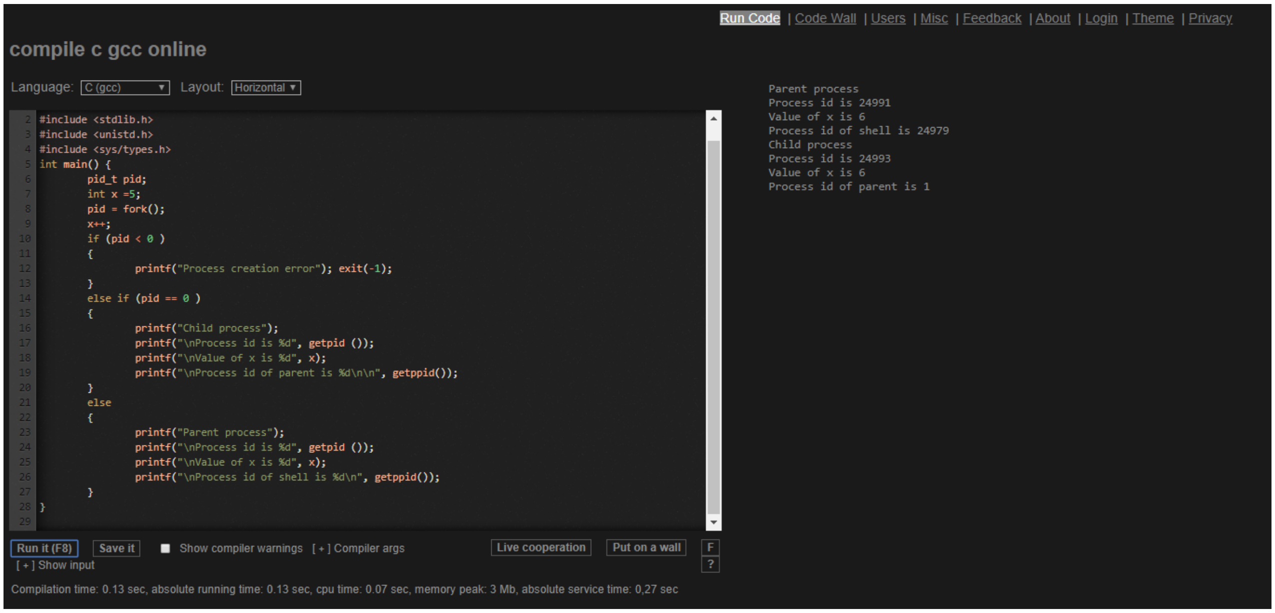
Praktikum Sistem Operasi

MODUL 8

Nama : Edi Supriyanto

NIM : L200180002

Fork.c



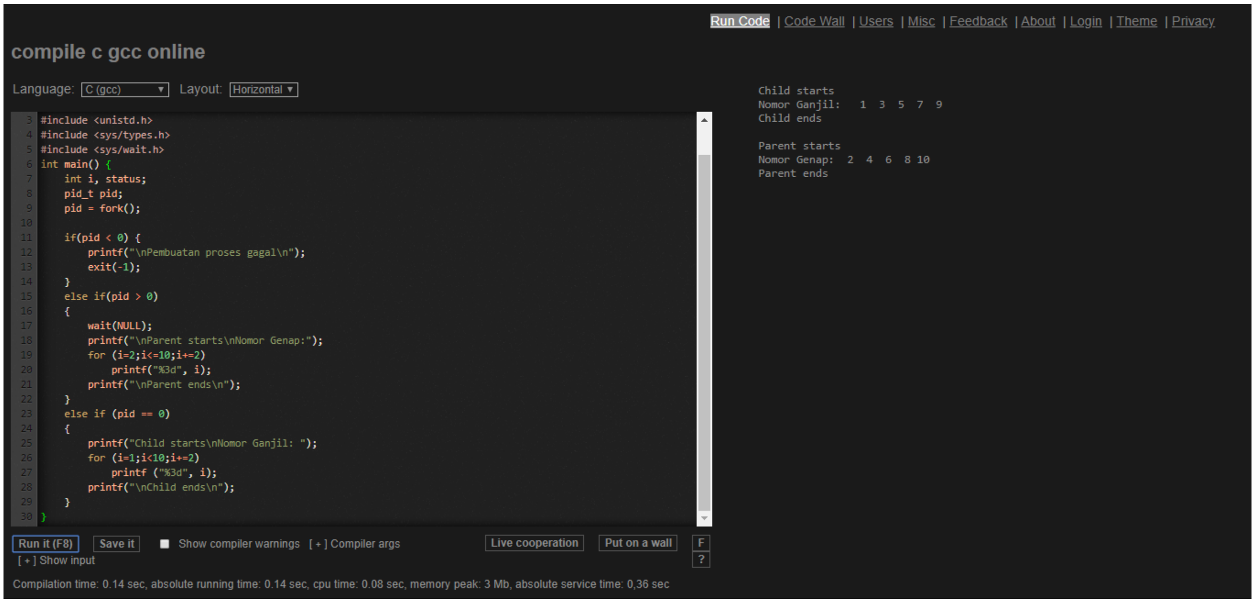
Language: C (gcc) Layout: Horizontal

```c
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
int main() {
        pid_t pid;
        int x =5;
        pid = fork();
        x++;
        if (pid < 0 )
        {
                printf("Process creation error"); exit(-1);
        }
        else if (pid == 0 )
        {
                printf("Child process");
                printf("\nProcess id is %d", getpid ());
                printf("\nValue of x is %d", x);
                printf("\nProcess id of parent is %d\n\n", getppid());
        }
        else
        {
                printf("Parent process");
                printf("\nProcess id is %d", getpid ());
                printf("\nValue of x is %d", x);
                printf("\nProcess id of shell is %d\n", getppid());
        }
}
```

```
Parent process
Process id is 24991
Value of x is 6
Process id of shell is 24979
Child process
Process id is 24993
Value of x is 6
Process id of parent is 1
```

Compilation time: 0.13 sec, absolute running time: 0.13 sec, cpu time: 0.07 sec, memory peak: 3 Mb, absolute service time: 0,27 sec

Wait.c



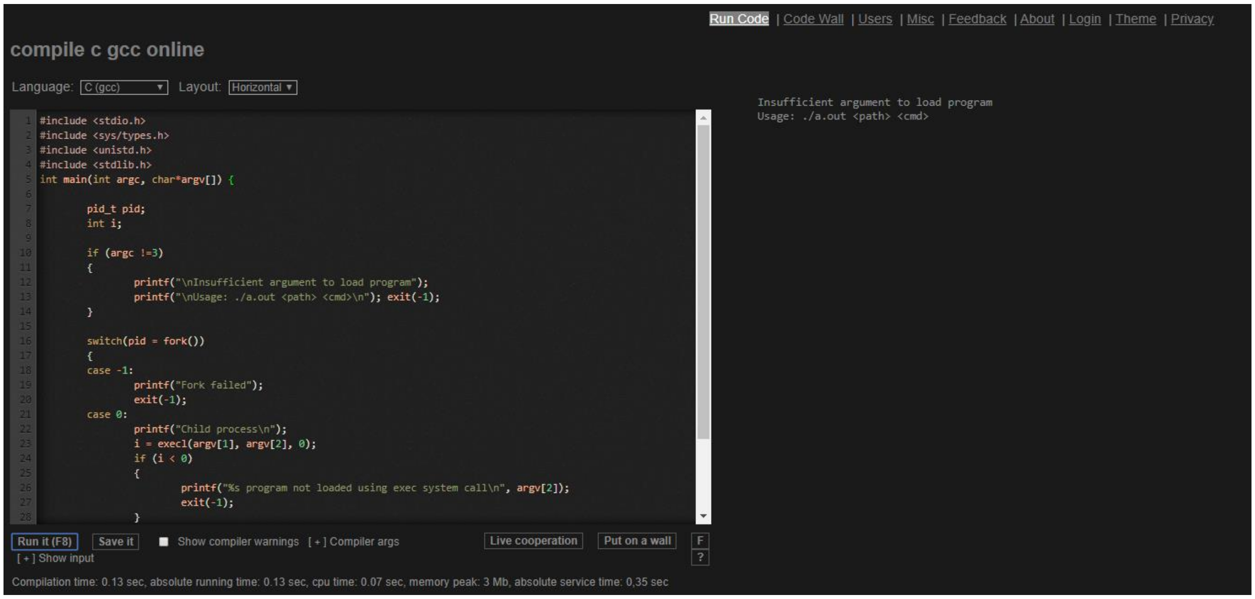
Language: C (gcc) Layout: Horizontal

```c
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
int main() {
    int i, status;
    pid_t pid;
    pid = fork();

    if(pid < 0) {
        printf("\nPembuatan proses gagal\n");
        exit(-1);
    }
    else if(pid > 0)
    {
        wait(NULL);
        printf("\nParent starts\nNomor Genap:");
        for (i=2;i<=10;i+=2)
            printf("%3d", i);
        printf("\nParent ends\n");
    }
    else if (pid == 0)
    {
        printf("Child starts\nNomor Ganjil: ");
        for (i=1;i<10;i+=2)
            printf ("%3d", i);
        printf("\nChild ends\n");
    }
}
```

```
Child starts
Nomor Ganjil:   1  3  5  7  9
Child ends

Parent starts
Nomor Genap:  2  4  6  8 10
Parent ends
```

Compilation time: 0.14 sec, absolute running time: 0.14 sec, cpu time: 0.08 sec, memory peak: 3 Mb, absolute service time: 0,36 sec

Exec.c



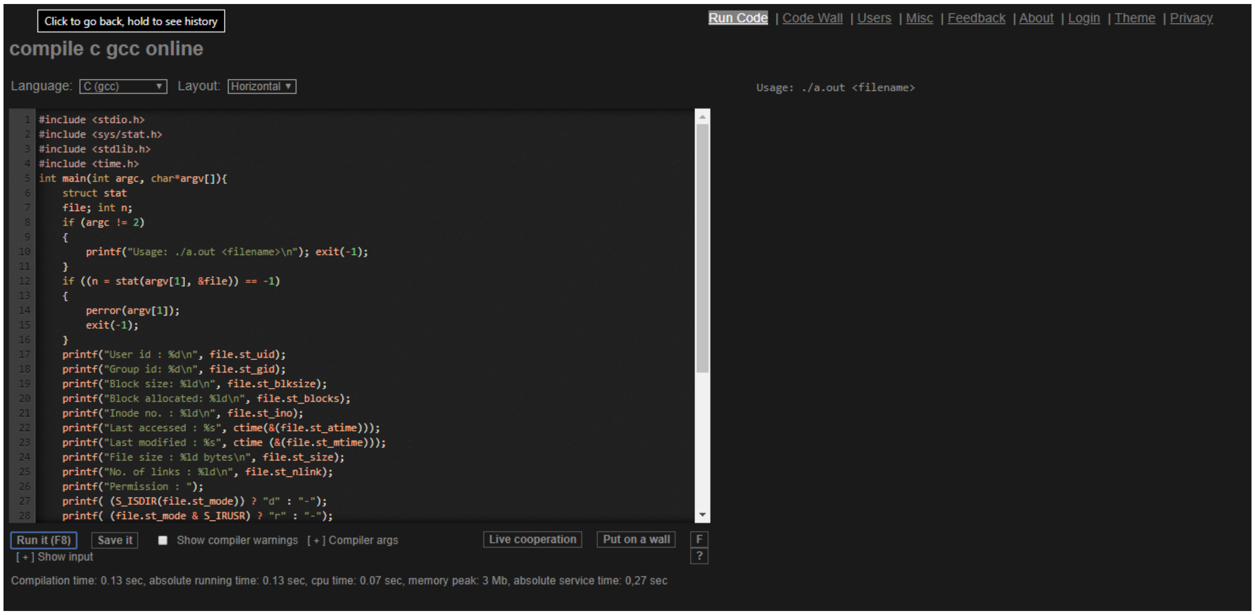
Language: C (gcc) Layout: Horizontal

```c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>
int main(int argc, char*argv[]) {

        pid_t pid;
        int i;

        if (argc !=3)
        {
                printf("\nInsufficient argument to load program");
                printf("\nUsage: ./a.out <path> <cmd>\n"); exit(-1);
        }

        switch(pid = fork())
        {
        case -1:
                printf("Fork failed");
                exit(-1);
        case 0:
                printf("Child process\n");
                i = execl(argv[1], argv[2], 0);
                if (i < 0)
                {
                        printf("%s program not loaded using exec system call\n", argv[2]);
                        exit(-1);
                }
```

```
Insufficient argument to load program
Usage: ./a.out <path> <cmd>
```

Compilation time: 0.13 sec, absolute running time: 0.13 sec, cpu time: 0.07 sec, memory peak: 3 Mb, absolute service time: 0,35 sec

Stat.c



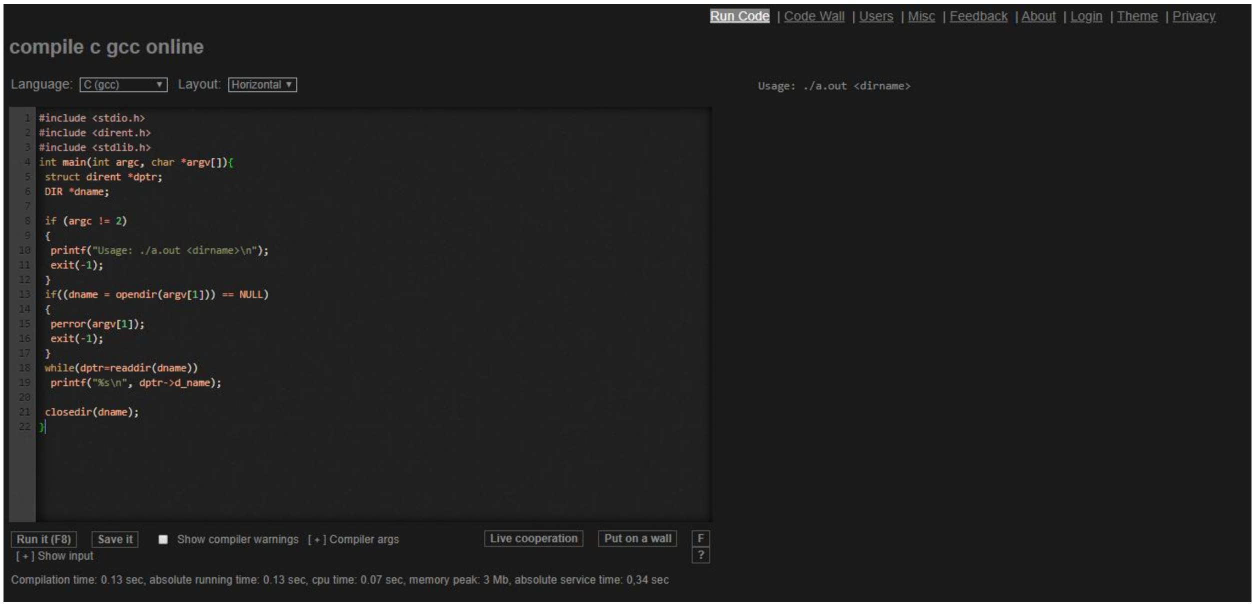
Language: C (gcc) Layout: Horizontal

```c
#include <stdio.h>
#include <sys/stat.h>
#include <stdlib.h>
#include <time.h>
int main(int argc, char*argv[]){
    struct stat
    file; int n;
    if (argc != 2)
    {
        printf("Usage: ./a.out <filename>\n"); exit(-1);
    }
    if ((n = stat(argv[1], &file)) == -1)
    {
        perror(argv[1]);
        exit(-1);
    }
    printf("User id : %d\n", file.st_uid);
    printf("Group id: %d\n", file.st_gid);
    printf("Block size: %ld\n", file.st_blksize);
    printf("Block allocated: %ld\n", file.st_blocks);
    printf("Inode no. : %ld\n", file.st_ino);
    printf("Last accessed : %s", ctime(&(file.st_atime)));
    printf("Last modified : %s", ctime (&(file.st_mtime)));
    printf("File size : %ld bytes\n", file.st_size);
    printf("No. of links : %ld\n", file.st_nlink);
    printf("Permission : ");
    printf( (S_ISDIR(file.st_mode)) ? "d" : "-");
    printf( (file.st_mode & S_IRUSR) ? "r" : "-");
```

```
Usage: ./a.out <filename>
```

Compilation time: 0.13 sec, absolute running time: 0.13 sec, cpu time: 0.07 sec, memory peak: 3 Mb, absolute service time: 0,27 sec

Dirlist.c



Language: C (gcc) Layout: Horizontal

```
#include <stdio.h>
#include <dirent.h>
#include <stdlib.h>
int main(int argc, char *argv[]){
 struct dirent *dptr;
 DIR *dname;

 if (argc != 2)
 {
  printf("Usage: ./a.out <dirname>\n");
  exit(-1);
 }
 if((dname = opendir(argv[1])) == NULL)
 {
  perror(argv[1]);
  exit(-1);
 }
 while(dptr=readdir(dname))
  printf("%s\n", dptr->d_name);

 closedir(dname);
}
```

```
Usage: ./a.out <dirname>
```

Compilation time: 0.13 sec, absolute running time: 0.13 sec, cpu time: 0.07 sec, memory peak: 3 Mb, absolute service time: 0,34 sec