

## PRAKTIKUM SISTEM OPERASI

<b>Nama</b>	R. Aisha Syauqi Ramadhani	<b>No. Modul</b>	10
<b>NPM</b>	2306250554	<b>Tipe</b>	CS

<https://github.com/rdnaishaa/shell-os>

```
coklat@DESKTOP-EV1050I:/mnt/c/Users/USER/Documents/SEM4/SistemOperasi/Praktikum/Modul 9-10/cs$ ./shell
```



```
Copyright @ R. Aisha Syauqi Ramadhani - 2306250554

Type "help" to see the list of commands
Type "exit" to quit the shell
Type "clear" to clear the screen
Type "{Command} -h" to see the help for a command
Type "Command1 | Command2" to pipe output from Command1 to Command2
Type "{Command} | print" to see the formatted output
```

### Fitur Utama:

#### 1. Flag untuk print.c

print.c	kode
	<pre>netlab&gt;&gt; print -h Usage: print [OPTION]... [TEXT] Options:   -u          Print text in uppercase   -l          Print text in lowercase   -r          Print text in reverse   -h          Show this help message netlab&gt;&gt; print -u halow ini diprint: HALOW netlab&gt;&gt; print -l TUTAMOS ini diprint: tutamos netlab&gt;&gt; print -r modul10 ini diprint: 01ludom</pre>

## 2. Flag help dan pipe untuk all commands

### Mengupdate printhelp:

```
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
  -h, --help                  : Show print command help
  -u                           : Convert text to UPPERCASE
  -l                           : Convert text to lowercase
  -r                           : Print text in reverse
> buatdong {filename} {content} : Write content to a file
  -h, --help                  : Show buatdong command help
> bacadong {filename}          : Read from a file
  -h, --help                  : Show bacadong command help
> rahasiabanget {filename} {content} : Write to a file (encrypted)
  -h, --help                  : Show rahasiabanget command help
> pembacapistiran {filename}    : Read from a file (encrypted)
  -h, --help                  : Show pembacapistiran command help
> itungwoi {add|sub|mul|div} num1 num2 : Do some calculation
  -h, --help                  : Show itungwoi command help
> help                         : Show all commands in shell
> exit                         : Exit the shell
> clear                        : Clear the terminal screen
> waktusekarang                : Show current date and time
  -h, --help                  : Show waktusekarang command help
> hitungram {filename}         : Count lines, words, and characters in a file
  -h, --help                  : Show hitungram command help
> list                         : List the contents of the current directory
> rename {oldname} {newname}   : Rename a file or directory
  -h, --help                  : Show rename command help
> hapusfile {filename}        : Delete a file
  -h, --help                  : Show hapusfile command help

Special Features:
> Pipe (|)                    : Connect the output of first command as input to second command
  Example: itungwoi add 5 3 | print : Will print the result of addition (8.00)
  Example: list | print          : Will print the contents of the directory

General Usage Tips:
> Type "{command} -h" to see the help for a specific command
> All commands support the -h or --help flag for displaying command-specific help
> Use pipe (|) to redirect output from one command to another
```

### Flag dan Pipe:

buatdong.c	
Flag -h / --help	<pre>netlab&gt;&gt; buatdong -h Simple program to write content to a file. Content can be written in uppercase or lowercase. Usage: buatdong {filename} {content} Options:   -h, --help    Show this help message netlab&gt;&gt;</pre>
Pipe	<pre>netlab&gt;&gt; buatdong icak.txt tutams   print ini diprint: Written to icak.txt netlab&gt;&gt;</pre>
bacadong.c	
Flag -h / --help	<pre>netlab&gt;&gt; bacadong -h Simple program to read and display the content of a file. Usage: bacadong {filename} Options:   -h, --help    Show this help message netlab&gt;&gt;</pre>
Pipe	<pre>netlab&gt;&gt; bacadong icar.txt   print ini diprint: tutams netlab&gt;&gt;</pre>
rahasiabanget.c	

Flag -h / --help	<pre>netlab&gt;&gt; rahasiabanget --help Simple Caesar cipher decryption program with a shift of 3 to decode secret messages. Usage: rahasiabanget {filename} {content} Options:   -h, --help    Show this help message netlab&gt;&gt;  </pre>
Pipe	<pre>netlab&gt;&gt; rahasiabanget tutams.txt icakcacar   print ini diprint: Suttu diem aja ya file tutams.txt rahasia</pre>
bacadong	
<pre>netlab&gt;&gt; bacadong tutams.txt   print ini diprint: lfdnfdudu netlab&gt;&gt; bacadong tutams.txt lfdnfdudu</pre>	
pembacapikiran.c	
Flag -h / --help	<pre>netlab&gt;&gt; pembacapikiran --help Simple program to decrypt a Caesar cipher encoded text with a shift of 3, revealing the original message. Usage: pembacapikiran {filename} Options:   -h, --help    Show this help message netlab&gt;&gt;  </pre>
Pipe	<pre>netlab&gt;&gt; pembacapikiran tutams.txt Rahasianya: icakcacar netlab&gt;&gt; pembacapikiran tutams.txt   print ini diprint: Rahasianya: icakcacar netlab&gt;&gt;  </pre>
itungwoi.c	
Flag -h / --help	<pre>netlab&gt;&gt; itungwoi -h Simple calculator to add, subtract, multiply, or divide two numbers. Usage: itungwoi {add sub mul div} num1 num2 Options:   -h, --help    Show this help message</pre>
Pipe	<pre>netlab&gt;&gt; itungwoi add 3 7 10.00 netlab&gt;&gt; itungwoi mul 3 7   print ini diprint: 21.00 netlab&gt;&gt;  </pre>
Fitur Tambahan modul 9	
<pre>netlab&gt;&gt; waktusekarang Current time: 2025-05-13 14:46:15 netlab&gt;&gt; hitunggram tutams.txt Lines: 0 Words: 0 Characters: 9 netlab&gt;&gt; rename tutams.txt tutamos10.txt File renamed successfully. netlab&gt;&gt; bacadong tutams.txt Error opening file: No such file or directory netlab&gt;&gt; bacadong tutamos10.txt lfdnfdudu netlab&gt;&gt; pembacapikiran tutamos10.txt Rahasianya: icakcacar netlab&gt;&gt; hapusfile tutamos10.txt File 'tutamos10.txt' deleted successfully! netlab&gt;&gt; pembacapikiran tutamos10.txt Failed to open the file: No such file or directory netlab&gt;&gt;  </pre>	

### 3. Kode pipe di main, shell.c

#### Shell.c kode

```
void parseInput(char *input)
{
    if (strcmp(input, "exit") == 0)
        exit(0);
    else if (strlen(input) != 0)
    {
        add_history(input);

        // Salin input asli untuk digunakan
        char input_copy[1024];
        strcpy(input_copy, input);

        // Memeriksa apakah ada pipe dalam perintah
        if (strchr(input, '|') != NULL) {
            // Split command by pipe
            char *cmd1 = strtok(input_copy, "|");
            char *cmd2 = strtok(NULL, "\n");

            if (cmd1 == NULL || cmd2 == NULL) {
                printf("Invalid pipe command\n");
                return;
            }

            // Trim whitespace dari kedua command
            while (*cmd1 == ' ') cmd1++;
            while (*cmd2 == ' ') cmd2++;

            // Hapus whitespace di akhir cmd1
            char *end = cmd1 + strlen(cmd1) - 1;
            while (end > cmd1 && *end == ' ') end--;
            *(end + 1) = '\0';

            int pipefd[2];
            pid_t pid1, pid2;

            if (pipe(pipefd) == -1) {
                perror("pipe");
                return;
            }

            pid1 = fork();
            if (pid1 == 0) {
                // CHILD 1 (left command)
                dup2(pipefd[1], STDOUT_FILENO); // Redirect stdout ke pipe
                close(pipefd[0]); // Tutup read end yang tidak digunakan
                close(pipefd[1]); // Tutup setelah dup2

                // Parsing command pertama
                char cmd1_cpy[512];
                strcpy(cmd1_cpy, cmd1);

                char *args1[20] = {NULL};
                int i = 0;

                char *token = strtok(cmd1_cpy, " \t");
                if (token == NULL) {
                    fprintf(stderr, "Invalid command\n");
                    exit(1);
                }

                // Penanganan khusus untuk perintah bawaan
                if (strcmp(token, "itungwoi") == 0) {
                    args1[0] = "./itungwoi";
                } else if (strcmp(token, "print") == 0) {
                    args1[0] = "./print";
                } else if (strcmp(token, "buatdong") == 0) {
                    args1[0] = "./buatdong";
                } else if (strcmp(token, "bacadong") == 0) {
                    args1[0] = "./bacadong";
                } else if (strcmp(token, "rahasiabanget") == 0) {
                    args1[0] = "./rahasiabanget";
                } else if (strcmp(token, "pembacapikiran") == 0) {
                    args1[0] = "./pembacapikiran";
                }
            }
        }
    }
}
```

```

    } else if (strcmp(token, "waktusekarang") == 0) {
        args1[0] = "./waktusekarang";
    } else if (strcmp(token, "hitungram") == 0) {
        args1[0] = "./hitungram";
    } else if (strcmp(token, "rename") == 0) {
        args1[0] = "./rename";
    } else if (strcmp(token, "hapusfile") == 0) {
        args1[0] = "./hapusfile";
    } else if (strcmp(token, "list") == 0) {
        args1[0] = "/bin/ls";
    } else if (strcmp(token, "clear") == 0) {
        args1[0] = "/bin/clear";
    } else {
        args1[0] = token; // perintah yang tidak dikenali
    }

    i = 1;
    while ((token = strtok(NULL, " \t")) != NULL && i < 19) {
        args1[i++] = token;
    }
    args1[i] = NULL;

    execvp(args1[0], args1);
    perror("exec failed");
    exit(1);
}

pid2 = fork();
if (pid2 == 0) {
    // CHILD 2 (right command)
    dup2(pipefd[0], STDIN_FILENO); // Redirect stdin dari pipe
    close(pipefd[1]); // Tutup write end yang tidak digunakan
    close(pipefd[0]); // Tutup setelah dup2

    // Parsing command kedua
    char cmd2_cpy[512];
    strcpy(cmd2_cpy, cmd2);

    char *args2[20] = {NULL};
    int j = 0;

    char *token = strtok(cmd2_cpy, " \t");
    if (token == NULL) {
        fprintf(stderr, "Invalid command\n");
        exit(1);
    }

    // Penanganan khusus untuk perintah bawaan
    if (strcmp(token, "itungwoi") == 0) {
        args2[0] = "./itungwoi";
    } else if (strcmp(token, "print") == 0) {
        args2[0] = "./print";
    } else if (strcmp(token, "buatdong") == 0) {
        args2[0] = "./buatdong";
    } else if (strcmp(token, "bacadong") == 0) {
        args2[0] = "./bacadong";
    } else if (strcmp(token, "rahasiabanget") == 0) {
        args2[0] = "./rahasiabanget";
    } else if (strcmp(token, "pembacapikiran") == 0) {
        args2[0] = "./pembacapikiran";
    } else if (strcmp(token, "waktusekarang") == 0) {
        args2[0] = "./waktusekarang";
    } else if (strcmp(token, "hitungram") == 0) {
        args2[0] = "./hitungram";
    } else if (strcmp(token, "rename") == 0) {
        args2[0] = "./rename";
    } else if (strcmp(token, "hapusfile") == 0) {
        args2[0] = "./hapusfile";
    } else if (strcmp(token, "list") == 0) {
        args2[0] = "/bin/ls";
    } else if (strcmp(token, "clear") == 0) {
        args2[0] = "/bin/clear";
    } else {
        args2[0] = token; // perintah yang tidak dikenali
    }

    j = 1;
    while ((token = strtok(NULL, " \t")) != NULL && j < 19) {
        args2[j++] = token;
    }
}

```

```

    }
    args2[j] = NULL;

    execvp(args2[0], args2);
    perror("exec failed");
    exit(1);
}

// Parent process
close(pipefd[0]);
close(pipefd[1]);
waitpid(pid1, NULL, 0);
waitpid(pid2, NULL, 0);
return;
}

```

## Fitur Tambahan:

### Update printhelp fitur tambahan

```

> caridong {pattern} [directory]      : Searches for files containing a specific pattern
  -i                                  : Case insensitive search
  -r                                  : Recursive search in subdirectories
  -n                                  : Show line numbers with matches
  -c                                  : Count matches only
> sortirdong {filename} [column]      : Sort file contents
  -n                                  : Sort numerically
  -r                                  : Sort in reverse order
  -u                                  : Remove duplicates
  -f                                  : Case insensitive sorting
> proseswoi                          : List or manage running processes
  -a                                  : Show all processes
  -k {pid}                           : Kill process by PID
  -m                                  : Sort by memory usage
  -c                                  : Sort by CPU usage
> netwoi [interface]                 : Display network information
  -a                                  : Show all interfaces
  -i                                  : Show IP addresses only
  -s                                  : Show connection statistics
  -m                                  : Show MAC addresses
> statsdong {filename}                : Perform statistical analysis on data file
  -a                                  : Calculate mean, median, mode
  -g                                  : Generate graph of data distribution
  -c {column}                         : Select column for analysis
  -n                                  : Normalize data (0-1 range)

Special Features:
> Pipe (|)                           : Connect the output of first command as input to second command
  Example: itungwoi add 5 3 | print    : Will print the result of addition (8.00)
  Example: list | print                : Will print the contents of the directory
  Example: caridong "text" | sortirdong -u : Find text and display unique results
  Example: bacadong file.txt | statsdong -a : Read file and calculate statistics
  Example: netwoi -i | print -u        : Display IP addresses in uppercase

General Usage Tips:
> Type "{Command} -h" to see the help for a specific command
> All commands support the -h or --help flag for displaying command-specific help
> Use pipe (|) to redirect output from one command to another
> Multiple flags can be combined (e.g., 'caridong -ir pattern')

```

## 1. 5 new commands

caridong.c	
Flag -h / -help	<pre> netlab&gt;&gt; caridong -h Usage: caridong {pattern} [directory] Search for files containing a specific pattern  Flags:   -i: Case insensitive search   -r: Recursive search in subdirectories   -n: Show line numbers with matches   -c: Count matches only  Multiple flags can be combined (e.g., caridong -ir pattern)  Examples: caridong "hello" .: Search for 'hello' in current directory caridong -i "Hello" /tmp: Case insensitive search for 'Hello' in /tmp caridong -rn "function" src: Recursive search with line numbers caridong -c "error" logs: Count number of matches only caridong "text"   sortirdong -u: Find text and pipe to sortirdong </pre>
Pipe	<pre> netlab&gt;&gt; caridong hello   print ini diprint: ./caridong.c: printf(" caridong \"hello\" : Search for 'hello' in current directory\n"); netlab&gt;&gt; caridong -c error   print ini diprint: ./bacadong.c: 1 matches </pre>
<pre> netlab&gt;&gt; caridong -c print ./bacadong.c: 8 matches ./buatdong.c: 9 matches ./caridong.c: 35 matches ./hapusfile.c: 2 matches ./help: 5 matches ./hitunggram.c: 2 matches ./itungwoi.c: 13 matches ./netwoi.c: 34 matches ./pembacapikiran.c: 8 matches ./print.c: 21 matches ./printHelp.c: 5 matches ./proseswoi.c: 29 matches ./rahasiabangget.c: 9 matches ./rename.c: 2 matches ./shell: 1 matches ./shell.c: 19 matches ./sortirdong.c: 27 matches ./statsdong.c: 55 matches ./waktusekarang.c: 1 matches netlab&gt;&gt; caridong 2306250554 ./help: Copyright @ R. Aisha Syauqi Ramadhani - 2306250554 ./printHelp.c: "Copyright @ R. Aisha Syauqi Ramadhani - 2306250554\n" ./shell: Copyright @ R. Aisha Syauqi Ramadhani - 2306250554 ./shell.c: "Copyright @ R. Aisha Syauqi Ramadhani - 2306250554\n" netlab&gt;&gt; </pre>	
sortirdong.c	

Flag -h / --help	<pre>netlab&gt;&gt; sortirdong --help  ===== SORTIRDONG HELP ===== Usage: sortirdong {filename} [column] Description: Sort file contents  Flags:   -n: Sort numerically   -r: Sort in reverse order   -u: Remove duplicates   -f: Case insensitive sorting   -h, --help: Show this help message  Examples:   sortirdong data.txt: Sort lines in data.txt alphabetically   sortirdong -nr data.csv 2: Sort data.csv numerically by column 2 in reverse order   sortirdong -uf names.txt: Sort names.txt case insensitively and remove duplicates netlab&gt;&gt;  </pre>
Pipe	<pre>netlab&gt;&gt; sortirdong ica.txt   print ini diprint: tutams netlab&gt;&gt;  </pre>
<div> <pre>netlab&gt;&gt; bacadong kanae.txt kanae wifi mati wifi pw tapi wifi jelek</pre> <pre>netlab&gt;&gt; sortirdong kanae.txt jelek kanae mati pw tapi wifi wifi wifi</pre> <pre>netlab&gt;&gt; sortirdong -uf kanae.txt jelek kanae mati pw tapi wifi netlab&gt;&gt;  </pre> </div>	
proseswoi.c	
Flag -h / --help	<pre>netlab&gt;&gt; proseswoi -h  ===== PROSESWOI HELP ===== Usage: proseswoi Description: List or manage running processes  Flags:   -a: Show all processes   -k {pid}: Kill process by PID   -m: Sort by memory usage   -c: Sort by CPU usage   -h, --help: Show this help message  Examples:   proseswoi: Show your processes   proseswoi -a: Show all processes   proseswoi -k 1234: Kill process with PID 1234   proseswoi -am: Show all processes sorted by memory usage netlab&gt;&gt;  </pre>
Pipe	<pre>netlab&gt;&gt; proseswoi -a   print ini diprint: PID      USER      CPU%      MEM(MB)   COMMAND 17364  </pre>



```
netlab>> proseswui -a
PID    USER    CPU%    MEM(MB)  COMMAND
1      root     15.3    21.1     /sbin/init
2      root     1.3     2.7      /init
7      root     0.0     2.7      plan9
55     root     0.4     65.2     /usr/lib/systemd/systemd-journald
78     root     0.9     23.4     /usr/lib/systemd/systemd-udev
141    systemd-resolve 0.2    20.9     /usr/lib/systemd/systemd-resolved
142    systemd-timesync 0.1    88.9     /usr/lib/systemd/systemd-timesyncd
151    root     0.2     4.1      /usr/sbin/cron
152    messagebus 0.2    9.3      @dbus-daemon
168    root     0.2     17.6     /usr/lib/systemd/systemd-logind
171    root     0.2     1714.9   /usr/libexec/wsl-pro-service
176    root     0.0     3.1      /sbin/agetty
189    syslog   0.2     217.3    /usr/sbin/rsyslogd
193    root     0.0     3.0      /sbin/agetty
204    root     0.3     104.5    /usr/bin/python3
299    root     0.0     2.7      /init
300    root     0.1     2.7      /init
301    coklat   7.8     6.2      -bash
302    root     0.0     6.7      /bin/login
390    coklat   0.2     19.8     /usr/lib/systemd/systemd
391    coklat   0.0     20.7     (sd-pam)
404    coklat   0.0     6.0      -bash
1087   coklat   0.7     3.7      ./shell
1159   coklat   0.0     3.1      ./proseswui

netlab>> proseswui -m
PID    USER    CPU%    MEM(MB)  COMMAND
390    coklat   0.2     19.8     /usr/lib/systemd/systemd
391    coklat   0.0     20.7     (sd-pam)
301    coklat   7.8     6.2      -bash
404    coklat   0.0     6.0      -bash
1087   coklat   0.7     3.7      ./shell
1160   coklat   0.0     3.1      ./proseswui

netlab>> proseswui -c
PID    USER    CPU%    MEM(MB)  COMMAND
301    coklat   7.8     6.2      -bash
390    coklat   0.2     19.8     /usr/lib/systemd/systemd
391    coklat   0.0     20.7     (sd-pam)
404    coklat   0.0     6.0      -bash
1087   coklat   0.7     3.7      ./shell
1161   coklat   0.0     3.1      ./proseswui
```

## netwoi.c

Flag -h / -help

```
netlab>> netwoi -h

===== NETWOI HELP =====
Usage: netwoi [interface]
Description: Display network information

Flags:
  -a: Show all interfaces
  -i: Show IP addresses only
  -s: Show connection statistics
  -m: Show MAC addresses
  -h, --help: Show this help message

Examples:
  netwoi: Show main interface information
  netwoi eth0: Show information for eth0
  netwoi -a: Show all interfaces
  netwoi -im: Show IP and MAC addresses for all interfaces
```

Pipe

```
netlab>> netwoi -i | print
ini diprint: lo: 127.0.0.1

netlab>> netwoi -i | print -u
ini diprint: LO: 127.0.0.1
```

```
netlab>> netwoi -a

--- lo ---
Status: DOWN
IP Address: 127.0.0.1 (IPv4)

--- eth0 ---
Status: UP
IP Address: 172.25.52.120 (IPv4)
netlab>> netwoi -s

--- lo ---
Status: DOWN
IP Address: 127.0.0.1 (IPv4)
Received: 7.53 KB
Transmitted: 7.53 KB

--- eth0 ---
Status: UP
IP Address: 172.25.52.120 (IPv4)
Received: 57.23 KB
Transmitted: 4.22 KB
```

## statsdong.c

### Flag -h / -help

```
netlab>> statsdong -h

===== STATSDONG HELP =====
Usage: statsdong {filename}
Description: Perform statistical analysis on data file

Flags:
  -a: Calculate mean, median, mode
  -g: Generate graph of data distribution
  -c {column}: Select column for analysis
  -n: Normalize data (0-1 range)
  -h, --help: Show this help message

Examples:
  statsdong data.txt: Basic stats for data.txt
  statsdong -a data.csv: Calculate all statistics
  statsdong -c 2 grades.csv: Analyze column 2 of grades.csv
  statsdong -agn temperature.txt: Full analysis with graph and normalization
```

```
netlab>> bacadong cobaitung.txt
7
6
5
4
3
2
1
netlab>> |
```

```
netlab>> statsdong cobaitung.txt

===== Statistics for cobaitung.txt =====
Count: 7
Min: 1.0000
Max: 7.0000
Range: 6.0000
Sum: 28.0000
Mean: 4.0000
```

```
netlab>> statsdong -a cobaitung.txt

===== Statistics for cobaitung.txt =====
Count: 7
Min: 1.0000
Max: 7.0000
Range: 6.0000
Sum: 28.0000
Mean: 4.0000
Median: 4.0000
Mode: 1.0000 (occurs 1 times)
Standard Deviation: 2.0000
Variance: 4.0000
```