printf("Failed to open the file: %s\n",
argv[1]);
        return 1;
    }

    char content[1024];
    memset(content, 0, sizeof(content));

    fread(content, 1, sizeof(content) - 1, file);
    fclose(file);

    decrypt(content);
    printf("Rahasianya: %s\n", content);

    return 0;
}
```

Code di shell

```
        else if (strcmp(token,
"pembacapikiran") == 0) {
            char *filename =
strtok(NULL, " ");

            pid_t pid = fork();
            if (pid == 0) {
                if (filename != NULL) {
                    if
(execl("./pembacapikiran",
"pembacapikiran", filename, NULL) ==
-1) {
                        perror("exec
failed");
                        exit(1);
                    }
                } else {
                    printf("Usage:
pembacapikiran {filename}\n");
                    exit(1);
                }
            } else {
                wait(NULL);
            }
        }
```



```
netlab>> pembacapikiran inirahasia.txt
Rahasianya: capek pp
```

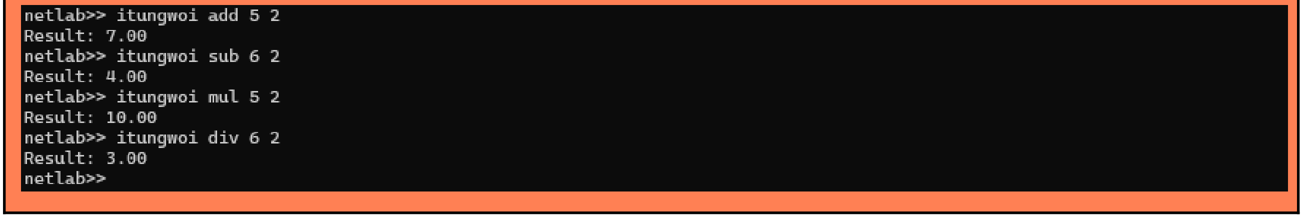
7. Kalkulator sederhana

itungwoi.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(int argc, char *argv[]) {
    if (argc < 4) {
        printf("Usage: itungwoi {add|sub|mul|div} num1
num2\n");
        return 1;
    }

    char *operation = argv[1];
    char *num1_str = argv[2];
    char *num2_str = argv[3];

    // Konversi string ke float
    float num1 = atof(num1_str);
    float num2 = atof(num2_str);
    float result;

    // Melakukan operasi sesuai perintah
    if (strcmp(operation, "add") == 0) {
        result = num1 + num2;
        printf("Result: %.2f\n", result);
    }
    else if (strcmp(operation, "sub") == 0) {
        result = num1 - num2;
        printf("Result: %.2f\n", result);
    }
    else if (strcmp(operation, "mul") == 0) {
        result = num1 * num2;
        printf("Result: %.2f\n", result);
    }
    else if (strcmp(operation, "div") == 0) {
        // Error handling untuk pembagian dengan nol
        if (num2 == 0) {
            printf("Error: Division by zero\n");
            return 1;
        }
        result = num1 / num2;
        printf("Result: %.2f\n", result);
    }
    else {
        printf("Invalid operation. Use add, sub, mul, or
div\n");
        return 1;
    }

    return 0;
}
```

Code di shell

```c
else if (strcmp(token, "itungwoi")
== 0)
        {
            char *operation =
strtok(NULL, " ");
            char *num1_str =
strtok(NULL, " ");
            char *num2_str =
strtok(NULL, " ");

            pid_t pid = fork();
            if (pid == 0)
            {
                if (operation !=
NULL && num1_str != NULL && num2_str
!= NULL) {

execl("./itungwoi", "itungwoi",
operation, num1_str, num2_str,
NULL);
                } else if (operation
!= NULL) {

execl("./itungwoi", "itungwoi",
operation, NULL);
                } else {

execl("./itungwoi", "itungwoi",
NULL);
                }
                perror("exec
failed");
                exit(1);
            }
            else
            {
                wait(NULL);
            }
        }
```



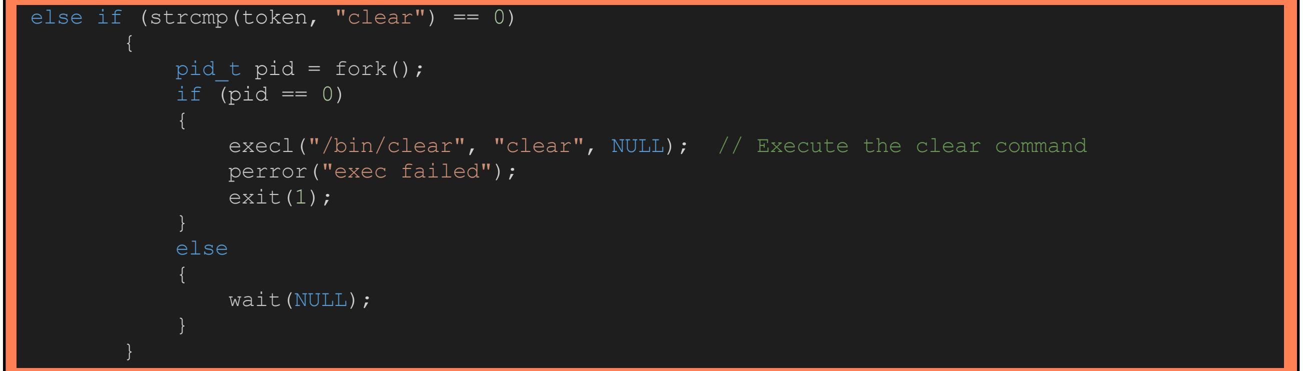
```
netlab>> itungwoi add 5 2
Result: 7.00
netlab>> itungwoi sub 6 2
Result: 4.00
netlab>> itungwoi mul 5 2
Result: 10.00
netlab>> itungwoi div 6 2
Result: 3.00
netlab>>
```

Fitur Tambahan:

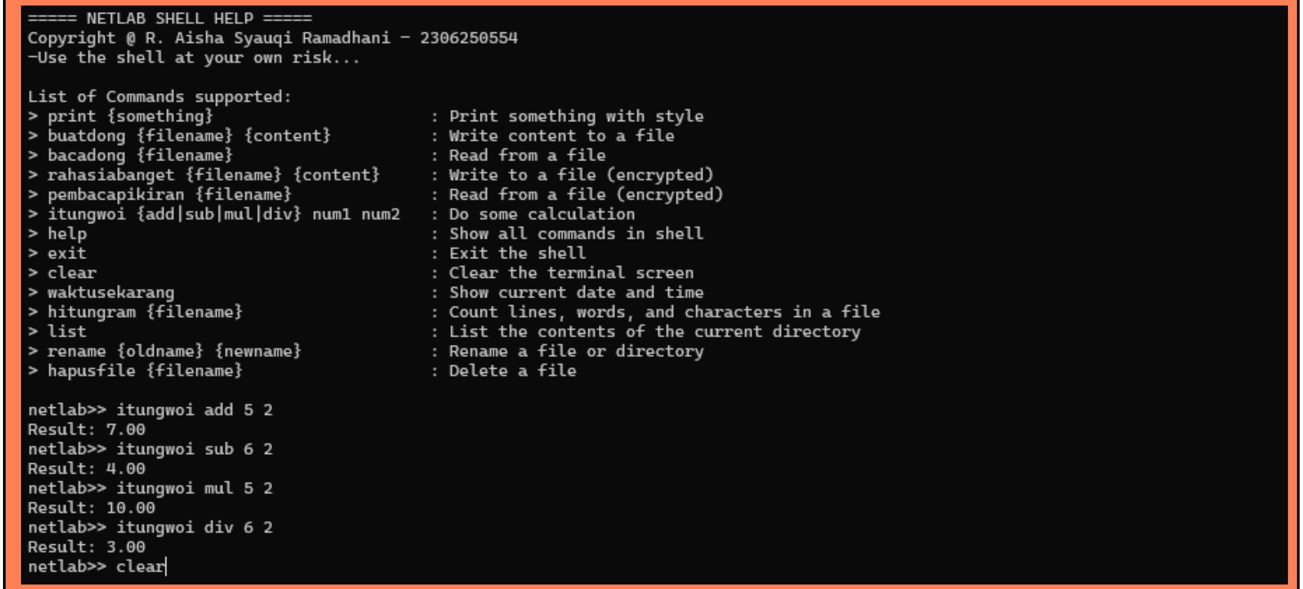
1. **Menambahkan fitur basic**

   a. **Membersihkan terminal**

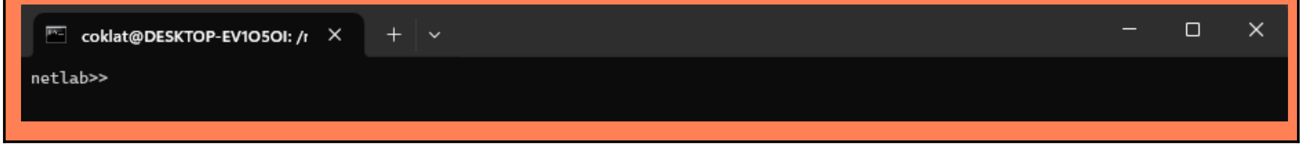
Code di shell



```c
else if (strcmp(token, "clear") == 0)
        {
            pid_t pid = fork();
            if (pid == 0)
            {
                execl("/bin/clear", "clear", NULL);   // Execute the clear command
                perror("exec failed");
                exit(1);
            }
            else
            {
                wait(NULL);
            }
        }
```



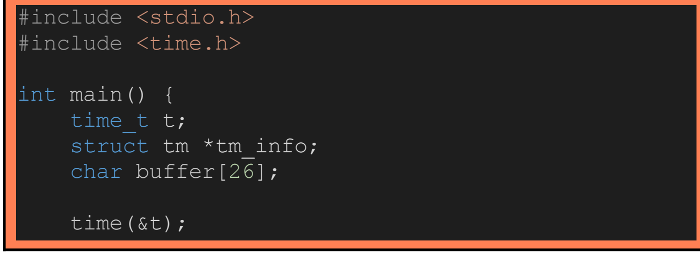
```
===== NETLAB SHELL HELP =====
Copyright @ R. Aisha Syauqi Ramadhani - 2306250554
-Use the shell at your own risk...

List of Commands supported:
> print {something}                    : Print something with style
> buatdong {filename} {content}        : Write content to a file
> bacadong {filename}                  : Read from a file
> rahasiabanget {filename} {content}   : Write to a file (encrypted)
> pembacapikiran {filename}            : Read from a file (encrypted)
> itungwoi {add|sub|mul|div} num1 num2 : Do some calculation
> help                                 : Show all commands in shell
> exit                                 : Exit the shell
> clear                                : Clear the terminal screen
> waktusekarang                        : Show current date and time
> hitungram {filename}                 : Count lines, words, and characters in a file
> list                                 : List the contents of the current directory
> rename {oldname} {newname}           : Rename a file or directory
> hapusfile {filename}                 : Delete a file

netlab>> itungwoi add 5 2
Result: 7.00
netlab>> itungwoi sub 6 2
Result: 4.00
netlab>> itungwoi mul 5 2
Result: 10.00
netlab>> itungwoi div 6 2
Result: 3.00
netlab>> clear
```



```
netlab>>
```

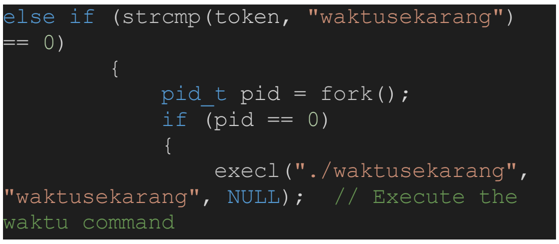
   b. **Menampilkan waktu dan tanggal sekarang**

waktusekarang.c

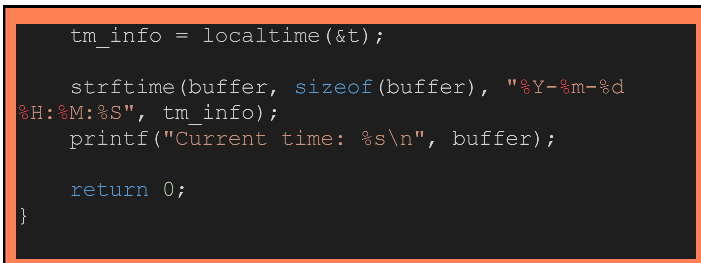


```c
#include <stdio.h>
#include <time.h>

int main() {
    time_t t;
    struct tm *tm_info;
    char buffer[26];

    time(&t);
```

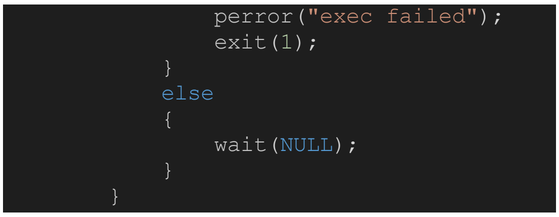
Code di shell



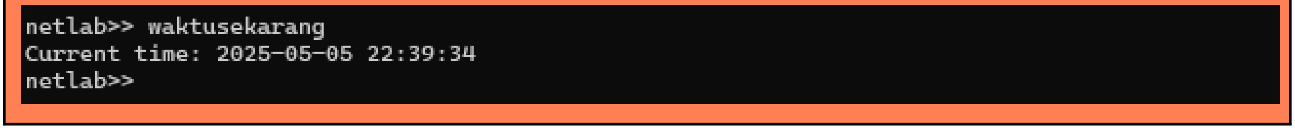
```c
else if (strcmp(token, "waktusekarang")
== 0)
        {
            pid_t pid = fork();
            if (pid == 0)
            {
                execl("./waktusekarang",
"waktusekarang", NULL);   // Execute the
waktu command
```



```
    tm_info = localtime(&t);

    strftime(buffer, sizeof(buffer), "%Y-%m-%d
%H:%M:%S", tm_info);
    printf("Current time: %s\n", buffer);

    return 0;
}
```



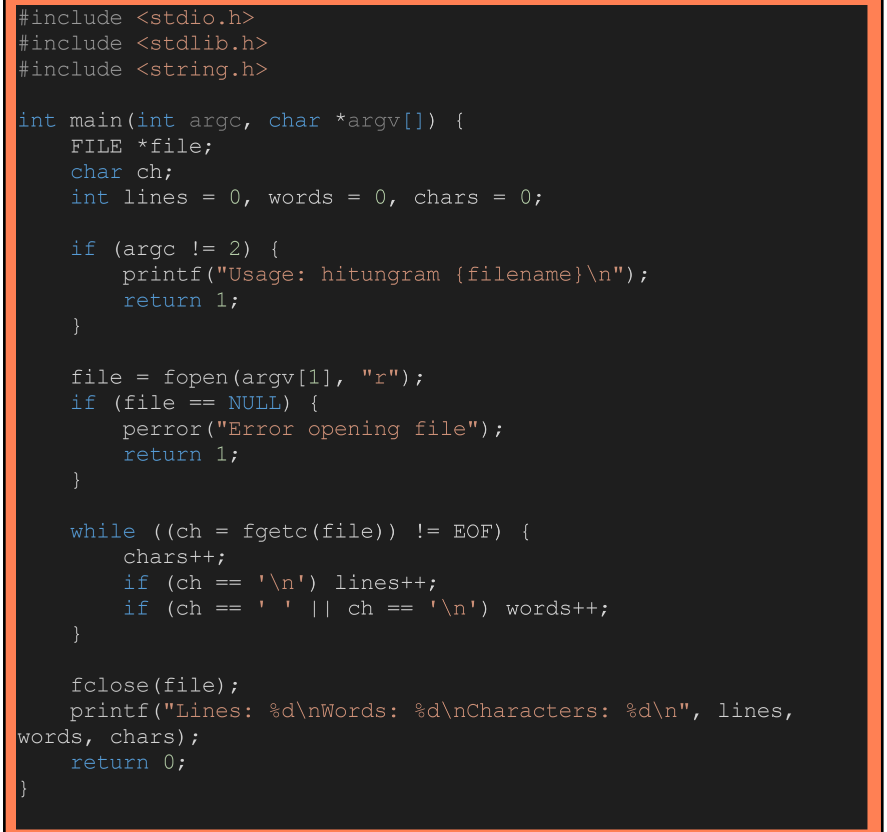
```
                perror("exec failed");
                exit(1);
            }
            else
            {
                wait(NULL);
            }
        }
```



```
netlab>> waktusekarang
Current time: 2025-05-05 22:39:34
netlab>>
```

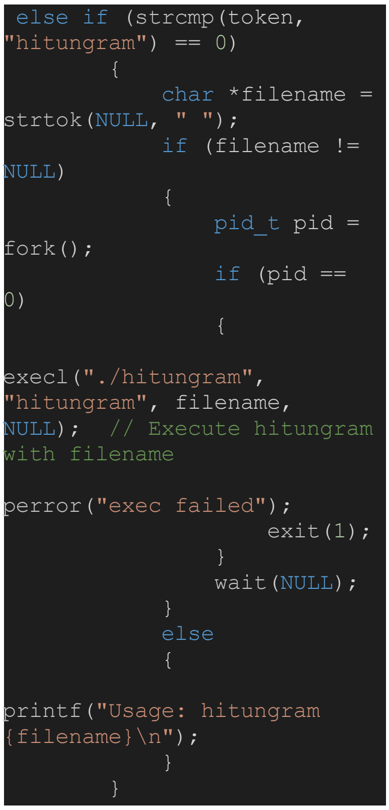
c. Menghitung baris, kata, dan karakter dari file

hitungram.c

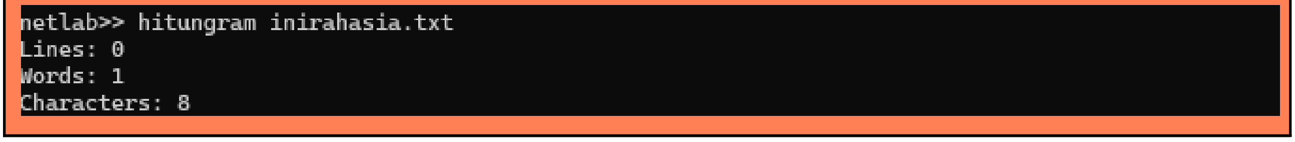


```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(int argc, char *argv[]) {
    FILE *file;
    char ch;
    int lines = 0, words = 0, chars = 0;

    if (argc != 2) {
        printf("Usage: hitungram {filename}\n");
        return 1;
    }

    file = fopen(argv[1], "r");
    if (file == NULL) {
        perror("Error opening file");
        return 1;
    }

    while ((ch = fgetc(file)) != EOF) {
        chars++;
        if (ch == '\n') lines++;
        if (ch == ' ' || ch == '\n') words++;
    }

    fclose(file);
    printf("Lines: %d\nWords: %d\nCharacters: %d\n", lines,
words, chars);
    return 0;
}
```

Code di shell



```
 else if (strcmp(token,
"hitungram") == 0)
        {
            char *filename =
strtok(NULL, " ");
            if (filename !=
NULL)
            {
                pid_t pid =
fork();
                if (pid ==
0)
                {

execl("./hitungram",
"hitungram", filename,
NULL);  // Execute hitungram
with filename

perror("exec failed");
                    exit(1);
                }
                wait(NULL);
            }
            else
            {

printf("Usage: hitungram
{filename}\n");
            }
        }
```

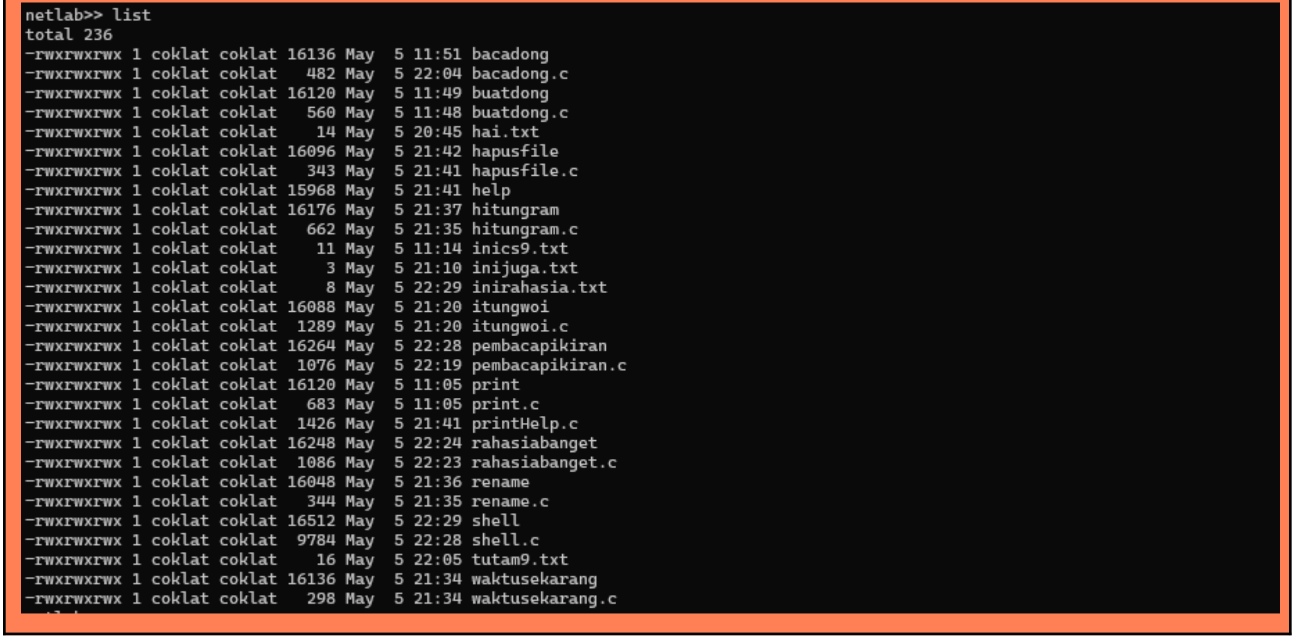


```
netlab>> hitungram inirahasia.txt
Lines: 0
Words: 1
Characters: 8
```

d. Menampilkan isi direktori saat ini

Code di shell

```
else if (strcmp(token, "list") == 0)
        {
            pid_t pid = fork();
            if (pid == 0)
            {
                execl("/bin/ls", "ls", "-l", NULL);  // Execute the list command
                perror("exec failed");
                exit(1);
            }
            else
            {
                wait(NULL);
            }
        }
```



```
netlab>> list
total 236
-rwxrwxrwx 1 coklat coklat 16136 May  5 11:51 bacadong
-rwxrwxrwx 1 coklat coklat   482 May  5 22:04 bacadong.c
-rwxrwxrwx 1 coklat coklat 16120 May  5 11:49 buatdong
-rwxrwxrwx 1 coklat coklat   560 May  5 11:48 buatdong.c
-rwxrwxrwx 1 coklat coklat    14 May  5 20:45 hai.txt
-rwxrwxrwx 1 coklat coklat 16096 May  5 21:42 hapusfile
-rwxrwxrwx 1 coklat coklat   343 May  5 21:41 hapusfile.c
-rwxrwxrwx 1 coklat coklat 15968 May  5 21:41 help
-rwxrwxrwx 1 coklat coklat 16176 May  5 21:37 hitungram
-rwxrwxrwx 1 coklat coklat   662 May  5 21:35 hitungram.c
-rwxrwxrwx 1 coklat coklat    11 May  5 11:14 inics9.txt
-rwxrwxrwx 1 coklat coklat     3 May  5 21:10 inijuga.txt
-rwxrwxrwx 1 coklat coklat     8 May  5 22:29 inirahasia.txt
-rwxrwxrwx 1 coklat coklat 16088 May  5 21:20 itungwoi
-rwxrwxrwx 1 coklat coklat  1289 May  5 21:20 itungwoi.c
-rwxrwxrwx 1 coklat coklat 16264 May  5 22:28 pembacapikiran
-rwxrwxrwx 1 coklat coklat  1076 May  5 22:19 pembacapikiran.c
-rwxrwxrwx 1 coklat coklat 16120 May  5 11:05 print
-rwxrwxrwx 1 coklat coklat   683 May  5 11:05 print.c
-rwxrwxrwx 1 coklat coklat  1426 May  5 21:41 printHelp.c
-rwxrwxrwx 1 coklat coklat 16248 May  5 22:24 rahasiabanget
-rwxrwxrwx 1 coklat coklat  1086 May  5 22:23 rahasiabanget.c
-rwxrwxrwx 1 coklat coklat 16048 May  5 21:36 rename
-rwxrwxrwx 1 coklat coklat   344 May  5 21:35 rename.c
-rwxrwxrwx 1 coklat coklat 16512 May  5 22:29 shell
-rwxrwxrwx 1 coklat coklat  9784 May  5 22:28 shell.c
-rwxrwxrwx 1 coklat coklat    16 May  5 22:05 tutam9.txt
-rwxrwxrwx 1 coklat coklat 16136 May  5 21:34 waktusekarang
-rwxrwxrwx 1 coklat coklat   298 May  5 21:34 waktusekarang.c
```

e. Mengganti nama file atau direktori

rename.c

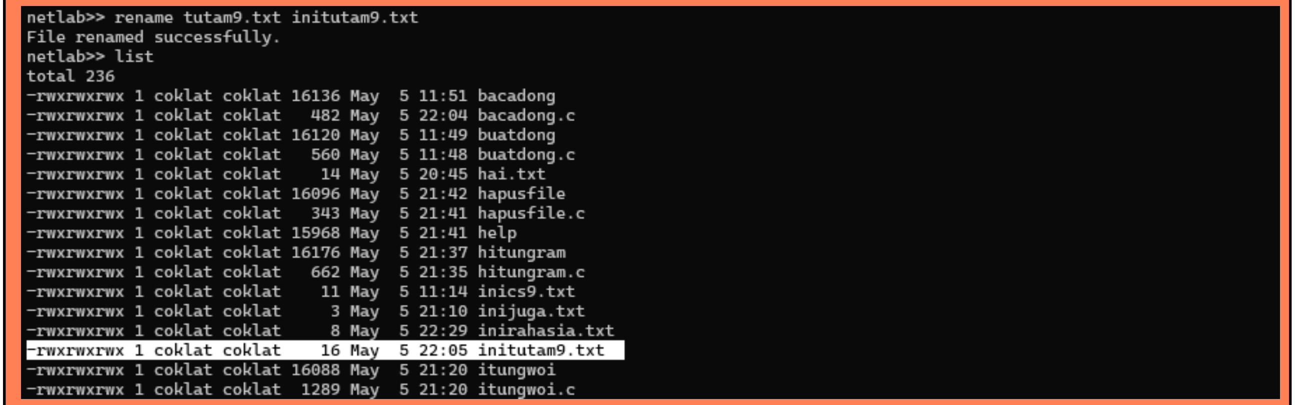
```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    if (argc != 3) {
        printf("Usage: rename {oldname} {newname}\n");
        return 1;
    }

    if (rename(argv[1], argv[2]) == 0)
        printf("File renamed successfully.\n");
    else
        perror("Error renaming file");
```

Code di shell

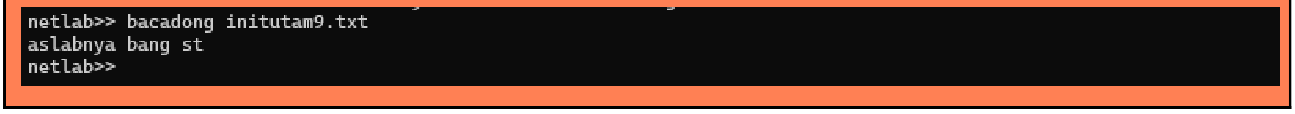
```
else if (strcmp(token, "rename") == 0)
       {
           char *oldname = strtok(NULL, " ");
           char *newname = strtok(NULL, " ");
           if (oldname != NULL && newname != NULL)
           {
               pid_t pid = fork();
               if (pid == 0)
               {
                   execl("./rename", "rename", oldname, newname, NULL);  // Execute rename command
                   perror("exec failed");
                   exit(1);
               }
               wait(NULL);
           }
           else
```

```
    return 0;
}
```

```
                {
                    printf("Usage: rename {oldname}
{newname}\n");
                }
```



```
netlab>> rename tutam9.txt initutam9.txt
File renamed successfully.
netlab>> list
total 236
-rwxrwxrwx 1 coklat coklat 16136 May  5 11:51 bacadong
-rwxrwxrwx 1 coklat coklat   482 May  5 22:04 bacadong.c
-rwxrwxrwx 1 coklat coklat 16120 May  5 11:49 buatdong
-rwxrwxrwx 1 coklat coklat   560 May  5 11:48 buatdong.c
-rwxrwxrwx 1 coklat coklat    14 May  5 20:45 hai.txt
-rwxrwxrwx 1 coklat coklat 16096 May  5 21:42 hapusfile
-rwxrwxrwx 1 coklat coklat   343 May  5 21:41 hapusfile.c
-rwxrwxrwx 1 coklat coklat 15968 May  5 21:41 help
-rwxrwxrwx 1 coklat coklat 16176 May  5 21:37 hitungram
-rwxrwxrwx 1 coklat coklat   662 May  5 21:35 hitungram.c
-rwxrwxrwx 1 coklat coklat    11 May  5 11:14 inics9.txt
-rwxrwxrwx 1 coklat coklat     3 May  5 21:10 inijuga.txt
-rwxrwxrwx 1 coklat coklat     8 May  5 22:29 inirahasia.txt
-rwxrwxrwx 1 coklat coklat    16 May  5 22:05 initutam9.txt
-rwxrwxrwx 1 coklat coklat 16088 May  5 21:20 itungwoi
-rwxrwxrwx 1 coklat coklat  1289 May  5 21:20 itungwoi.c
```



```
netlab>> bacadong initutam9.txt
aslabnya bang st
netlab>>
```

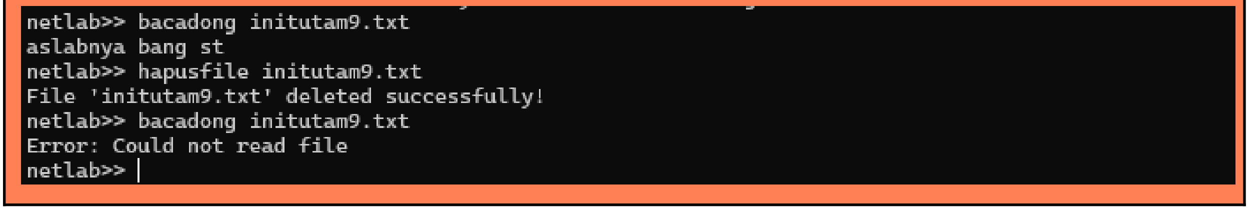
f. Menghapus file

hapusfile.c

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    if (argc != 2) {
        printf("Usage: hapusfile
<filename>\n");
        return 1;
    }

    if (remove(argv[1]) == 0)
        printf("File '%s'
deleted successfully!\n",
argv[1]);
    else
        perror("Error deleting
file");

    return 0;
}
```

Code di shell

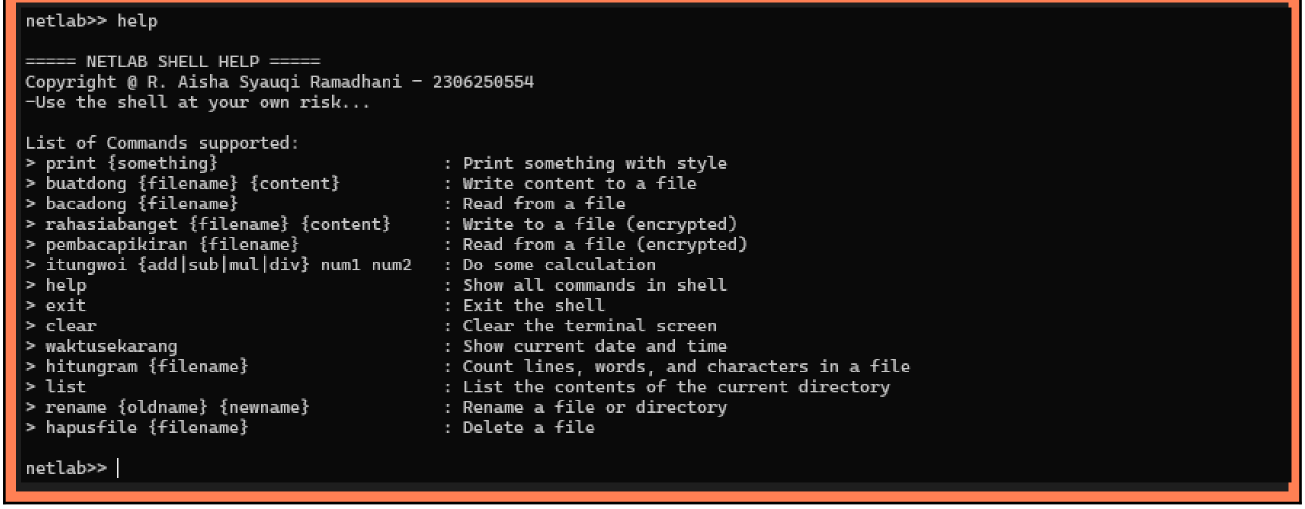
```
else if (strcmp(token, "hapusfile") == 0)
        {
            char *filename = strtok(NULL, " ");

            if (filename != NULL)
            {
                pid_t pid = fork();
                if (pid == 0)
                {

                    perror("exec failed");
                    exit(1);
                }
                wait(NULL);
            }
            else
            {
                printf("Usage: hapusfile {filename}\n");
            }
        }
```



```
netlab>> bacadong initutam9.txt
aslabnya bang st
netlab>> hapusfile initutam9.txt
File 'initutam9.txt' deleted successfully!
netlab>> bacadong initutam9.txt
Error: Could not read file
netlab>> |
```

2. Menampilkan help update

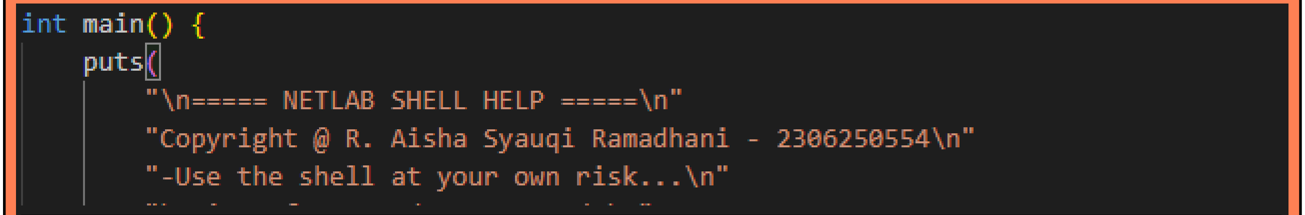
printHelp.c



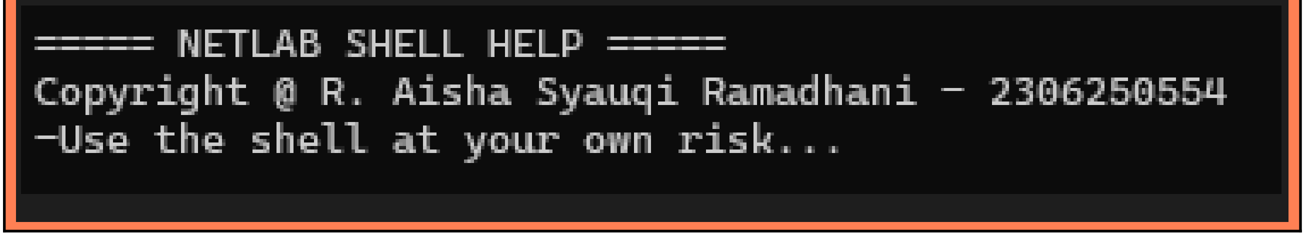
```
netlab>> help

===== NETLAB SHELL HELP =====
Copyright @ R. Aisha Syauqi Ramadhani - 2306250554
-Use the shell at your own risk...

List of Commands supported:
> print {something}                      : Print something with style
> buatdong {filename} {content}          : Write content to a file
> bacadong {filename}                    : Read from a file
> rahasiabanget {filename} {content}     : Write to a file (encrypted)
> pembacapikiran {filename}              : Read from a file (encrypted)
> itungwoi {add|sub|mul|div} num1 num2   : Do some calculation
> help                                   : Show all commands in shell
> exit                                   : Exit the shell
> clear                                  : Clear the terminal screen
> waktusekarang                          : Show current date and time
> hitungram {filename}                   : Count lines, words, and characters in a file
> list                                   : List the contents of the current directory
> rename {oldname} {newname}             : Rename a file or directory
> hapusfile {filename}                   : Delete a file

netlab>> |
```

3. Mengubah copyright

printHelp.c



```c
int main() {
    puts(
        "\n===== NETLAB SHELL HELP =====\n"
        "Copyright @ R. Aisha Syauqi Ramadhani - 2306250554\n"
        "-Use the shell at your own risk...\n"
```



```
===== NETLAB SHELL HELP =====
Copyright @ R. Aisha Syauqi Ramadhani - 2306250554
-Use the shell at your own risk...
```