

# Manual Sesi Lab 2

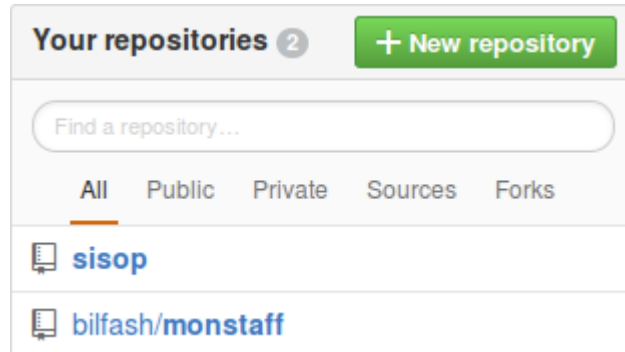
## Sistem Operasi (E)



Kelompok E14:  
Rani Aulia H 5114100044  
Luqman Ahmad 5114100187

## Git

Git merupakan *version control system* yang berguna untuk melihat alur pengerjaan program dari awal hingga berkembang. Klik *new* untuk membuat *new repository*. Buatlah *file ignore* untuk mencegah *file-file* tertentu untuk diunggah ke repositori.



Jika menggunakan *proxy* ITS, ubahlah proxy di terminal dengan mengetikkan

*export*

*https\_proxy=http://emailAnda%40mhs.if.its.ac.id:passwordAnda@proxy.its.ac.id:8080/*

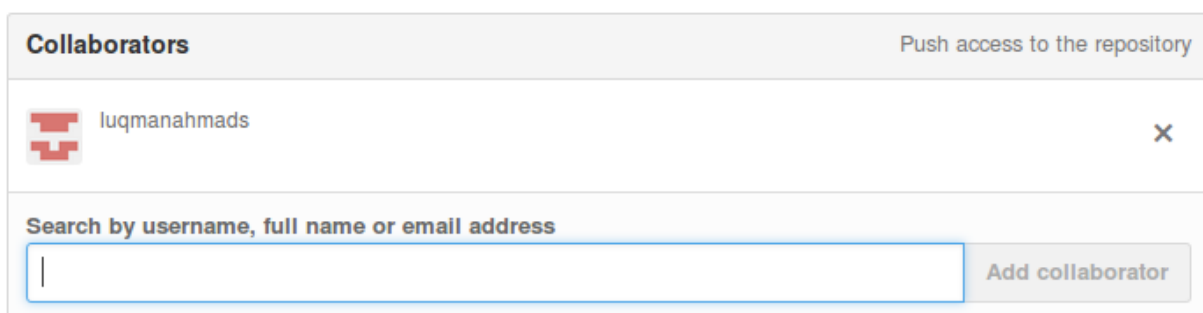
*export*

*http\_proxy=http://emailAnda%40mhs.if.its.ac.id:passwordAnda@proxy.its.ac.id:8080/*

contoh salah satu export

```
ranitera@ranitera-X450LCP:~/sisop$ export http_proxy=http://emailAnda%40mhs.if.its.ac.id:passwordAnda@proxy.its.ac.id:8080
```

Untuk menambahkan orang yang dapat bekerja pada repositori tersebut, pilih *setting* dan klik *add collaborators* setelah mengetikkan *username* dari orang yang akan ditambahkan.



Berikut adalah perintah-perintah yang akan dilakukan dengan menggunakan Git.

- *git add namafile*
- *git status*
- *git commit -m "komentar"*
- *git config*

- `git config --global user.email "email"`
- `git config --global user.name "username"`
- `git push`
- `git --version`
- `git rm namafile`
- `git pull`

Jika Anda ingin meng-*clone repository* yang ada pada suatu, ketikkan perintah berikut pada terminal Anda.

*git clone https\_clone\_url*

Anda bisa mendapatkan *https clone url* dengan menyalinnya dari repositori tersebut.

```
ranitera@ranitera-X450LCP:~/sisop$ git clone https://github.com/ranitera/sisop.git
```

## String Processing

Pada *string processing* ini, kita akan memasukkan sebuah kalimat dan memecahnya menjadi beberapa bagian.

- `strstr`

Ketikkan *man strstr* pada terminal untuk mengetahui isi manual dari `strstr`.

```
STRSTR(3)                                Linux Programmer's Manual                                STRSTR(3)

NAME
    strstr, strcasestr - locate a substring

SYNOPSIS
    #include <string.h>

    char *strstr(const char *haystack, const char *needle);

    #define _GNU_SOURCE                    /* See feature_test_macros(7) */

    #include <string.h>

    char *strcasestr(const char *haystack, const char *needle);

DESCRIPTION
    The strstr() function finds the first occurrence of the substring nee-
    Manual page strstr(3) line 1 (press h for help or q to quit)
```

`strstr` berfungsi untuk mencari kemunculan substring pertama yang dicari. Terdapat dua parameter, `haystack` dan `needle`. Perintah ini akan mencetak isi dari karakter yang ditemukan sampai `null`.

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

int main(){
    char input[50];
    printf("Masukkan string: ");
    scanf("%[^\n]", input);

    printf("%s\n", input);
    char *substr = strstr(input, "o");

    printf("Hasil strstr() : %s\n", substr);

    return 0;
}
```

```
ranitera@ranitera-X450LCP: ~  
ranitera@ranitera-X450LCP:~$ ./cobastr  
Masukkan string: Hello World  
Hello World  
Hasil strstr() : o World  
ranitera@ranitera-X450LCP:~$
```

- strtok

Parameter yang ada pada strtok adalah string itu sendiri, dan delim (pemisah). Kita dapat menggunakan looping untuk mencari bagian-bagian tertentu.

```
#include<stdio.h>  
#include<string.h>  
  
int main(){  
    char input[50];  
    printf("Masukkan string: ");  
    scanf("%[^\n]", input);  
  
    char *token = strtok(input, "o");  
  
    while(token!=NULL){  
        printf("Hasil strtok() : %s\n", token);  
        token = strtok(NULL, "o");  
    }  
  
    return 0;  
}
```

```
ranitera@ranitera-X450LCP: ~  
ranitera@ranitera-X450LCP:~$ ./cobatok  
Masukkan string: Hello World  
Hasil strtok() : Hell  
Hasil strtok() : W  
Hasil strtok() : rld  
ranitera@ranitera-X450LCP:~$
```

## Thread

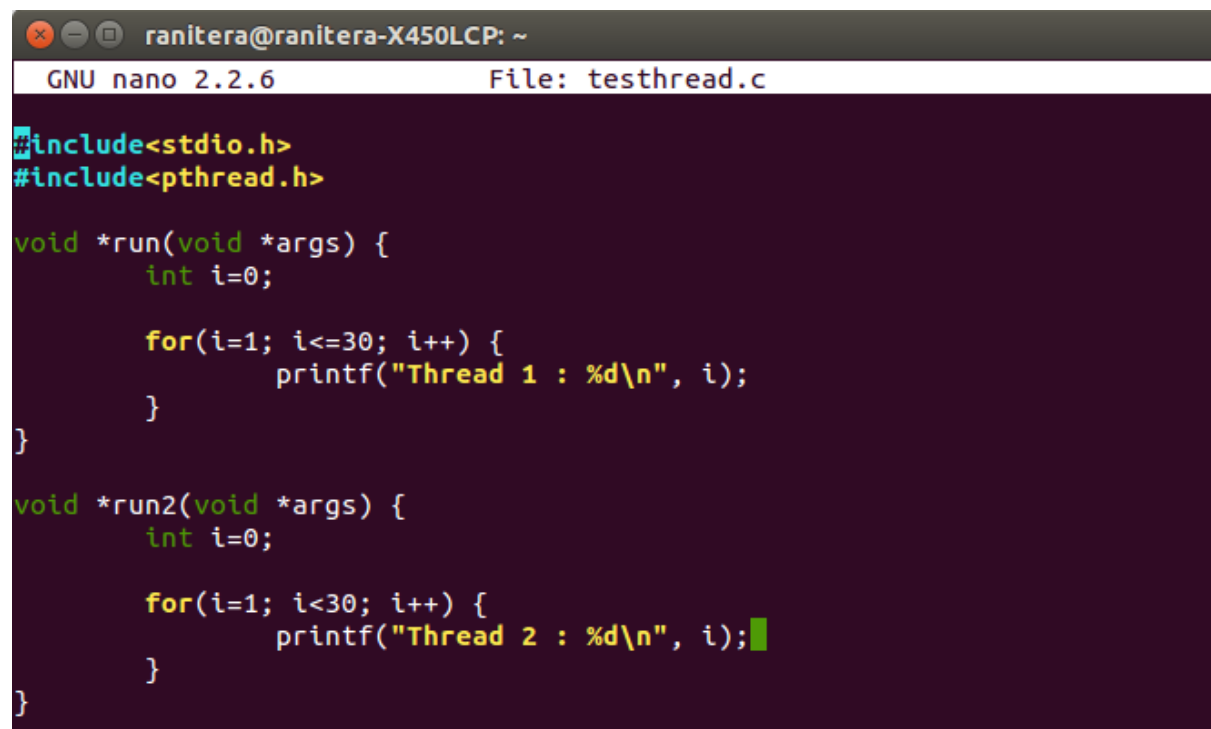
Sebuah thread dibuat agar program dapat memilih fungsi mana yang akan dijalankan. Fungsi tersebut harus mengikuti format tertentu

Gunakan library `#include <pthread.h>`

Pthread berarti POSIX Thread

- `pthread_t` digunakan untuk mendeklarasikan tipe data thread
- `pthread_create` digunakan untuk membuat suatu thread sekaligus langsung menjalankannya, dan membutuhkan beberapa parameter seperti ID Thread, attribute, dan nama fungsi yang akan dijalankan.
- `pthread_join` digunakan untuk menyinkronisasikan semua kerja thread. Defaultnya, saat sebuah thread selesai mengerjakan tugasnya, maka otomatis seluruh proses akan terhenti.

Penggunaan posix thread harus menggunakan return value void.



```
ranitera@ranitera-X450LCP: ~  
GNU nano 2.2.6 File: testthread.c  
  
#include<stdio.h>  
#include<pthread.h>  
  
void *run(void *args) {  
    int i=0;  
  
    for(i=1; i<=30; i++) {  
        printf("Thread 1 : %d\n", i);  
    }  
}  
  
void *run2(void *args) {  
    int i=0;  
  
    for(i=1; i<30; i++) {  
        printf("Thread 2 : %d\n", i);  
    }  
}
```

```

void *run2(void *args) {
    int i=0;

    for(i=1; i<30; i++) {
        printf("Thread 2 : %d\n", i);
    }
}

void main() {
    pthread_t t1, t2;
    pthread_create(&t1, NULL, run, NULL);
    pthread_create(&t2, NULL, run2, NULL);

    int i=0;

    pthread_join(t1, NULL);

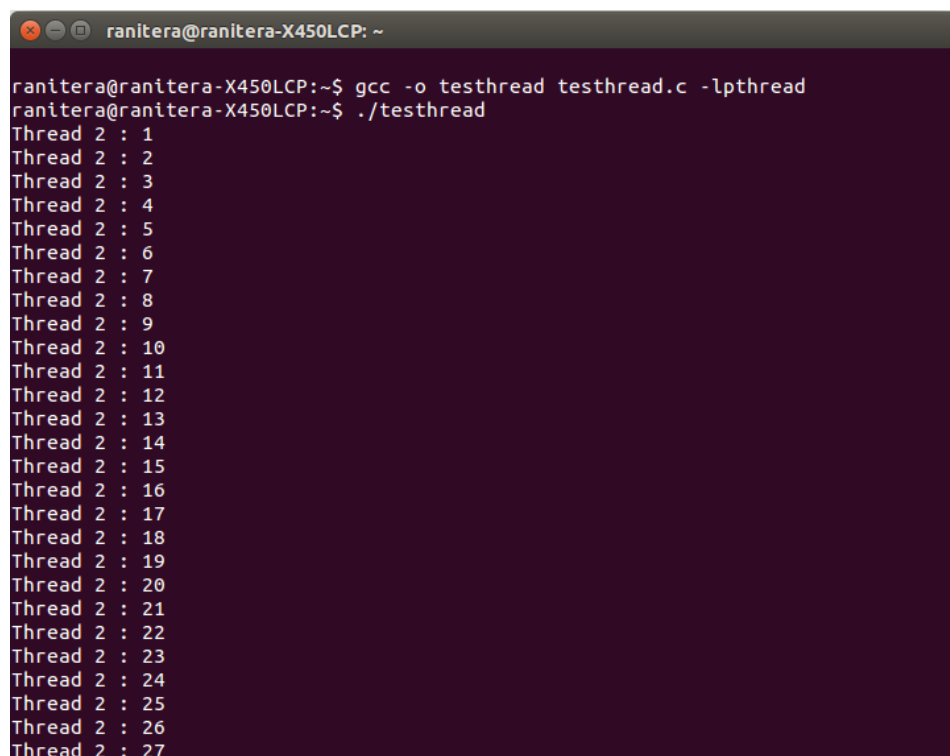
    for(i=1; i<30; i++){
        printf("Main : %d\n", i);
    }

    pthread_join(t2, NULL);
}

```

Program tersebut tidak dapat *dicompile* seperti biasa. Untuk meng-*compile*, ketikkan `gcc -o testthread testthread.c -lpthread`

Jadwal thread akan bergantung pada suatu sistem operasi. Kita tidak dapat memastikan suatu thread untuk berjalan terlebih dahulu, tetapi kita dapat membuat sebuah thread untuk menunggu thread lain dijalankan.



```

ranitera@ranitera-X450LCP: ~
ranitera@ranitera-X450LCP:~$ gcc -o testthread testthread.c -lpthread
ranitera@ranitera-X450LCP:~$ ./testthread
Thread 2 : 1
Thread 2 : 2
Thread 2 : 3
Thread 2 : 4
Thread 2 : 5
Thread 2 : 6
Thread 2 : 7
Thread 2 : 8
Thread 2 : 9
Thread 2 : 10
Thread 2 : 11
Thread 2 : 12
Thread 2 : 13
Thread 2 : 14
Thread 2 : 15
Thread 2 : 16
Thread 2 : 17
Thread 2 : 18
Thread 2 : 19
Thread 2 : 20
Thread 2 : 21
Thread 2 : 22
Thread 2 : 23
Thread 2 : 24
Thread 2 : 25
Thread 2 : 26
Thread 2 : 27

```

```
Thread 2 : 27
Thread 2 : 28
Thread 2 : 29
Thread 1 : 1
Thread 1 : 2
Thread 1 : 3
Thread 1 : 4
Thread 1 : 5
Thread 1 : 6
Thread 1 : 7
Thread 1 : 8
Thread 1 : 9
Thread 1 : 10
Thread 1 : 11
Thread 1 : 12
Thread 1 : 13
Thread 1 : 14
Thread 1 : 15
Thread 1 : 16
Thread 1 : 17
Thread 1 : 18
Thread 1 : 19
Thread 1 : 20
Thread 1 : 21
Thread 1 : 22
Thread 1 : 23
Thread 1 : 24
Thread 1 : 25
Thread 1 : 26
Thread 1 : 27
```

```
Thread 1 : 25
Thread 1 : 26
Thread 1 : 27
Thread 1 : 28
Thread 1 : 29
Thread 1 : 30
Main : 1
Main : 2
Main : 3
Main : 4
Main : 5
Main : 6
Main : 7
Main : 8
Main : 9
Main : 10
Main : 11
Main : 12
Main : 13
Main : 14
Main : 15
Main : 16
Main : 17
Main : 18
Main : 19
Main : 20
Main : 21
Main : 22
Main : 23
Main : 24
```

```
Main : 1
Main : 2
Main : 3
Main : 4
Main : 5
Main : 6
Main : 7
Main : 8
Main : 9
Main : 10
Main : 11
Main : 12
Main : 13
Main : 14
Main : 15
Main : 16
Main : 17
Main : 18
Main : 19
Main : 20
Main : 21
Main : 22
Main : 23
Main : 24
Main : 25
Main : 26
Main : 27
Main : 28
Main : 29
ranitera@ranitera-X450LCP:~$ clear
```