

# BELAJAR LINUX <sup>(2<sup>nd</sup> edition)</sup>

YUK!



# git

“Modul tutorial Git & GitHub ; String Processing ; Forking ;  
Threading”

Oleh :

**Setyassida Novian Putra Damara (5114100024)**

**Jeffry Nasri Faruki (5114100043)**

Dosen Pembimbing :

**Baskoro Adi Pratomo, S. Kom, M. Kom**

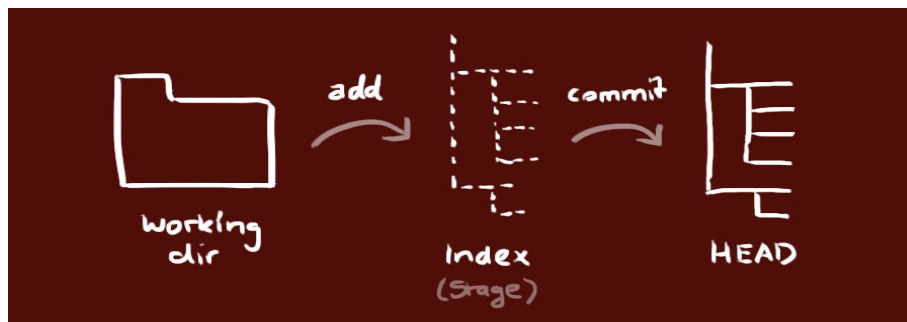
## A) GIT & GITHUB

### ✓ Apa itu Git?

Git adalah tools yang berfungsi sebagai Version Control System (VCS) dan apa itu VCS? VCS adalah sebuah sistem pelacak perubahan pada file. Git sendiri dibuat oleh orang yang menciptakan Kernel Linux, Linus Torvalds. Pada awalnya Git digunakan untuk membantu pengembangan Kernel Linux sebelum akhirnya dipakai lebih luas di berbagai pengembangan software lainnya dan dibuat multi-platform sehingga tersedia di hampir semua OS. Selain Git ada beberapa alternatif lain dari sistem VCS, contohnya: Subversion, Mercurial, CVS, dll.



### ✓ Apa pentingnya Git?



Kenapa Git penting untuk tim developer? Dengan menggunakan Git, setiap orang dalam sebuah tim dapat melakukan perubahan pada source-code tanpa harus takut terjadi bentrok ataupun kesulitan dalam menggabungkan hasil perubahan yang mereka lakukan. Dengan menggunakan Git, setiap perubahan pada source-code akan terlacak pesan perubahannya, apa saja yang diubah, siapa yang mengubah dan kapan waktunya.

### ✓ Apa itu Github?

Github adalah sebuah website yang memberikan pelayanan untuk menyimpan repository (biasa disebut dengan repo saja) anda secara gratis. Banyak perintah yang ada di git bash dan git gui bisa dilakukan melalui Github. Tidak hanya itu, Github juga memudahkan

kolaborasi dalam suatu proyek dengan fitur-fitur tambahan seperti pull request, diskusi di patch, mengatur bugs, dan lain-lain.

Github juga menggabungkan elemen-elemen dari social network ke dalam sistemnya. Untuk urusan hosting Git dan alat kolaborasi, Github adalah website paling populer di dunia maya. Perusahaan seperti Facebook dan Google saja menggunakan Github. Untuk perorangan, nama-nama seperti Linus Torvalds, John Resig (pembuat JQuery), dan DHH (pembuat Rails) pun mempunyai akun di Github.



#### ✓ Langkah-langkah membuat akun Github

- a) Kunjungi halaman web Github di <http://github.com/>.
- b) Pada halaman awal isi form pada kolom sign up, seperti gambar.

c)

The screenshot shows the GitHub website's sign-up page. The browser's address bar shows 'https://github.com'. The page has a dark background with a pattern of small icons. The main heading is 'Where software is built'. Below it, there is a paragraph: 'Powerful collaboration, code review, and code management for open source and private projects. Public projects are always free. Private plans start at \$7/mo.' To the right, there is a sign-up form with three input fields: a username field containing 'setyassidanovian', an email field containing 'example.setyassida@gmail.com', and a password field with a strength indicator. Below the password field, there is a green button labeled 'Sign up for GitHub'. At the bottom of the page, there is a dark blue banner with the text 'Want to use GitHub on your servers?'.

Setelah mengisi form secara benar, klik tombol “Sign up for GitHub”.

- d) Pada halaman berikutnya pastikan menu yang anda pilih seperti gambar di bawah ini, lalu klik tombol “Finish sign up”.

You've taken your first step into a larger world, @setyassidanovian

✓ Completed  
Set up a personal account

📄 Step 2:  
Choose your plan

🕒 Step 3:  
Go to your dashboard

### Choose your personal plan

Plan	Cost (view in IDR)	Private repos	
Large	\$50/month	50	<button>Choose</button>
Medium	\$22/month	20	<button>Choose</button>
Small	\$12/month	10	<button>Choose</button>
Micro	\$7/month	5	<button>Choose</button>
Free	\$0/month	0	<button>Chosen</button>

**Each plan includes:**

- Unlimited collaborators
- Unlimited public repositories
- ✓ Free setup
- ✓ SSL Protection
- ✓ Email support
- ✓ Wikis, Issues, Pages, & more

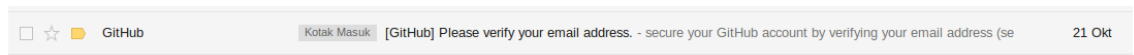
Charges to your account will be made in US Dollars. Converted prices are provided as a convenience and are only an estimate based on current exchange rates. Local prices will change as the exchange rate fluctuates. Don't worry, you can cancel or upgrade at any time.

☐ Help me set up an organization next  
Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees.  
[Learn more about organizations.](#)

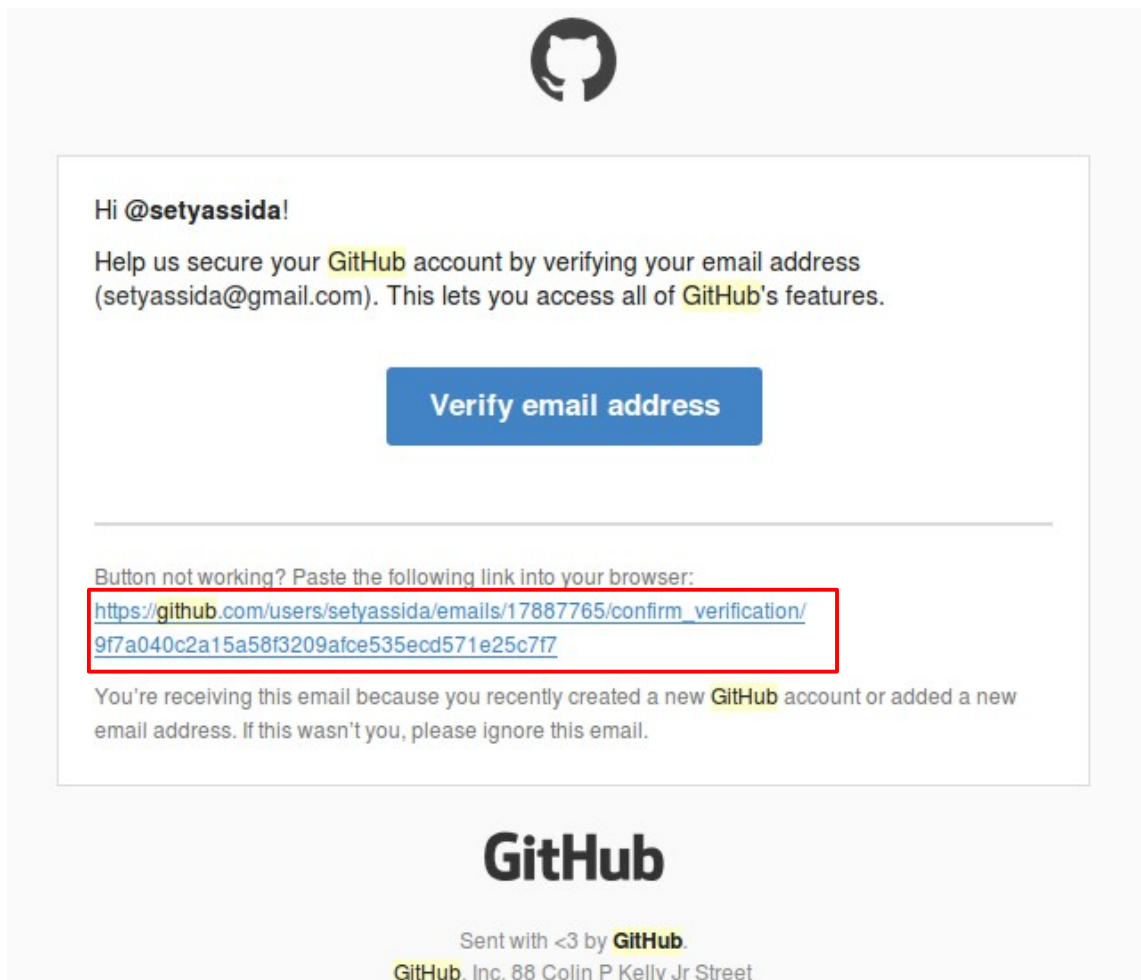
Finish sign up

**Jangan di centang (1)** →  
**2** →

e) Setelah itu, lakukan email verifikasi. Buka akun email anda lalu lihat email baru dari GitHub.



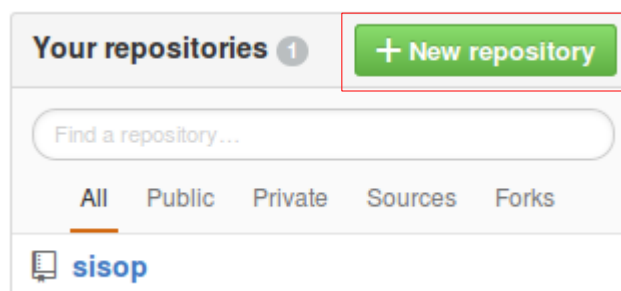
f) Klik tombol "Verify email address" atau klik link email verifikasi pada email tersebut.



- g) Setelah itu anda akan dialihkan kembali ke halaman GitHub. Lalu klik "Confirm".
- h) Sampai di sini anda telah berhasil mendaftar akun GitHub.

✓ **Langkah-langkah membuat REPOSITORY di Github**

- a) Masuk/Sign in ke akun GitHub anda.
- b) Klik tombol "+ New repository" pada bagian kanan bawah halaman.




- c) Isi form yang ada sesuai keinginan anda, contohnya bisa anda lihat pada gambar di bawah ini, jika sudah selesai mengisi form lanjutkan dengan menekan tombol "Create repository".

## Create a new repository


A repository contains all the files for your project, including the revision history.

Owner

Repository name

 setyassida


 / 


praktikumsisop 

Great repository names are short and memorable. Need inspiration? How about [irksom-sniffle](#).

Description (optional)

Ini adalah deskripsi dari Repository "praktikumsisop"

☒  **Public**  
Anyone can see this repository. You choose who can commit.


☐  **Private**  
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None**

 | 

Add a license: **None**



Create repository

d) Selamat, repository anda sudah selesai dibuat.

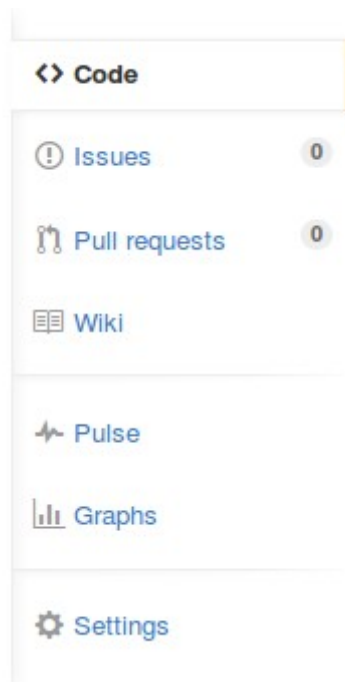
### ✓ Menambahkan Collaborators dalam repository GitHub

Collaborators adalah rekan kerja atau anggota tim yang mengerjakan sebuah project dalam suatu repository.

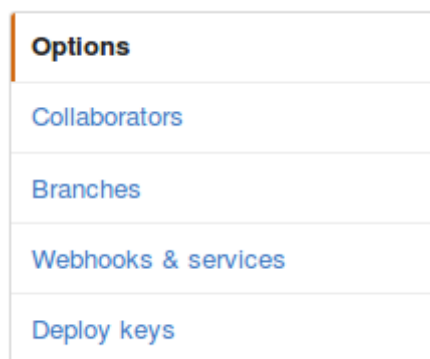
- Masuk ke dalam akun GitHub anda.
- Pilih repository yang ingin anda tambahkan dengan collaborators.



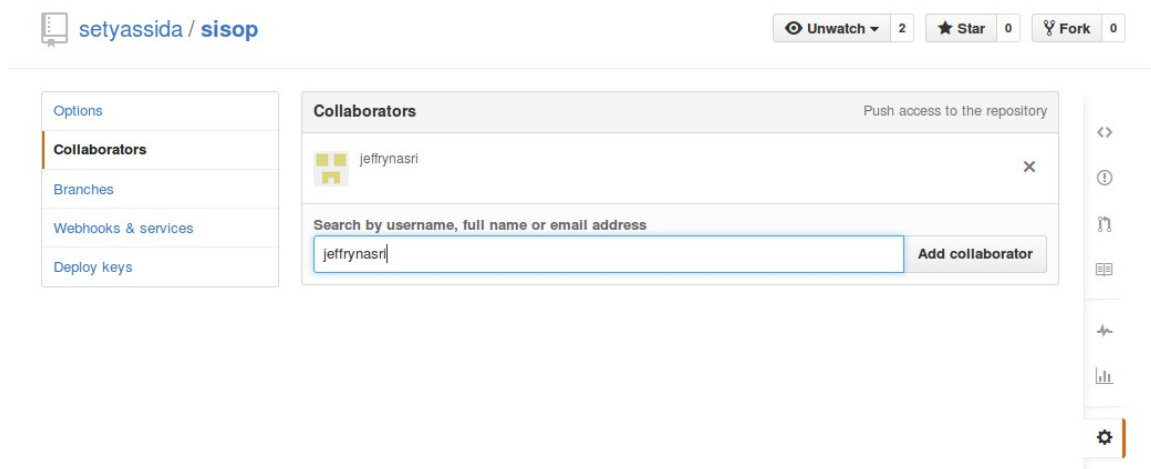
c) Pilih menu "Setting" pada pilihan menu yang berada di bagian kanan.



d) Pilih menu “Collaborators” pada meni bagian kiri.



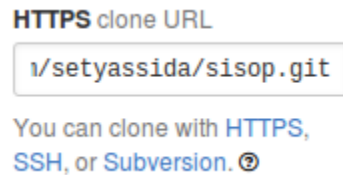
e) Isikan username collaborators anda pada kolom “Search by username, full name or email address” jika sudah ditemukan, lalu klik tombol “Add collaborators”



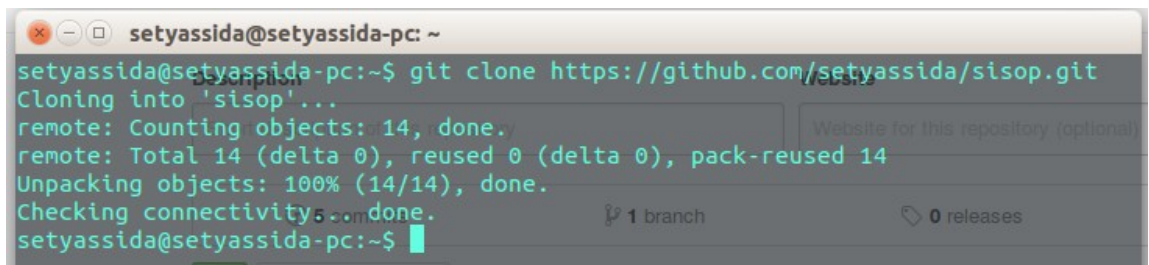
f) Selesai, satu collaborators sudah ditambahkan dalam repository anda. Lakukan dengan cara yang sama jika anda ingin menambahkan collaborators lagi dalam repository anda.

### ✓ Cara Clone repository GitHub

- Buka terminal console pada Linux anda.
- Pastikan laptop/PC anda sudah terhubung dengan internet
- Copy link repository yang ada dalam akun GitHub

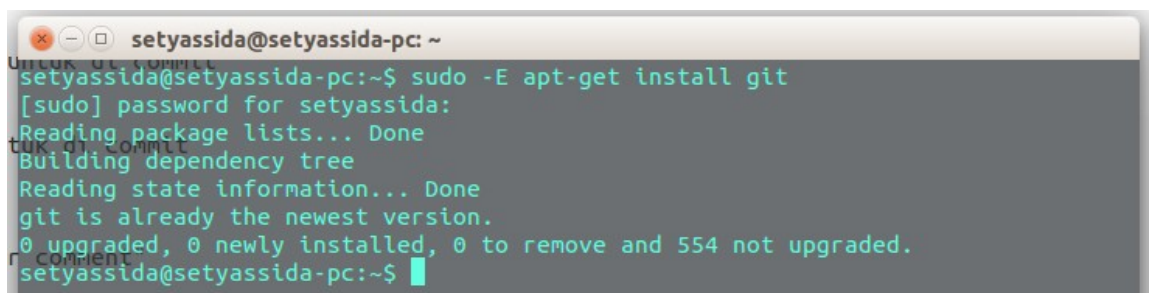


- Lalu ketikkan “git clone [https://github.com/nama\\_akun/nama\\_repo.git](https://github.com/setyassida/sisop.git)” (tanpa tanda petik). Contohnya “git clone <https://github.com/setyassida/sisop.git>”.
- Lalu jika muncul tampilan seperti gambar di bawah ini berarti clone repository berhasil dilakukan.



### ✓ Cara install Git

- Buka terminal console pada linux anda.
- Pastikan laptop/PC anda sudah tersambungan dengan internet.
- Ketikkan “sudo -E apt-get install git”



- Masukkan password anda.
- Gin akan secara otomatis ter-install.

### ✓ Perintah dasar dalam Git

Perintah-perintah dasar dari Git yang paling sering digunakan dan wajib kita kuasai yaitu: ***git init, git status, git add, git commit, git pull / push, git checkout, git diff, git merge, git remote dan git reset.*** Sebetulnya masih banyak fungsi lain dari Git untuk kasus-kasus



tertentu tapi mungkin perintah yang telah disebutkan sebelumnya yang akan paling sering kita gunakan, berikut ini adalah penjelasan singkatnya:

- ***git init*** untuk membuat repo lokal baru pada perintah ini akan dibuat sebuah folder baru yang bernama ".git"
- ***git status*** untuk melihat status dari repo lokal
- ***git add*** untuk menambahkan file ke dalam repo yang sebelumnya sudah dibuat
- ***git commit*** untuk menyimpan seluruh perubahan yang terjadi
- ***git pull / push*** untuk menyimpan dan mengambil data dari remote repo
- ***git checkout*** untuk pindah branch
- ***git diff*** untuk membandingkan perubahan file
- ***git merge*** untuk melakukan penggabungan antar branch
- ***git remote*** untuk menambahkan remote repo baru
- ***git reset*** untuk membatalkan perubahan pada repo lokal

## B) STRING PROCESSING

### ✓ **Scanf**

Asumsikan kita mempunyai array of char : `char A[100];`

a) Syntax dibawah ini akan membaca input sampai menemukan newline (`\n`).

```
scanf("%[^\\n]", A);
```

b) Syntax dibawah ini akan membaca input sampai menemukan newline (`\n`), dan membuang atau tidak mengikutsertakan newline (`\n`) ke dalam array of char `A[100]`.

```
scanf("%[^\\n]\\n", A);
```

c) Syntax dibawah ini akan membaca input sampai menemukan tanda koma (`,`).

```
scanf("%[^,]", A);
```

d) Syntax dibawah ini akan membaca input sampai menemukan tanda koma (`,`), dan membuang atau tidak mengikutsertakan tanda koma(`,`) tersebut ke dalam array of char `A[100]`.

```
Scanf("%[^,],", A);
```

### ✓ **Strtok**

Fungsi `strtok` adalah membagi string menjadi beberapa bagian yang dibatasi oleh

karakter yang telah ditentukan.

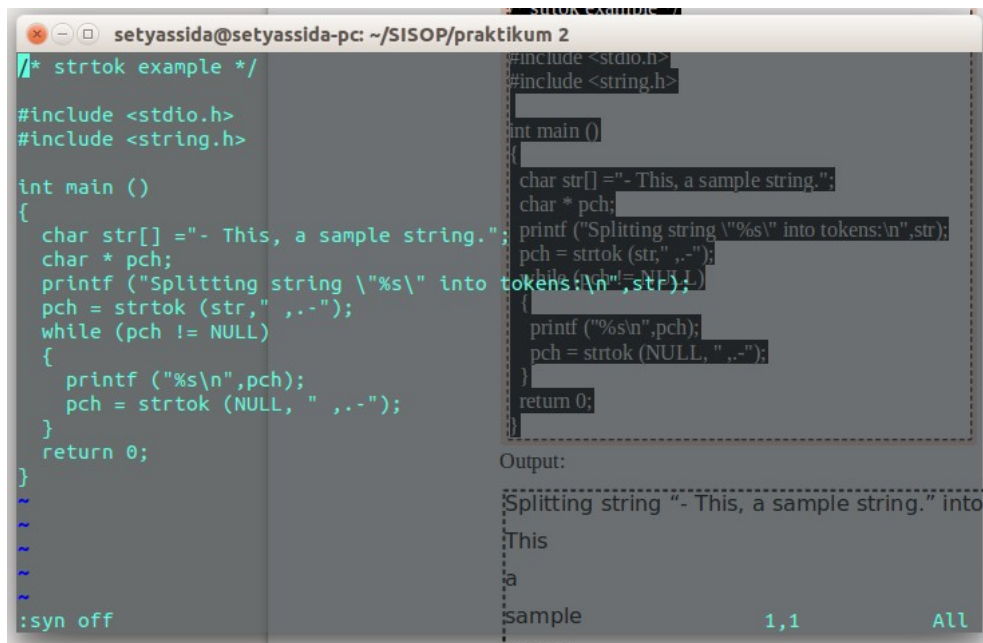
```
char * strtok ( char * str, const char * delimiters )
```

Contoh:

```
/* strtok example */

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

int main ()
{
    char str[] = "- This, a sample string.";
    char * pch;
    printf ("Splitting string \"%s\" into tokens:\n",str);
    pch = strtok (str, " ,.-");
    while (pch != NULL)
    {
        printf ("%s\n",pch);
        pch = strtok (NULL, " ,.-");
    }
    return 0;
}
```



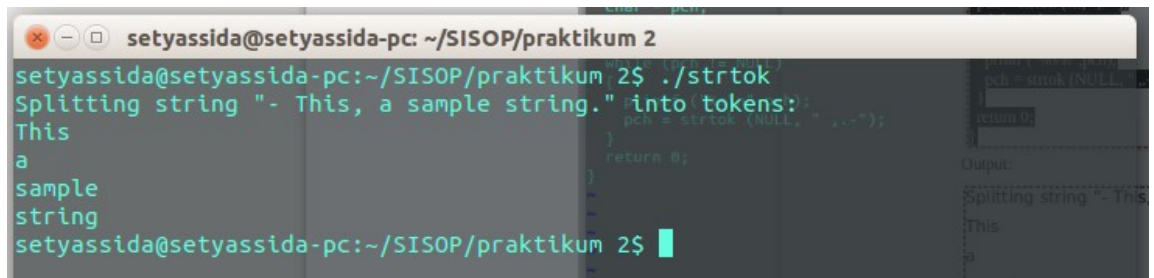
```
setyassida@setyassida-pc: ~/SISOP/praktikum 2
/* strtok example */
#include <stdio.h>
#include <string.h>

int main ()
{
    char str[] = "- This, a sample string.";
    char * pch;
    printf ("Splitting string \"%s\" into tokens:\n",str);
    pch = strtok (str, " ,.-");
    while (pch != NULL)
    {
        printf ("%s\n",pch);
        pch = strtok (NULL, " ,.-");
    }
    return 0;
}

Output:
Splitting string "- This, a sample string." into tokens:
This
a
sample
string
```

Output:

```
Splitting string "- This, a sample string." into tokens:
This
a
sample
string
```



```
setyassida@setyassida-pc: ~/SISOP/praktikum 2$ ./strtok
Splitting string "- This, a sample string." into tokens:
This
a
sample
string
setyassida@setyassida-pc:~/SISOP/praktikum 2$
```

Pada contoh di atas, kita memiliki string sample. Pada fungsi strtok, kita membagi string tadi menjadi beberapa bagian dengan pembatas karakter yaitu “,-“. Kita akan mendapatkan 3 buah string pecahan.

#### ✓ Strstr

Fungsi strstr() berguna untuk mencari urutan pertama suatu string di string lain.

```
const char * strstr ( const char * str1, const char * str2 );
char * strstr (      char * str1, const char * str2 );
```

Contoh :

```
#include<stdio.h>
#include<string.h>
int main(){
    char input[100];
    printf("HELLO WORLD\n");
    printf("Masukkan String : ");
    scanf("%[^\n]", input);
    printf("%s\n", input);

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

    char *token = strtok(input, " ");
    printf("%s\n", token);
return 0;
}
```

```
setyassida@setyassida-pc: ~/SISOP/praktikum 2
GNU nano 2.2.6 File: cekstring.c

#include<stdio.h>
#include<string.h>
int main(){
    char input[100];
    printf("HELLO WORLD\n");
    printf("Masukkan String : ");
    scanf("%s\n", input);
    printf("%s\n", input);

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

    char *token = strtok(input, " ");
    printf("%s\n", token);

    return 0;
}
```

Output:

```
setyassida@setyassida-pc: ~/SISOP/praktikum 2$ ./cekstring
HELLO WORLD
Masukkan String : hello world
hello world
world
hello
setyassida@setyassida-pc:~/SISOP/praktikum 2$
```

### c) FORK & EXEC

Forking adalah pembuatan sebuah proses. Kerja Forking seperti amoeba , yang dapat membelah diri jadi 2 bagian. Setiap forking akan menghasilkan child dan parent.

- Contohnya : Setiap memanggil fungsi fork, akan terbentuk 2 proses. Jika memanggil fungsi fork sekali lagi, maka masing-masing proses sebelumnya akan membelah, dan menghasilkan 4 proses.
- **Langkah - langkah forking sebagai berikut :**
  - a) Buat sebuah file c
  - b) Tulis code seperti di bawah ini

```

#include <stdio.h>
#include <unistd.h>

void main(){
    pid_t process;
    process = fork();
    if(process==0){
        printf("This Is Child Process\n");
    }else{
        process=wait();
        printf("This Is Parent Process\n");
    }
}

```

Ket :

- #include <unistd.h> = Header dari tipe data pid\_t
- pid\_t = Pendeklarasian sebuah variabel process
- fork() = Pembuatan sebuah proses
- wait() = Parent process menunggu child process selesai

c) Compile dengan gcc

```

/ You're working under a slight handicap. \
\ You happen to be human. \
-----
\      ^ ^      \
\      (oo)\_____\
\      (__)\       )\/\
|       ||----w |
|       ||
jeffry@nasri-HP-Pavilion-g4-Notebook-PC ~ $ gcc -o fork fork.c

```

d) Jalankan Program

```

jeffry@nasri-HP-Pavilion-g4-Notebook-PC ~ $ ./fork
This Is Child Process
This Is Parent Process
jeffry@nasri-HP-Pavilion-g4-Notebook-PC ~ $

```

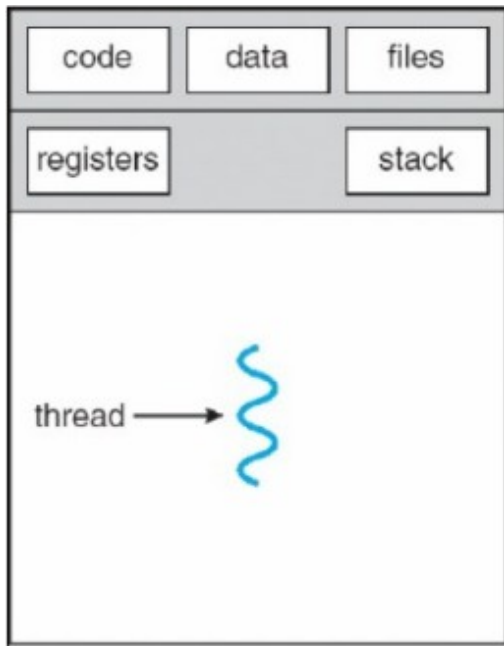
## D) THREADING

Thread adalah bagian terkecil dari suatu proses. Thread sering disebut dengan lightweight.

Dalam thread terdapat :

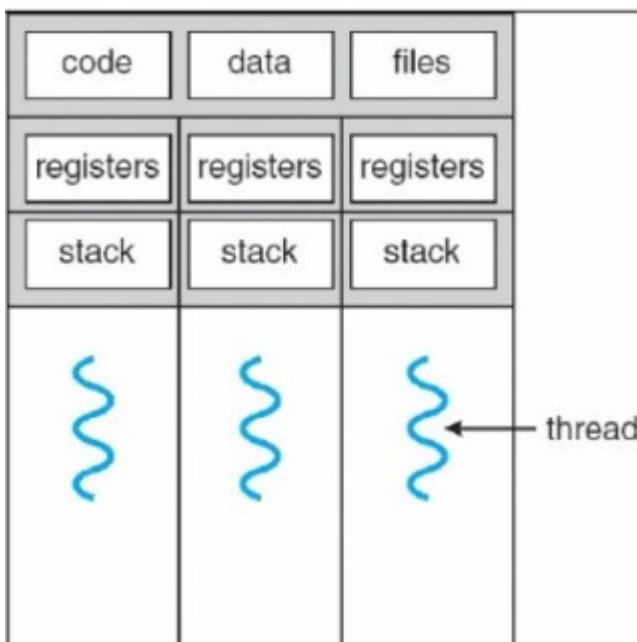
### ✓ Single Threading

Adalah sebuah proses yang hanya memiliki satu thread yang berjalan.



### ✓ Multi Threading

Adalah sebuah proses yang memiliki lebih dari satu thread yang berjalan didalamnya, sehingga 1 proses dapat menjalankan beberapa tugas dalam waktu yang bersamaan.



## ● Implementasi Threading

a) Buat sebuah file .c dan tulis code seperti yang dibawah ini

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

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

    for(i=0; i<=5; i++) {
        printf("Thread 1 : %d\n", i);
    }
}

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

    for(i=1; i<5; 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<5; i++) {
        printf("Main : %d\n", i);
    }

    pthread_join(t2, NULL);
}
```

Keterangan :

- pthread\_create : Untuk membuat threat
- pthread\_join : Untuk menunggu suatu thread

b) Compile dengan gcc di terminal

```
jeffry@nasri-HP-Pavilion-g4-Notebook-PC ~/Documents/Praktikum2 $ gcc -o testthread testthread.c -lpthread
jeffry@nasri-HP-Pavilion-g4-Notebook-PC ~/Documents/Praktikum2 $
```

c) Jalankan program

```
jeffry@nasri-HP-Pavilion-g4-Notebook-PC ~/Documents/Praktikum2 $ ./testthread
Thread 2 : 1 pthread_create(&t1, NULL, run, NULL);
Thread 2 : 2 pthread_create(&t2, NULL, run2, NULL);
Thread 2 : 3
Thread 2 : 4 i=0;
Thread 1 : 0
Thread 1 : 1 pthread_join(t1, NULL);
Thread 1 : 2
Thread 1 : 3 i=1; i<5; i++) {
Thread 1 : 4     printf("Main : %d\n", i);
Thread 1 : 5
Main : 1
Main : 2 pthread_join(t2, NULL);
Main : 3
Main : 4
```



## Daftar Pustaka

<http://blog.coldev.co/apa-itu-git/>

<http://www.boztutorial.ga/2015/03/cara-daftar-akun-github-terbaru.html>

<https://syaimif.wordpress.com/2014/03/12/cara-menggunakan-github-via-terminal-create-clone-push/>

<http://duken.info/blog/2009/12/berbagai-fungsi-string-pada-c-language/>

<http://www.cplusplus.com/reference/cstring/strtok/>

<http://stackoverflow.com/questions/3889992/how-does-strtok-split-the-string-into-tokens-in-c>

<http://dcc-dp.org/berita419-macam-macam-fungsi-string-pada-c-.html>

<http://www.cplusplus.com/reference/cstring/strstr/>