

## DAFTAR ISI

DAFTAR ISI.....	1
GIT HUB UP FILE.....	2
A. TUJUAN.....	2
B. TEORI SINGKAT.....	2
C. PRAKTIK.....	2
D. TUGAS.....	8
GITHUB PAGES.....	9
A. TUJUAN.....	9
B. TEORI SINGKAT.....	9
C. PRAKTIK.....	9
D. LATIHAN.....	19
E. TUGAS.....	19
GITHUB KOLABORASI.....	20
A. TUJUAN.....	20
B. TEORI SINGKAT.....	20
C. PRAKTIK.....	21
D. TUGAS.....	29
GOOGLE APPS ENGINE.....	30
A. TUJUAN.....	30
B. TEORI SINGKAT.....	30
C. PRAKTIK.....	30
D. TUGAS.....	33
Pengenalan Heroku.....	34
A. TUJUAN.....	34
B. TEORI SINGKAT.....	34
C. PRAKTIK.....	34
D. TUGAS.....	46
DOCKER.....	47
A. TUJUAN.....	47
B. TEORI SINGKAT.....	47
C. PRAKTIK.....	47
D. LATIHAN.....	60
E. TUGAS.....	60

<b>DOCKER CONTAINER</b> .....	61
<b>A. TUJUAN</b> .....	61
<b>B. TEORI SINGKAT</b> .....	61
<b>C. PRAKTIK</b> .....	61
<b>D. LATIHAN</b> .....	75
<b>E. TUGAS</b> .....	75
<b>DOCKER COMPOSE</b> .....	76
<b>A. TUJUAN</b> .....	76
<b>B. TEORI SINGKAT</b> .....	76
<b>C. PRAKTIK</b> .....	76
<b>D. LATIHAN</b> .....	82
<b>E. TUGAS</b> .....	82

## **GIT HUB UP FILE**

### **A. TUJUAN**

- Mahasiswa memahami penggunaan **Git**.
- Mahasiswa mampu memanfaatkan **Git** dan **GitHub**.

### **B. TEORI SINGKAT**

#### **Definisi**

GitHub adalah situs yang menyediakan layanan hosting dan version control repository Git yang berbasis web. GitHub umumnya digunakan untuk menghosting pengembangan proyek yang bersifat open source. GitHub menyediakan berbagai fitur seperti pengendalian akses(access control) dan beberapa fitur lain untuk pengerjaan bersama seperti bug tracking, permintaan fitur, pengelolaan tugas, dan wiki untuk setiap proyek yang sedang dikerjakan. Situs ini didirikan serta dikembangkan oleh Tom Preston-Werner, Chris Wanstrath, dan PJ Hyett.

#### **Manfaat**

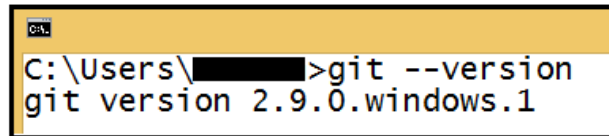
Seseorang atau tim developer pasti pernah menghadapi proyek yang sedang dikerjakan berantakan serta informasi mengenai perubahan maupun tambahan proyek hanya di komunikasikan secara personal maupun via alat komunikasi. Hal tersebut mempersulit dalam perbaikan proyek maupun pengembangan lanjutan. Berikut manfaat GitHub untuk menjawab tantangan tersebut:

1. Memudahkan Developer untuk berpartisipasi dalam pengembangan berbagai software open-source.
2. Memudahkan dokumentasi dari proyek yang sedang dikerjakan.
3. Dengan menggunakan GitHub maka perubahan pada tiap file akan bisa terlacak. Contohnya jika seseorang ingin membandingkan performance antara versi baru dengan versi sebelumnya.
4. Seperti layanan VCS (Version Control System) lainnya, kita dapat membuat local repository yang disimpan secara offline pada komputer kita, maupun secara online.
5. GitHub memiliki aplikasi desktop maupun web yang memiliki banyak fitur-fitur yang menunjang developer yang bekerja secara berjauhan maupun berpindah pindah tempat.

6. Fitur social tambahan seperti seseorang developer dapat mengikuti GitHub user lainnya, membookmark suatu project, dan banyak lagi.

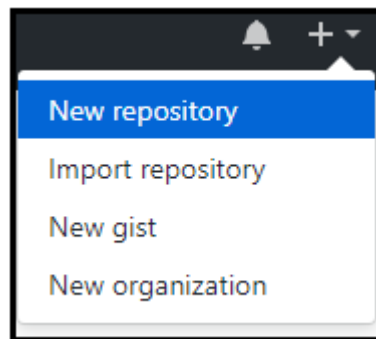
### C. PRAKTIK

1. Memastikan sudah menginstall git bash. Apabila belum, kunjungi <https://gitforwindows.org/> untuk mengunduh git bash yang diperlukan sesuai system operasi perangkat Anda. Kemudian, melakukan pengecekan software tersebut pada command prompt maupun pada fitur search bawaah system operasi anda. Berikut pengecekan git pada cmd:



```
C:\Users\>git --version
git version 2.9.0.windows.1
```

2. Mendaftar Github pada situs berikut <https://github.com/join?source=header-home> dengan mengisikan username, alamat email, password dan verifikasi akun.
3. Membuka akun Github Anda, lalu membuat repository(biasanya disingkat Repo) baru yang berada di kanan atas dengan disimbolkan “+”. Setelah itu pilih New repository.




4. Mengisi repo baru dengan mengisikan nama repo, deskripsi, sifat repo(default Public) dan inisiasi ReadMe seperti berikut:

## Create a new repository

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

---

Owner:  masihsendiri999 / Repository name:  ✓

Great repository names are short and memorable. Need inspiration? How about *crispy-dollop*.

Description (optional):

☒ **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: 
 Add a license:  ⓘ

**Create repository**

Tampilan setelah create Repo :

 masihsendiri999 / UpFiles
 Watch 0 Star 0 Fork 0


<> Code
 Issues 0
 Pull requests 0
 Projects 0
 Wiki
 Insights
 Settings

Ini kolom deskripsi. Edit

Manage topics

1 commit
 1 branch
 0 releases
 1 contributor

Branch: master
 New pull request
 Create new file
 Upload files
 Find file
 Clone or download

 masihsendiri999 Initial commit
 Latest commit dd9b7de just now


README.md Initial commit just now

README.md

## UpFiles

Ini kolom deskripsi.

url pada url bar tertera alamat repo seperti berikut :

 GitHub, Inc. [US] | <https://github.com/masihsendiri999/UpFiles>

5. Membuka Git Bash untuk menjalankan perintah git clone. Dimana perintah tersebut digunakan untuk mengunduh atau melakukan clone proyek yang ada di repo online atau di GitHub. Dengan cara copy url yang ada di url bar atau klik pada tombol clone lalu copy urlnya. Kemudian tempel alamat tersebut untuk menjalankan perintah “git clone **namaUrl**” (tanpa tanda petik ganda) pada git bash. Sebelumnya menjalankan perintah pwd untuk mengetahui letak file yang telah diunduh atau diclone menggunakan perintah git clone.

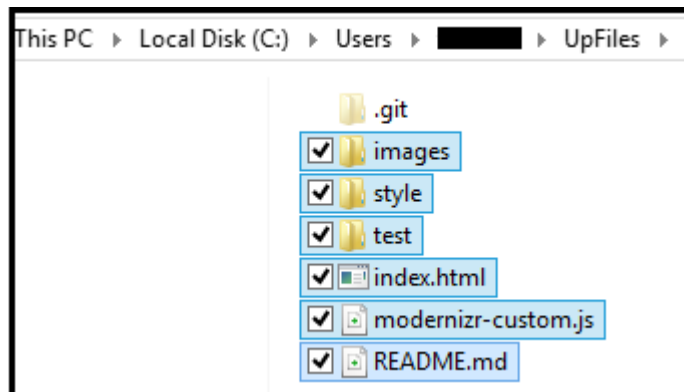
```
██████████@redvelvet MINGW64 ~  
$ pwd  
/c/Users/██████████  
  
██████████@redvelvet MINGW64 ~  
$ git clone https://github.com/masihsendiri999/UpFiles  
Cloning into 'UpFiles'...  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
Unpacking objects: 100% (3/3), done.  
Checking connectivity... done.  
  
██████████@redvelvet MINGW64 ~  
$ cd UpFiles/
```

Keterangan : yang disensor adalah NamaUser(nama user) perangkat.

6. Menjalankan perintah git status untuk mengecek status repository apakah ada perubahan maupun penambahan file.

```
██████████@redvelvet MINGW64 ~/UpFiles (master)  
$ git status  
On branch master  
  
Initial commit  
  
nothing to commit (create/copy files and use "git add" to track)
```

7. Menambahkan file seperti yang ada di repo <https://github.com/chessaAditirta/webCoba> yang berisi web statis. Dengan cara download zip, lalu unzip file tersebut. Setelah itu hasil ekstrak masukkan pada direktori UpFiles seperti di bawah :



8. Selanjutnya buka kembali git bash Anda jalankan perintah git status seperti sebelumnya dan perhatikan perbedaannya.

```
@redvelvet MINGW64 ~/UpFiles (master)
$ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        README.md
        images/
        index.html
        modernizr-custom.js
        style/
        test/

nothing added to commit but untracked files present (use "git add" to track)
```

Menjalankan perintah git add . atau git add \* untuk menambahkan file-file pada repo dan jalankan kembali perintah git status kemudian bandingkan perbedaannya.

```
@redvelvet MINGW64 ~/UpFiles (master)
$ git add .
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in modernizr-custom.js.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in style/main.css.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in style/temp.css.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in test/test1.html.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in test/test2.html.
The file will have its original line endings in your working directory.
```

Menjalankan perintah git status setelah di add.

```
@redvelvet MINGW64 ~/UpFiles (master)
$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   README.md
        new file:   images/2018-Dodge-Challenger-SRT-Demon-v3-1080.jpg
        new file:   images/2018-Dodge-Challenger-SRT1000.jpg
        new file:   images/2018-Dodge-Challenger-SRT1600.jpg
        new file:   images/2018-Dodge-Challenger-SRT500.jpg
        new file:   images/2018-Dodge-Challenger-SRT800.jpg
```

Setelah melakukan perintah di atas berikan sedikit ulasan mengenai perbedaan git status sebelumnya dan setelah dilakukan perintah git add.

9. Melakukan perintah git config untuk menginisiasi username github dan email github untuk commit nanti.



```
@redvelvet MINGW64 ~/UpFiles (master)
$ git config --global user.name "masihsendiri999"

@redvelvet MINGW64 ~/UpFiles (master)
$ git config --global user.email "masihsendiri999@gmail.com"
```

10. Melakukan perintah git commit. Perintah ini penting karena sebagai persiapan upload file-file(push) dalam arti lain persiapan push adalah memindahkan file-file yang akan di push ke sebuah tempat atau memori sementara untuk nanti di upload ke repository online(gitHub). Berikut format perintah yang dijalankan :

**git commit -m “Isi bebas lepas tanpa tekanan dan tak terbatas”**

```
@redvelvet MINGW64 ~/UpFiles (master)
$ git commit -m "0.1.3:Ini bisa diisi bebas"
[master dc89564] 0.1.3:Ini bisa diisi bebas
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working
directory.
warning: LF will be replaced by CRLF in modernizr-custom.js.
The file will have its original line endings in your working
directory.
warning: LF will be replaced by CRLF in style/main.css.
The file will have its original line endings in your working
directory.
warning: LF will be replaced by CRLF in style/temp.css.
The file will have its original line endings in your working
directory.
warning: LF will be replaced by CRLF in test/test1.html.
The file will have its original line endings in your working
directory.
warning: LF will be replaced by CRLF in test/test2.html.
The file will have its original line endings in your working
directory.
51 files changed, 503 insertions(+)
create mode 100644 images/2018-Dodge-Challenger-SRT-Demon-V
3-1080.jpg
create mode 100644 images/2018-Dodge-Challenger-SRT1000.jpg
create mode 100644 images/2018-Dodge-Challenger-SRT1600.jpg
create mode 100644 images/2018-Dodge-Challenger-SRT500.jpg
```

11. Terakhir melakukan perintah git push origin master. Diikuti dengan mengisi username dan password akun github Anda.



```

@redvelvet MINGW64 ~/UpFiles (master)
$ git push origin master
Fatal: HttpRequestException encountered.
Username for 'https://github.com': masihsendiri999
Counting objects: 57, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (56/56), done.
Writing objects: 100% (57/57), 4.47 MiB | 666.00 KiB/s, done
Total 57 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/masihsendiri999/UpFiles
    dd8b7de..dc89564  master -> master

```

Tampilan pada akun github anda terdapat perubahan atau lebih tempatnya penambahan file di repository UpFiles Anda.

masihsendiri999 / UpFiles

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Ini kolom deskripsi. Edit

Manage topics

2 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

File	Commit	Time
images	0.1.3:Ini bisa diisi bebas	4 minutes ago
style	0.1.3:Ini bisa diisi bebas	4 minutes ago
test	0.1.3:Ini bisa diisi bebas	4 minutes ago
README.md	0.1.3:Ini bisa diisi bebas	4 minutes ago
index.html	0.1.3:Ini bisa diisi bebas	4 minutes ago
modernizr-custom.js	0.1.3:Ini bisa diisi bebas	4 minutes ago

#### D. TUGAS

1. Buat aplikasi web statis menurut **imajinasi masing-masing** memiliki beberapa halaman web dan salah satu halamannya tersusun atas nama, nim serta hal yang menunjukkan profil saudara seperti halaman web mengenai tentang , profil atau About pada umumnya. (Bebas memakai bahasa atau framework apapun namun web CMS atau **Content Management System** tidak diijinkan).

**Catatan:**

Format nama folder dan nama aplikasi **namatugNim**. Contoh:  
**satriatug155610024**

2. Push aplikasi web statis tersebut disertai dengan alamat repo. Contoh :  
<https://github.com/satriang/satriatug155610024>

## GITHUB PAGES

### A. TUJUAN

- Mahasiswa memahami penggunaan **Github Pages**.
- Mahasiswa mampu memanfaatkan **Github Pages** untuk hosting.

### B. TEORI SINGKAT

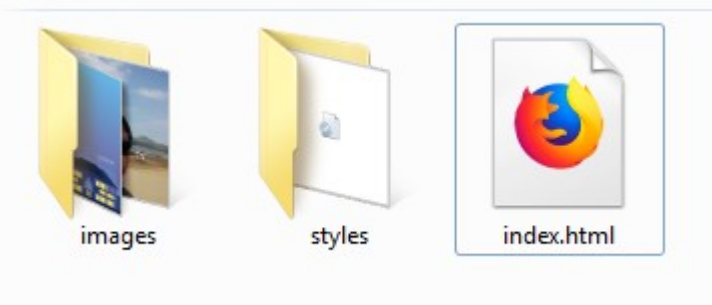
Github Pages merupakan hosting static file gratis yang disediakan oleh Github untuk para developer. Static file disini ialah seperti HTML, CSS, Javascript, Images, dll, yakni kode-kode client-side sebuah website yang tidak lagi memerlukan compile, transpile ataupun webserver untuk menjalankannya. Keuntungan Menggunakan GitHub Pages

- Tidak Perlu Setup Server. Kita tidak perlu setup server karena GitHub sendiri yang akan menjadi host GitHub Page kita.
- Keterbatasan dan Keamanan. GitHub Page hanya memperbolehkan/membatasi kita untuk menggunakan beberapa tools saja, misalkan Jekyll dan Hugo. Dengan ini kita tidak perlu pusing memikirkan keamanan.
- Cepat Deploy. Kita tinggal membuat repository dengan nama <username>.github.io lalu ketikkan kode HTML atau Markdown pada laptop kemudian push ke repository tersebut.

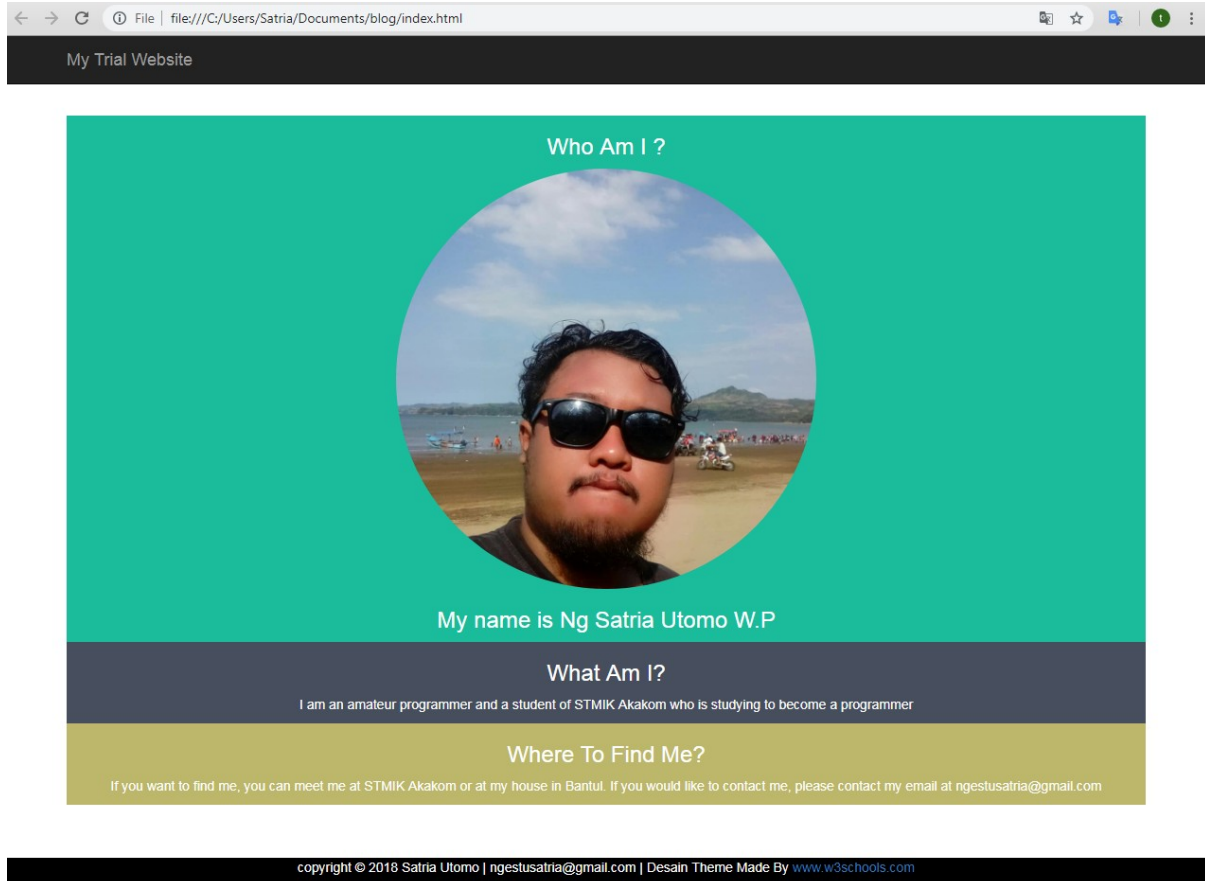
### C. PRAKTIK

Untuk dapat menggunakan github io pastikan anda sudah mendaftar akun di layanan github.


Setelah itu pastikan komputer anda sudah terinstal aplikasi untuk mengatur repositori Git kita dari desktop. Aplikasi yang digunakan bisa git bash atau Github Desktop. Siapkan web statis yang akan di upload pada repositori github



Pastikan dengan membuka web statis di lokal



Setelah itu buka <https://github.com/>

 [Why GitHub?](#) [Enterprise](#) [Explore](#) [Marketplace](#) [Pricing](#)  [Sign in](#) [Sign up](#)

# Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside 31 million developers.

Username

Pick a username

Email

you@example.com

Password

Create a password

Make sure it's more than 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

## Get started with GitHub Enterprise

Take collaboration to the next level with security and administrative features built for businesses.

Contact Sales

[Or start a free trial of Enterprise Server →](#)

Klik sign in kemudian masukkan akun github yang sudah pernah di daftarkan



## Sign in to GitHub

Username or email address

Password

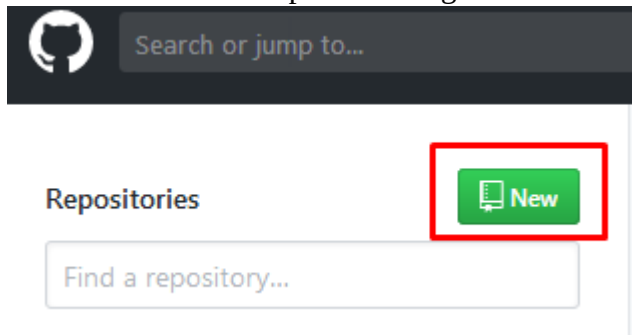
[Forgot password?](#)

Sign in

New to GitHub? [Create an account.](#)

[Terms](#) [Privacy](#) [Security](#) [Contact GitHub](#)

Setelah masuk buat repositori dengan klik “new” seperti contoh dibawah ini



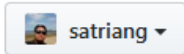
Agar repositori kita dapat berfungsi sebagai Github Pages yang dapat ditampilkan sebagai website, maka syarat dari Github adalah nama repositori mesti berformat username.github.io. Contohnya, karena username saya adalah **satriang**, maka saya harus membuat repository dengan nama **satriang.github.io**. Centang pada “Initialize this

repository with a README” Setelah itu klik tombol Create Repository seperti contoh dibawah ini

## Create a new repository

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

Owner



Repository name \*

satriang.github.io



Great repository names are short and memorable. Need inspiration? How about **silver-rotary-phone**.

Description (optional)



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 ▾

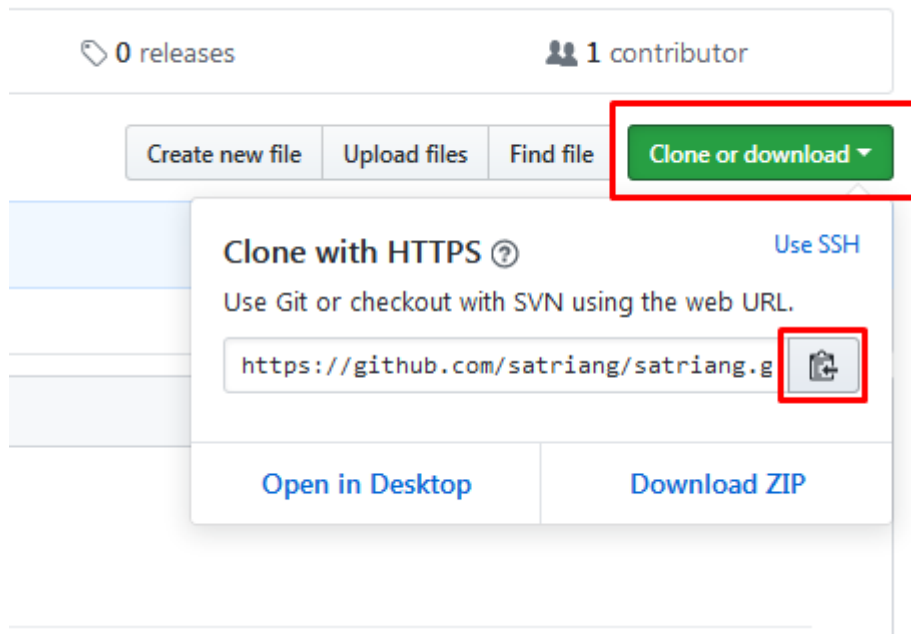
Add a license: None ▾



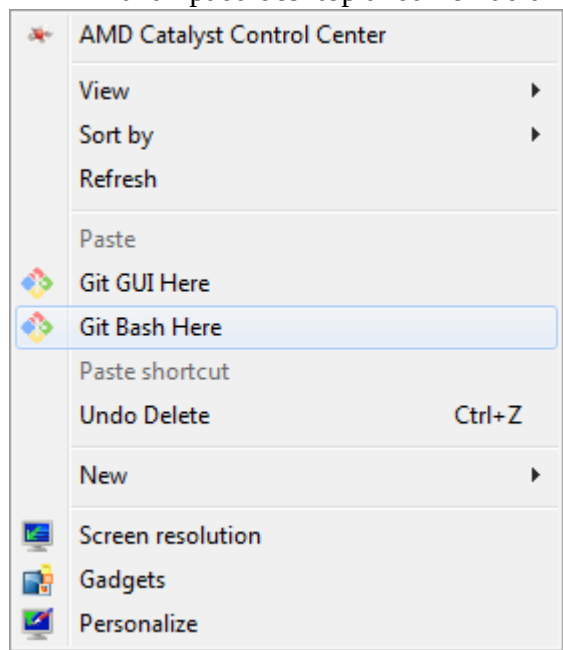
Create repository

Setelah itu klik “Clone or Download” lalu klik icon di sebelah url repositori

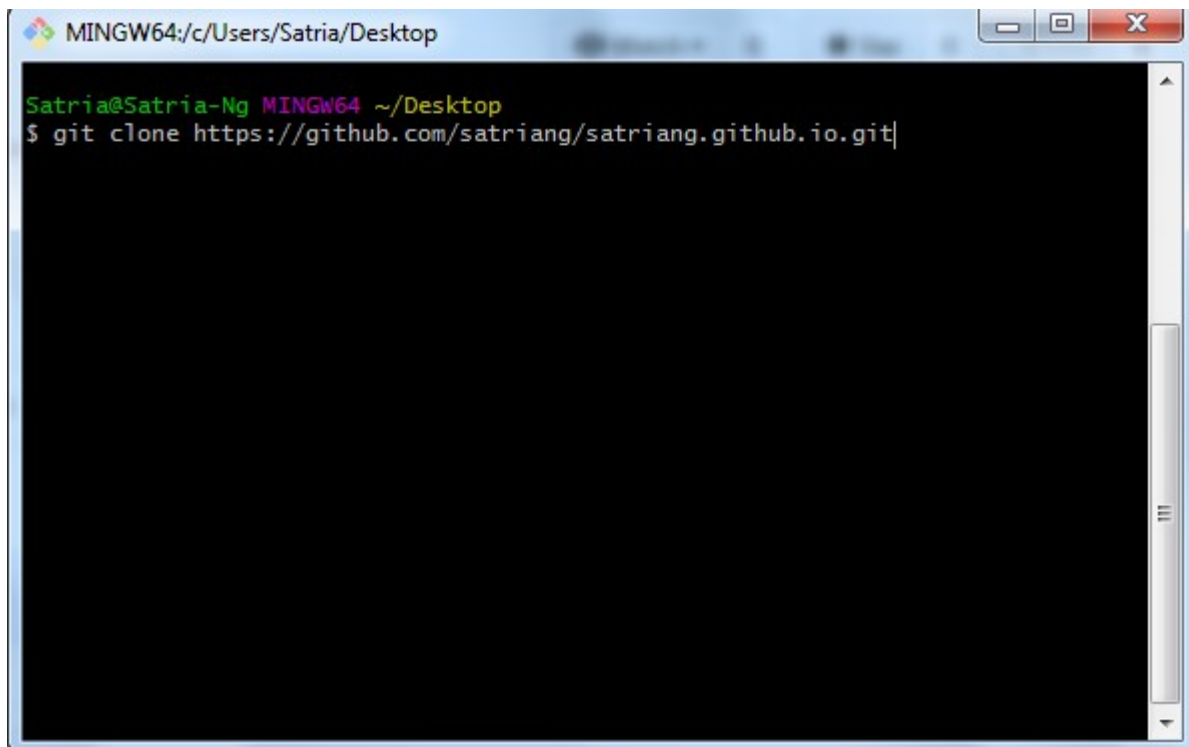




Buka aplikasi untuk git yang terinstall di komputer anda. Disini menggunakan git bash. Klik kanan pada desktop anda kemudian pilih “git bash use here” seperti contoh



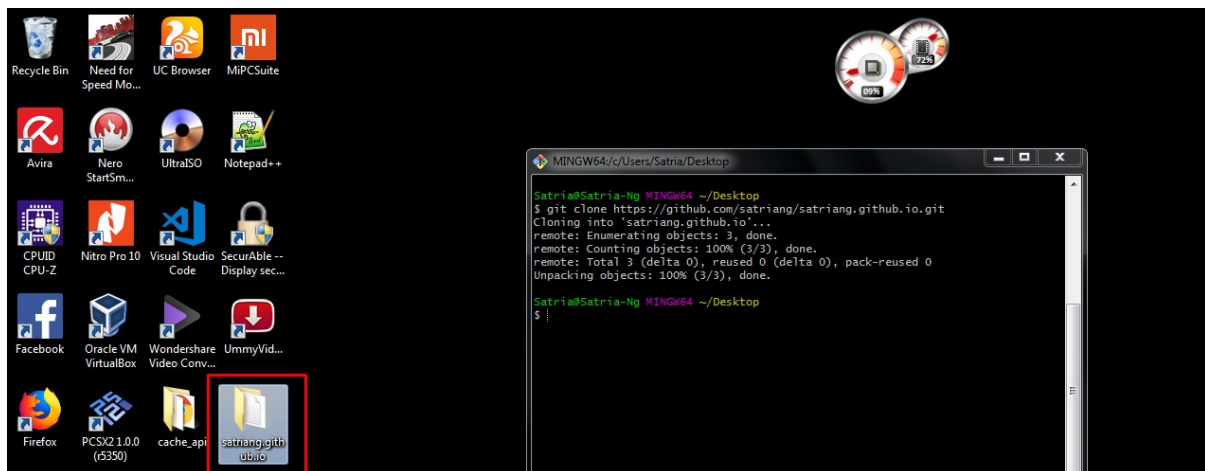
Ketikkan “git clone alamat repositori” untuk mendownload repositori. Contoh seperti dibawah ini



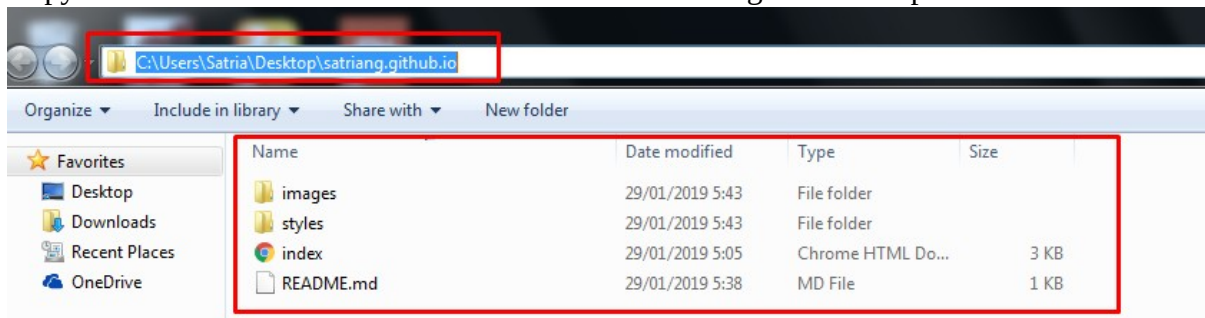
```
MINGW64:/c/Users/Satria/Desktop

Satria@Satria-Ng MINGW64 ~/Desktop
$ git clone https://github.com/satriang/satriang.github.io.git
```

Buka pada layar desktop komputer anda maka akan ada satu folder dengan nama repositori yang sama seperti nama repoitori yang sudah dibuat seperti contoh dibawah ini

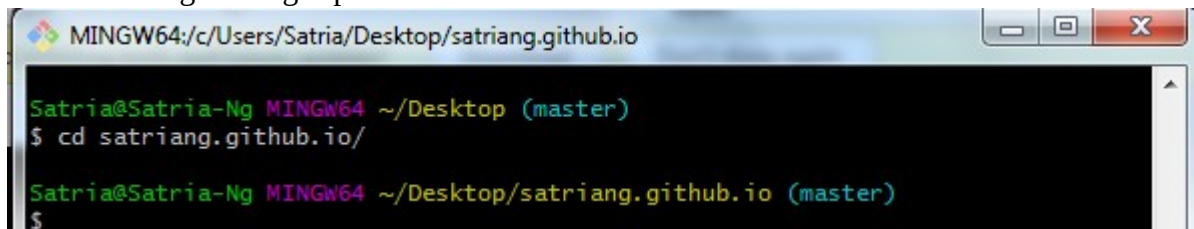


Copy semua file web statis ke dalam direktori “username.github.io” seperti contoh dibawah ini



Kemudian kembali pada git bash yang digunakan untuk clone atau download repositori.

Ketikkan perintah “cd username.github.io” sesuai dengan nama username github anda masing-masing seperti contoh dibawah ini

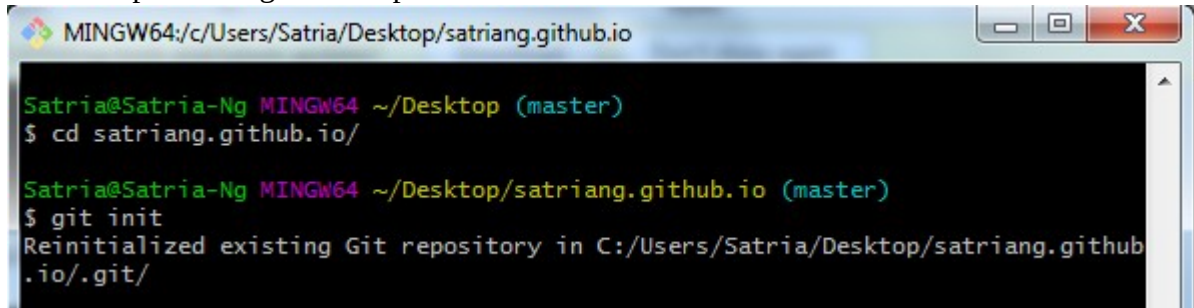


```
MINGW64:/c/Users/Satria/Desktop/satriang.github.io

Satria@Satria-Ng MINGW64 ~/Desktop (master)
$ cd satriang.github.io/

Satria@Satria-Ng MINGW64 ~/Desktop/satriang.github.io (master)
$
```

Ketikkan perintah “git init” seperti contoh dibawah ini

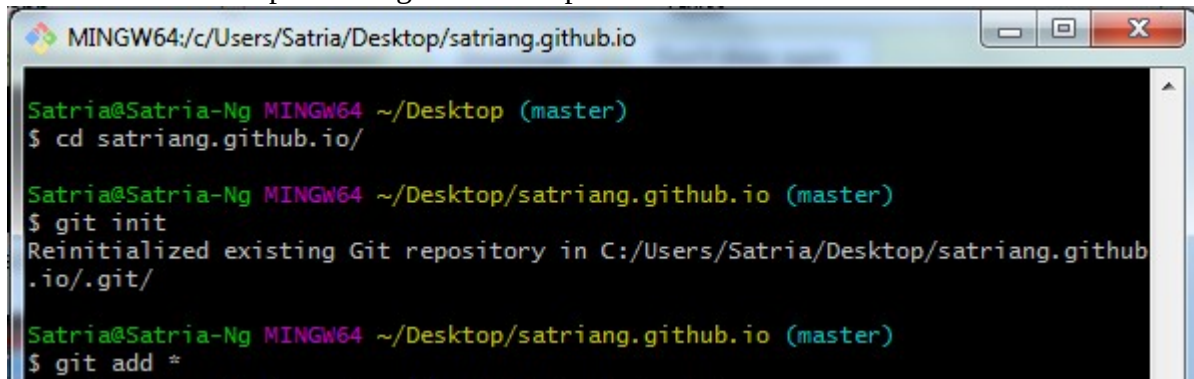


```
MINGW64:/c/Users/Satria/Desktop/satriang.github.io

Satria@Satria-Ng MINGW64 ~/Desktop (master)
$ cd satriang.github.io/

Satria@Satria-Ng MINGW64 ~/Desktop/satriang.github.io (master)
$ git init
Reinitialized existing Git repository in C:/Users/Satria/Desktop/satriang.github.io/.git/
```

Setelah itu ketikkan perintah “git add \* ” seperti contoh dibawah ini



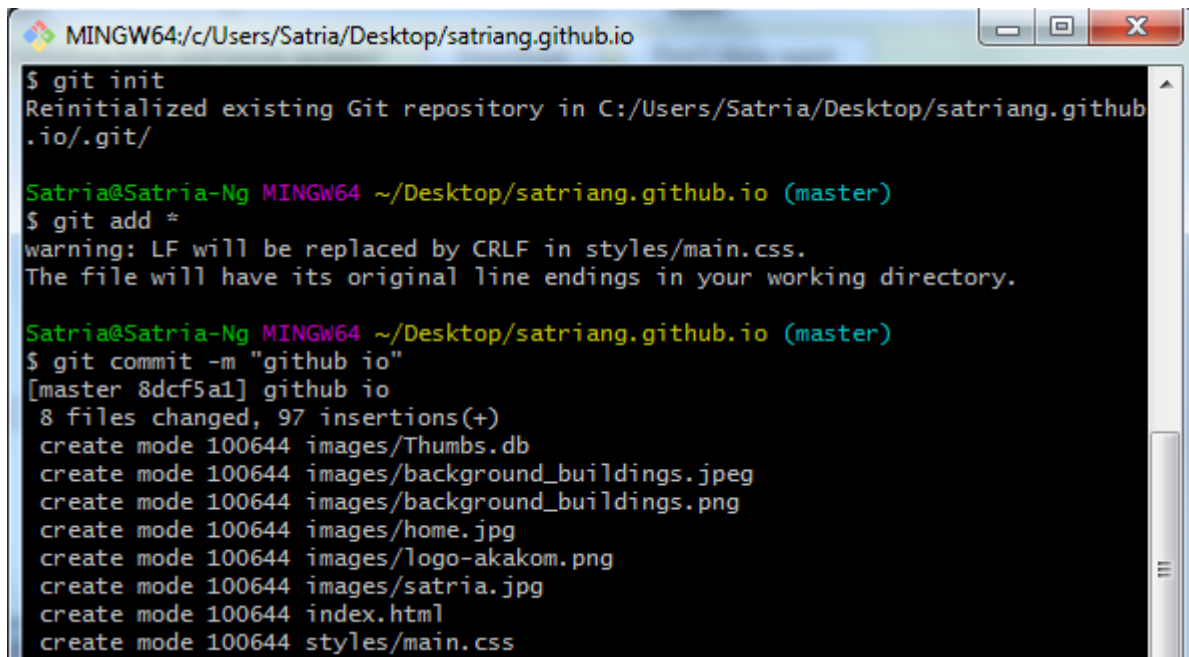
```
MINGW64:/c/Users/Satria/Desktop/satriang.github.io

Satria@Satria-Ng MINGW64 ~/Desktop (master)
$ cd satriang.github.io/

Satria@Satria-Ng MINGW64 ~/Desktop/satriang.github.io (master)
$ git init
Reinitialized existing Git repository in C:/Users/Satria/Desktop/satriang.github.io/.git/

Satria@Satria-Ng MINGW64 ~/Desktop/satriang.github.io (master)
$ git add *
```

Kemudian ketikkan “git commit -m “github io” seperti contoh dibawah ini

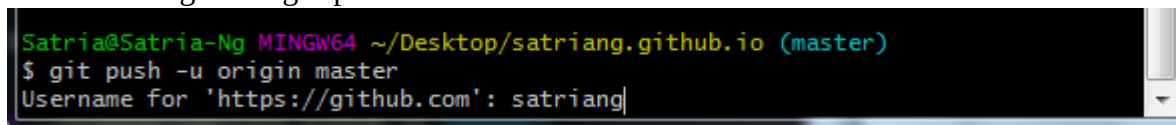


```
MINGW64:/c/Users/Satria/Desktop/satriang.github.io
$ git init
Reinitialized existing Git repository in C:/Users/Satria/Desktop/satriang.github.io/.git/

Satria@Satria-Ng MINGW64 ~/Desktop/satriang.github.io (master)
$ git add *
warning: LF will be replaced by CRLF in styles/main.css.
The file will have its original line endings in your working directory.

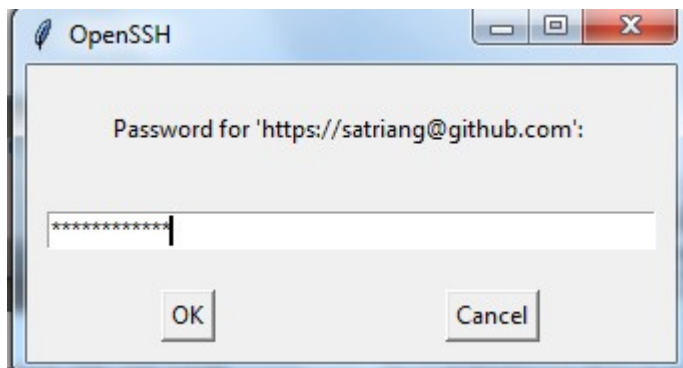
Satria@Satria-Ng MINGW64 ~/Desktop/satriang.github.io (master)
$ git commit -m "github io"
[master 8dcf5a1] github io
8 files changed, 97 insertions(+)
create mode 100644 images/Thumbs.db
create mode 100644 images/background_buildings.jpeg
create mode 100644 images/background_buildings.png
create mode 100644 images/home.jpg
create mode 100644 images/logo-akakom.png
create mode 100644 images/satria.jpg
create mode 100644 index.html
create mode 100644 styles/main.css
```

Kemudian ketikkan perintah “git push -u origin master” dan masukkan username github anda masing-masing seperti contoh dibawah ini



```
Satria@Satria-Ng MINGW64 ~/Desktop/satriang.github.io (master)
$ git push -u origin master
Username for 'https://github.com': satriang|
```

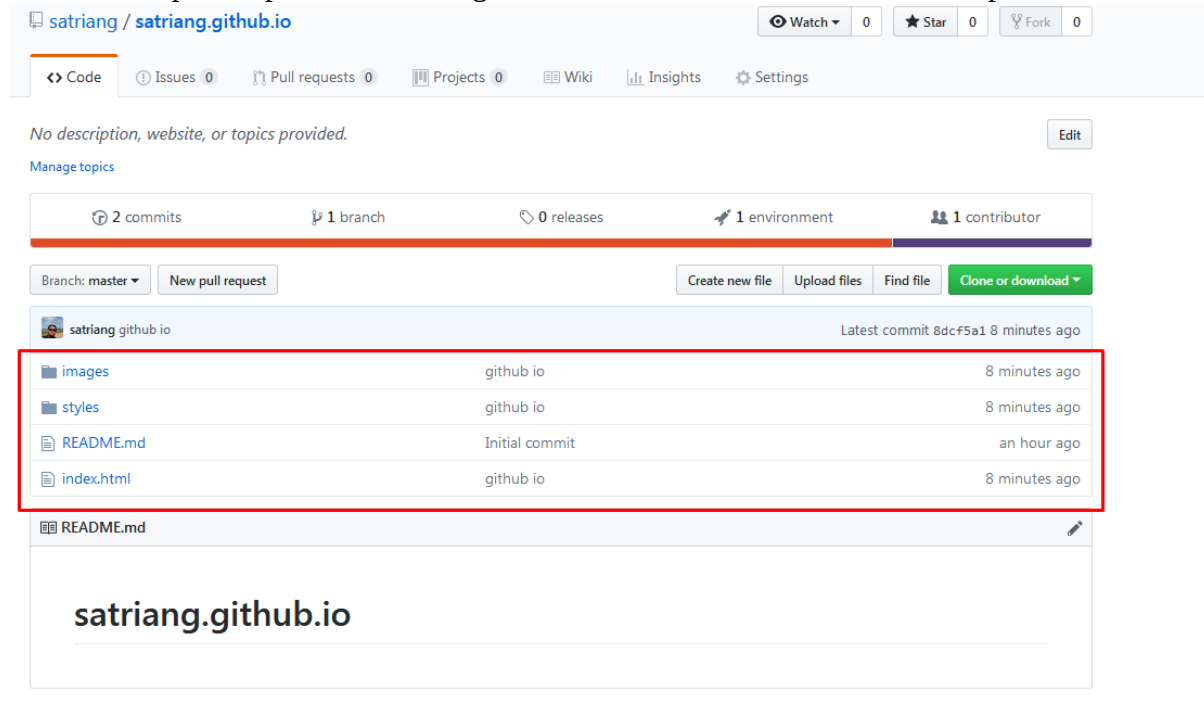
Setelah itu akan muncul permintaan password akun github dan isikan password sesuai akun github anda seperti contoh dibawah ini



Maka akan ada proses upload pada git bash seperti gambar dibawah ini

```
Satria@Satria-Ng MINGW64 ~/Desktop/satriang.github.io (master)
$ git push -u origin master
Username for 'https://github.com': satriang
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 2 threads.
Compressing objects: 100% (11/11), done.
Writing objects: 100% (12/12), 400.98 KiB | 14.85 MiB/s, done.
Total 12 (delta 0), reused 0 (delta 0)
To https://github.com/satriang/satriang.github.io.git
   b04d46e..8dcf5a1  master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

Cek kembali pada repositori anda di github untuk memastikan file sudah terupload



satriang / satriang.github.io

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

No description, website, or topics provided. Edit

Manage topics

2 commits 1 branch 0 releases 1 environment 1 contributor

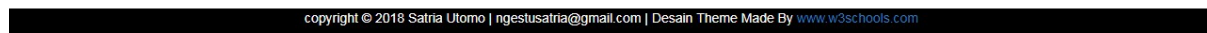
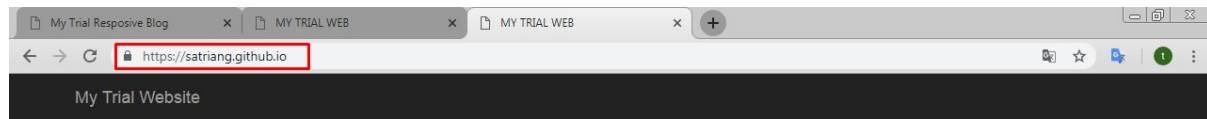
Branch: master New pull request Create new file Upload files Find file Clone or download

satriang github io Latest commit 8dcf5a1 8 minutes ago		
images	github io	8 minutes ago
styles	github io	8 minutes ago
README.md	Initial commit	an hour ago
index.html	github io	8 minutes ago

README.md

satriang.github.io

Kemudian kita coba mengakses web statis yang kita buat dengan menggunakan browser dan ketikkan “username.github.io” pada url browser. Contoh url “satriang.github.io”



Jika dapat diakses Github Pages sudah berhasil dibuat seperti contoh di atas

#### **D. LATIHAN**

- Latihan diberikan oleh dosen pengampu

#### **E. TUGAS**

- Tugas diberikan oleh dosen pengampu

## GITHUB KOLABORASI

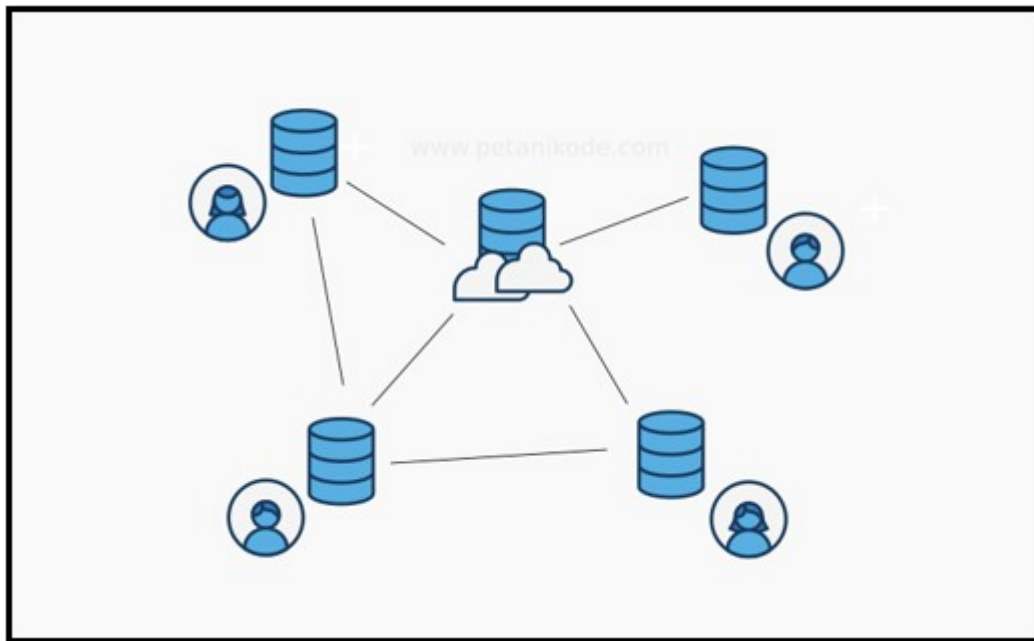
### A. TUJUAN

- Mahasiswa memahami penggunaan **Git** dan **Github**.
- Mahasiswa mampu memanfaatkan **Github** untuk melakukan kolaborasi.

### B. TEORI SINGKAT

GitHub tidak hanya menyediakan fitur untuk pengembang software atau aplikasi yang mandiri atau pengembang proyek pribadi, namun terdapat fitur kolaborasi yang memungkinkan untuk pengembangan software secara open source. Hal ini memungkinkan setiap orang dapat berkontribusi dalam suatu proyek.

Pada proyek pengembangan software yang melibatkan banyak orang (tim), tim tersebut tidak hanya akan menyimpan sendiri-sendiri repository(repo) proyeknya. Semua orang dalam tim yang terlibat dalam pengkodean (*coding*) akan menyimpan repository lokal di komputernya masing-masing. Setelah itu, akan dilakukan penggabungan ke repository inti atau *remote*. Biasanya akan ada repository pusat atau untuk menyimpan *source code* yang sudah digabungkan (*merge*) dari beberapa orang.



Gambar 1.1 Ilustrasi Github kolaborasi

Gambar 1.1 menunjukkan para developer dalam suatu tim memiliki repository masing-masing yang ada diperangkatnya namun semua repo tersebut berpusat pada suatu repo yang dinamakan remote repo.

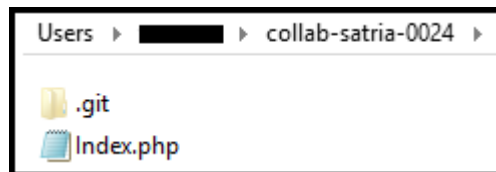


### C. PRAKTIK

1. Membuat repositori local dengan format **git init -collab-nama-4digitakhirnim**

```
@redvelvet MINGW64 ~  
$ git init collab-satria-0024  
Initialized empty Git repository in C:/Users/tamvan/collab-satria-0024/.git/  
  
@redvelvet MINGW64 ~  
$ cd collab-satria-0024/
```

2. Mengisikan repositori local dengan file Index.php yang ada di alamat <https://github.com/masihsendiri999/CollaborationTraining> cukup didonlot aja tidak perlu diclone projek. Hanya Index.php yang dibutuhkan.



3. Sebelumnya lakukan perintah git config agar semua perubahan yang telah Anda lakukan dapat Anda claim bahwa perubahan yang dikerjakan Adalah Authornya. **(INI LANGKAH PENTING! Bila Anda melewatinya dan apabila Anda menggunakan perangkat teman Anda. Bisa jadi akun git confignya masih milik teman Anda. Meski Anda berhasil mengubah nantinya kontribusi Anda tidak ada. Karena yang merubah masih atas nama teman Anda sebelumnya atau yang pertama melakukan git config.)**

```
@redvelvet MINGW64 ~/collab-satria-0024 (master)  
$ git config --global user.name "chessaAditirta"  
  
@redvelvet MINGW64 ~/collab-satria-0024 (master)  
$ git config --global user.email "chessatirta@gmail.com"
```

4. Memasukkan file tersebut ke repo local seperti GitHub Up Files dengan perintah git add . atau git add \* hingga git commit. **(Catatan : Tidak perlu sampai git push karena ini repo local Anda)**  
Memeriksa isi file dengan ls, git status sebelum git add dan git add.

```

@redvelvet MINGW64 ~/collab-satria-0024 (master)
$ ls
Index.php

@redvelvet MINGW64 ~/collab-satria-0024 (master)
$ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        Index.php

nothing added to commit but untracked files present (use "git add" to track)

@redvelvet MINGW64 ~/collab-satria-0024 (master)
$ git add .

```

Git status setelah git add, git commit (komentar commit bebas tidak harus sama dengan contoh di bawah) dan git status setelah git commit.

```

@redvelvet MINGW64 ~/collab-satria-0024 (master)
$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   Index.php

@redvelvet MINGW64 ~/collab-satria-0024 (master)
$ git commit -m "1.1.2:commit repo lokal"
[master (root-commit) e7ba9a6] 1.1.2:commit repo lokal
1 file changed, 4 insertions(+)
create mode 100644 Index.php

@redvelvet MINGW64 ~/collab-satria-0024 (master)
$ git status
On branch master
nothing to commit, working directory clean

```

5. Menambahkan remote (repositori pusat) repo local dengan format :  
git remote add -nama-remote -alamatURL-remote.  
Contoh :  
git remote add upstream <https://github.com/masihsendiri999/CollaborationTraining>

```
@redvelvet MINGW64 ~/collab-satria-0024 (master)
$ git remote add upstream https://github.com/masihsendiri999/collaborationTraining
```

Mengecek remote yang tersedia dengan perintah **git remote -v**

```
@redvelvet MINGW64 ~/collab-satria-0024 (master)
$ git remote -v
upstream      https://github.com/masihsendiri999/collaborationTraining (fetch)
upstream      https://github.com/masihsendiri999/collaborationTraining (push)
```

6. Membuat perubahan pada Index.php dengan menambahkan nama dan nim.

```
<?php
echo "<h2> Selamat berlatih menjadi kolaborator </h2>";
echo "<h3>Isi Nama dan Nim Anda di bawah baris code ini</h3>";
echo "<h3>Ng. Satria Utomo WP(155610024)</h3>"
?>
```

7. Menambahkan perubahan pada repositori lokal dengan mengamati git status sebelum melakukan git add, melakukan perintah git add dan mengamati perintah git status setelah git add tersebut.

```
@redvelvet MINGW64 ~/collab-satria-0024 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   Index.php

no changes added to commit (use "git add" and/or "git commit -a")

@redvelvet MINGW64 ~/collab-satria-0024 (master)
$ git add .

tamvan@redvelvet MINGW64 ~/collab-satria-0024 (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   Index.php
```


8. Melakukan git commit setelah merubah Index.php dan git log untuk memeriksa commit terakhir.

```

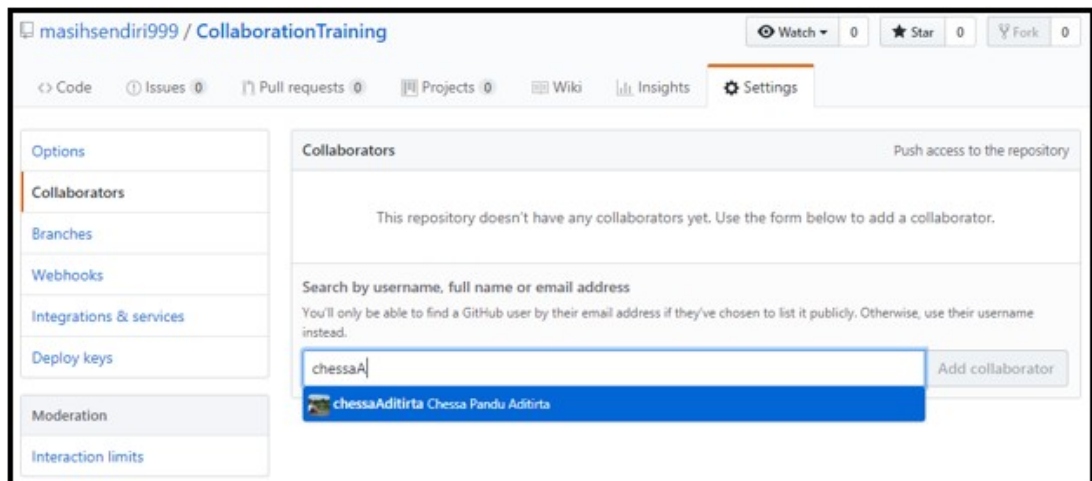
[redacted]@redvelvet MINGW64 ~/collab-satria-0024 (master)
$ git commit -m "Mengedit file Index.php"
[master ca61ebe] Mengedit file Index.php
1 file changed, 1 insertion(+)

[redacted]@redvelvet MINGW64 ~/collab-satria-0024 (master)
$ git log --oneline
ca61ebe Mengedit file Index.php
e7ba9a6 1.1.2:commit repo lokal

```

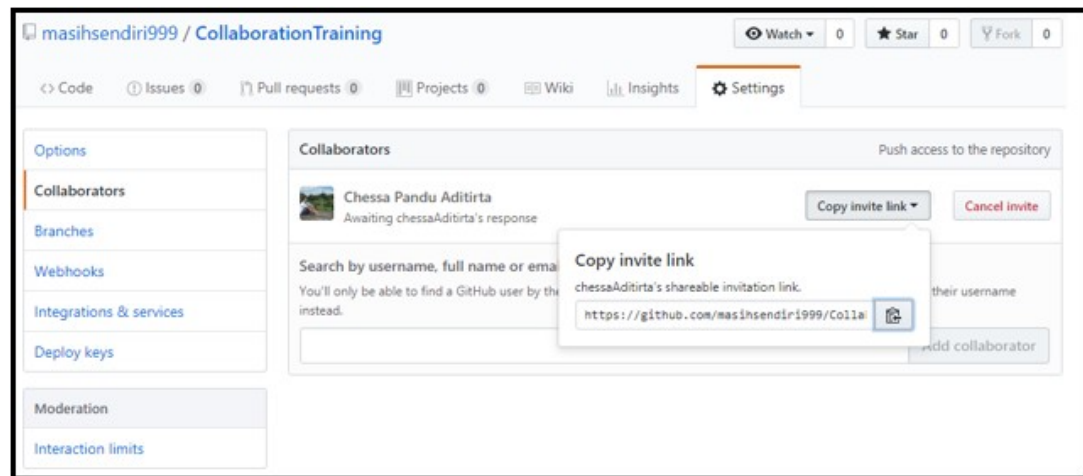
9. Memberikan Username atau alamat email Anda kepada pemilik repository CollaborationTraining yaitu <https://github.com/masihsendiri999/> agar Anda diberi izin untuk Push ke repo tersebut. Langkah ini artinya dijalankan oleh pemilik repository remote (pusat) untuk menambahkan collaborators(kolaborator) dengan alamat email atau username GitHub. Dan pemilik dapat menambahkan dengan cara membuka tab setting  collaborators pada repositorinya. Contoh alamat <https://github.com/masihsendiri999/ CollaborationTraining/settings/collaboration> .

Contoh cara menambahkan (**ingat yang menambahkan pemilik repository**) :

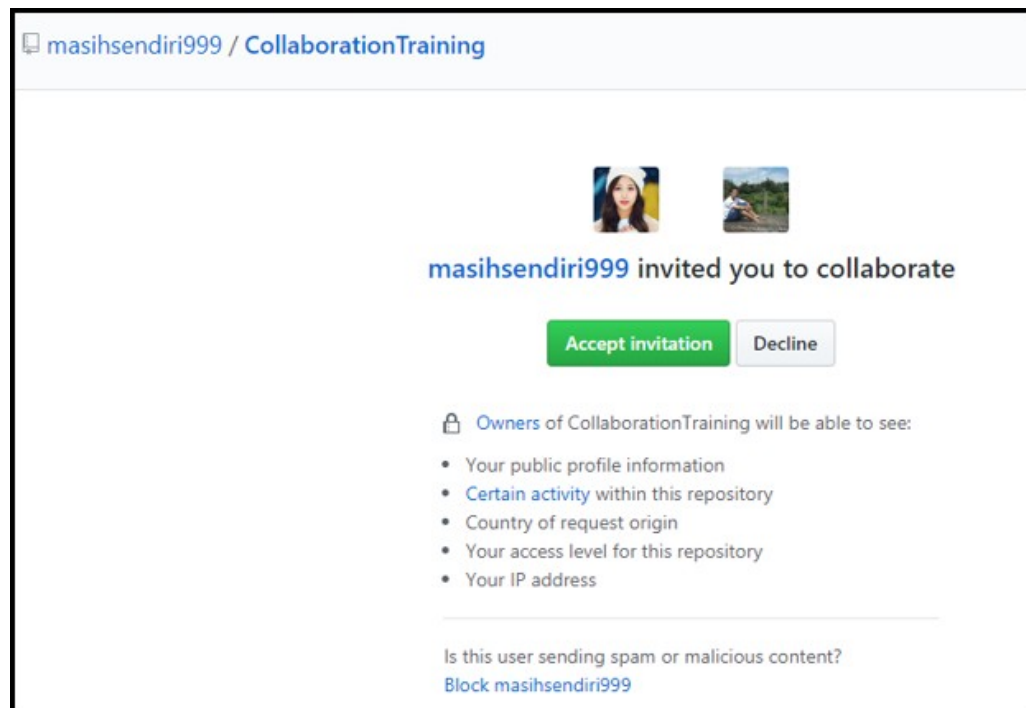


Keterangan : pemilik repo masihsendiri999 dan kolaborator chessaAditirta

10. Setelah Anda ditambahkan pemilik repo pusat(masihsendiri999) sebagai kolanorator , maka pemilik dapat membagikan link invitation kepada Anda. Contoh alamat url invitations nya <https://github.com/masihsendiri999/CollaborationTraining/invitations> Dibawah ini tampilan yang ada di akun **pemilik**(masihsendiri999) untuk memberikan link invitations tersebut.



Invitations link bila anda membuka pada akun github Anda. Dibawah ini adalah invitations yang dilihat dari akun chessaAditirta sebagai **Kolaborator** yang diundang tadi.





11. Setelah Anda terima invitations linknya, berikutnya membuka git Bash dan menjalankan perintah git pull untuk mengambil repository pusat dan digabungkan dengan repo lokal Anda.

```
@redvelvet MINGW64 ~/collab-satria-0024 (master)
$ git pull upstream master
From https://github.com/masihsendiri999/CollaborationTraining
 * branch                master       -> FETCH_HEAD
Updating e7ba9a6..3c7d164
Fast-forward
 Index.php | 2 ++
 README.md | 2 ++
 2 files changed, 4 insertions(+)
 create mode 100644 README.md
```

12. Menjalankan perintah git merge untuk menggabungkan repo lokal Anda dengan repo pusat. Lalu, Perbedaan dengan perintah git pull sebelumnya? Jawabannya sama, Namun git pull hanya sampai pada mengambil dan menggabungkan file-file code lokal dan code pusat. Sedangkan perintah merge dapat melakukan auto-merging dimana bila terdapat satu file diedit oleh dua atau lebih kolaborator berbeda dapat dilihat apakah ada konflik atau tidak bila ada maka diperbaiki pada file tersebut. Dan tambahan parameter -allow-unrelated-histories digunakan karena sumber repo lokal dan repo pusat tidak sama maupun berhubungan(karena repo lokal tidak dikenali hubungannya dengan repo pusat).

Format Perintah: **git merge master upstream/master --allow-unrelated-histories**

```
@redvelvet MINGW64 ~/collab-satria-0024 (master)
$ git merge master upstream/master --allow-unrelated-histories
Auto-merging Index.php
CONFLICT (add/add): Merge conflict in Index.php
Automatic merge failed; fix conflicts and then commit the result.
```

13. Dari perintah merge sebelumnya terdapat keterangan terdapat konflik pada Index.php untuk lebih jelasnya Anda lihat pada perintah git status dan perbaiki Index.php

```
@redvelvet MINGW64 ~/collab-satria-0024 (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")

Changes to be committed:
  new file:   README.md

Unmerged paths:
  (use "git add <file>..." to mark resolution)

    both added:   Index.php
```

Membuka file Index.php dan menghapus baris yang terdapat kata HEAD dan upstream/master.

```
<?php
echo "<h2> Selamat berlatih menjadi kolaborator </h2>";
echo "<h3>Isi Nama dan Nim Anda di bawah baris code ini</h3>";
<<<<<<< HEAD
echo "<h3>Ng. Satria Utomo WP(155610024)</h3>"
=====
>>>>>> upstream/master
?>|
```

File Index.php setelah dihapus.

```
<?php
echo "<h2> Selamat berlatih menjadi kolaborator </h2>";
echo "<h3>Isi Nama dan Nim Anda di bawah baris code ini</h3>";
echo "<h3>Ng. Satria Utomo WP(155610024)</h3>"

?>|
```

14. Melakukan git add . , git status dan git commit kembali setelah melakukan perbaikan file Index.php setelah merge sebelumnya.

```
@redvelvet MINGW64 ~/collab-satria-0024 (master|MERGING)
$ git add .

@redvelvet MINGW64 ~/collab-satria-0024 (master|MERGING)
$ git status
On branch master
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
      modified:   Index.php
      new file:   README.md

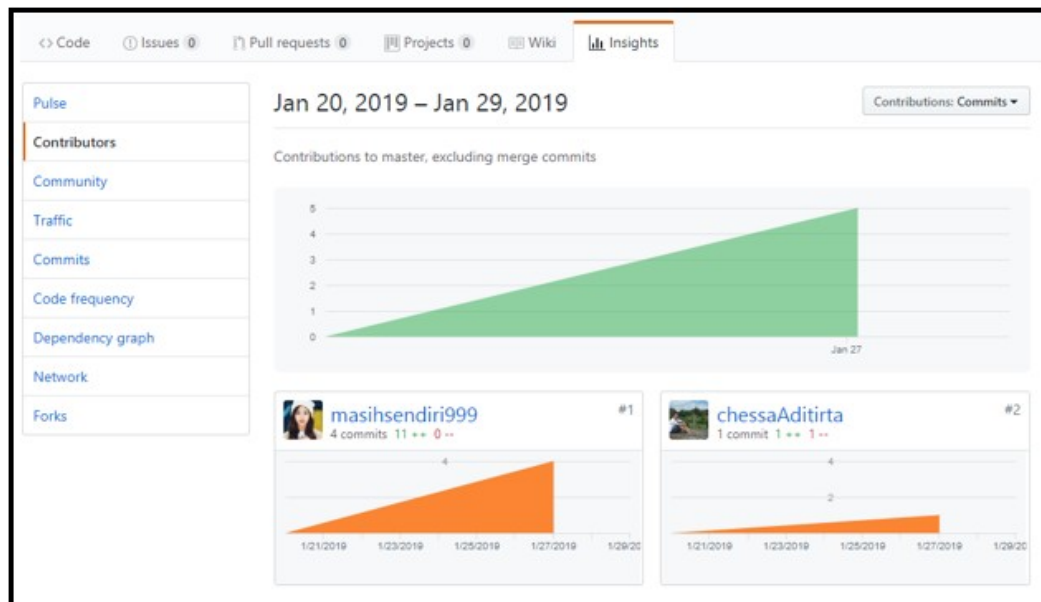
@redvelvet MINGW64 ~/collab-satria-0024 (master|MERGING)
$ git commit -m "perbaikan Index.php setelah merge"
[master 3c7d164] perbaikan Index.php setelah merge
```



15. Kemudian yang terakhir adalah git push dengan Format : **git push upstream master**.

```
@redvelvet MINGW64 ~/collab-satria-0024 (master)
$ git push upstream master
Fatal: HttpRequestException encountered.
Username for 'https://github.com': chessaAditirta
Counting objects: 8, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 799 bytes | 0 bytes/s, done.
Total 8 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/masihsendiri999/collaborationTraining
dce7479..3c7d164 master -> master
```

Masuk ke akun github anda kembali di browser, setelah itu buka <https://github.com/masihsendiri999/CollaborationTraining/graphs/contributors> untuk melihat kontribusi anda diprojek tersebut.



#### D. TUGAS

1. Buat kelompok maksimal 4 Orang(sendiri bukan merupakan kelompok meski menggunakan 2 akun gitHub).
2. Buat aplikasi web statis menurut **imajinasi masing-masing** memiliki beberapa halaman web dan salah satu halamannya tersusun atas nama, nim serta hal yang menunjukkan profil **kelompok** seperti halaman web mengenai tentang , profil atau About pada umumnya. (Bebas memakai bahasa atau framework apapun namun web CMS atau **Content Management System** tidak diijinkan). Dengan ketentuan 1 orang sebagai pemilik projek yang bertugas membuat repositori serta menggabungkan projek tersebut dengan temannya dan sisanya menjadi kolaborator seperti yang ada dipraktik. Contoh bila suatu kelompok terdiri atas A, B, C dan D. maka A membuat repository serta sebagai pemilik projeknya. B membuat halaman utama, C membuat Halaman kedua dan D membuat Halaman ketiga.
3. Tulis alamat repositori kelompok. (Tidak Perlu Screen Shoot tahapan cukup tulis alamat repo yg dapat diakses)

# GOOGLE APPS ENGINE

## A. TUJUAN

- Mahasiswa memahami penggunaan **Google Apps Engine**.
- Mahasiswa dapat menggunakan membuat aplikasi menggunakan **Google Apps Engine**.

## B. TEORI SINGKAT

### Definisi

Google App Engine merupakan sebuah Paas (Platform as a Service) yang menawarkan layanan cloud computing secara gratis. App Engine merupakan platform yang memungkinkan pengembang untuk membuat dan menjalankan aplikasi-aplikasi web dengan fasilitas hosting di server Google. Google App Engine mendukung aplikasi yang ditulis dalam bahasa pemrograman. Dengan lingkungan runtime Java App Engine, pengguna dapat membangun aplikasi yang menggunakan standar teknologi Java, termasuk JVM, Java servlet, dan bahasa pemrograman java. Google App Engine dirancang untuk mempermudah pembangunan aplikasi jaringan dengan memanfaatkan infrastruktur dan teknologi yang Google biasa gunakan untuk produk-produk mereka sendiri, seperti Gmail, Google+, dan Google Search.

### Manfaat

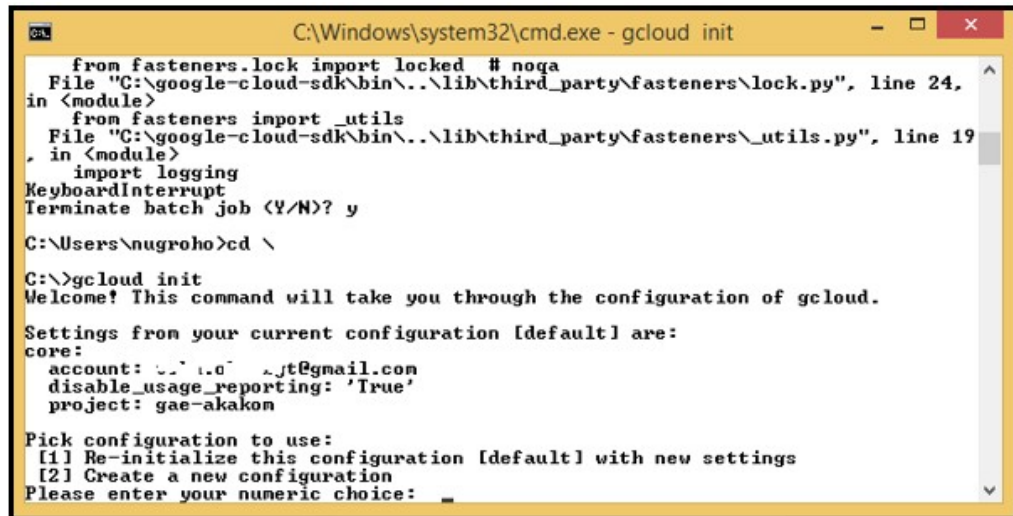
- Google App Engine ini adalah layanan ini disediakan secara gratis.
- penggunaannya tidak terbatas (unlimited).
- layanan ini juga terintegrasi dengan berbagai layanan lainnya dari Google
- jika aplikasi web anda membutuhkan autentifikasi user, anda bisa memanfaatkan Account Google sebagai autentifikasi use

## C. PRAKTIK

1. Memastikan Anda sudah menginstall python 2.7 dan gcloud SDK. Apabila belum kunjungi alamat <https://www.python.org/downloads/> pilih python 2.7 dan <https://cloud.google.com/> . Masuk pada Command prompt, lalu tuliskan perintah python untuk mengecek instalasi python dan gcloud -v untuk mngetahui gcloud sdk sudah terinstall.

```
C:\Windows\system32>cd \
C:\>python
Python 2.7 (r27:82525, Jul  4 2010, 07:43:08) [MSC v.1500 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>>
^C
C:\>gcloud -v
Google Cloud SDK 192.0.0
bq 2.0.29
core 2018.03.02
gsutil 4.28
```

2. Menjalankan gcloud init. Dengan menjalankan **gcloud init**, akan melakukan proses inisiasi awal berupa permintaan ijin aplikasi dengan login akun google anda



```
C:\Windows\system32\cmd.exe - gcloud init

from fasteners.lock import locked # noqa
File "C:\google-cloud-sdk\bin\..\lib\third_party\fasteners\lock.py", line 24,
in <module>
  from fasteners import _utils
File "C:\google-cloud-sdk\bin\..\lib\third_party\fasteners\_utils.py", line 19
, in <module>
  import logging
KeyboardInterrupt
Terminate batch job (Y/N)? y

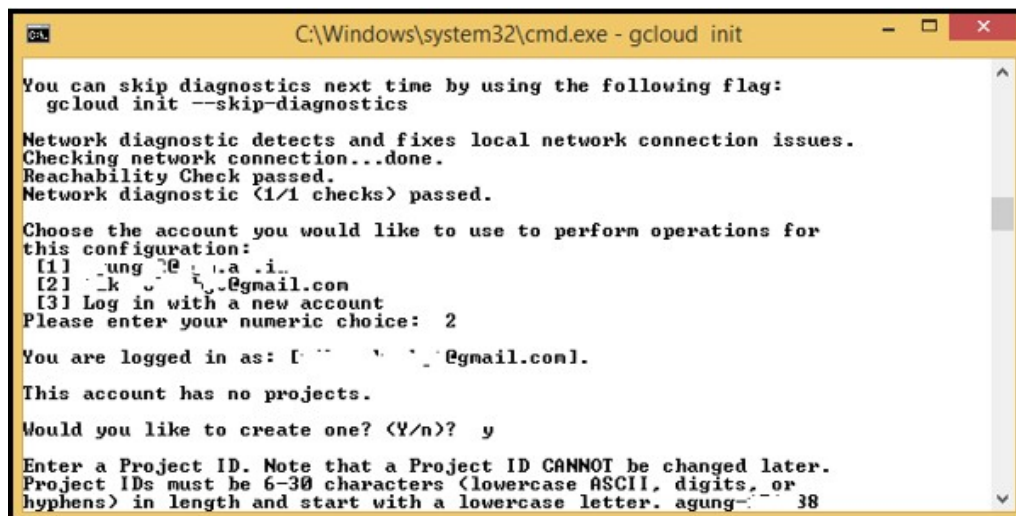
C:\Users\nugroho>cd \

C:\>gcloud init
Welcome! This command will take you through the configuration of gcloud.

Settings from your current configuration [default] are:
core:
  account: nugroho@gmail.com
  disable_usage_reporting: 'True'
  project: gae-akakom

Pick configuration to use:
[1] Re-initialize this configuration [default] with new settings
[2] Create a new configuration
Please enter your numeric choice: _
```

3. Pada git ini dilakukan setup projek apakah ingin menggunakan projek yang sudah ada maupun membuat yang baru



```
C:\Windows\system32\cmd.exe - gcloud init

You can skip diagnostics next time by using the following flag:
gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic (1/1 checks) passed.

Choose the account you would like to use to perform operations for
this configuration:
[1] nugroho@gmail.com
[2] nugroho@gmail.com
[3] Log in with a new account
Please enter your numeric choice: 2

You are logged in as: nugroho@gmail.com.

This account has no projects.

Would you like to create one? (Y/n)? y

Enter a Project ID. Note that a Project ID CANNOT be changed later.
Project IDs must be 6-30 characters (lowercase ASCII, digits, or
hyphens) in length and start with a lowercase letter. agung- 38
```

4. Membuat direktori webtcc/percobaan1 di disk :C (c:\webtcc\percobaan1) dan membuat main.php dan app.yaml

## main.php

```
<?php
echo "<h1>Ujicoba PHP di Google Apps Engine</h1>";
?>
```

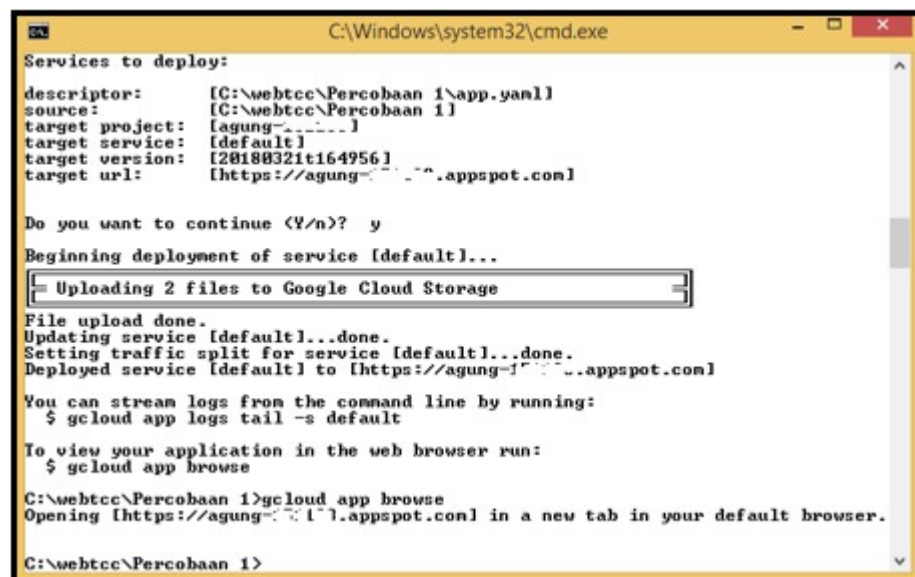
## app.yaml

```
runtime: php55
handlers:
- Url: /*
  script: main.php
```

5. Menjalankan perintah **dev\_appserver.py app.yaml** untuk menjalankan aplikasi di local. Dan masih berada pada direktori webtcc/percobaan1



6. Melakukan config projek untuk persiapan deploy. Dengan Format perintah : **gcloud config set project -namaproject**
7. Menjalankan perintah **gcloud app deploy** untuk melakukan deploy aplikasi dan **gcloud app browse** untuk melihat alamat deploy



8. Hasil gcloud app browse akan otomatis membuka default browser Anda dan menampilkan tampilan berikut.



#### D. TUGAS

1. Buat aplikasi web statis menurut **imajinasi masing-masing** memiliki beberapa halaman web dan salah satu halamannya tersusun atas nama, nim serta hal yang menunjukkan profil saudara seperti halaman web mengenai tentang , profil atau About pada umumnya. (Bebas memakai bahasa atau framework apapun namun web CMS atau **Content Management System** tidak diijinkan).

**Catatan:**

Format nama folder dan nama aplikasi **namatugNim**. Contoh:  
**satriatug155610024**

2. Deploy aplikasi tersebut menggunakan GAE.
3. Tuliskan URL hasil deploy.

## PENGENALAN HEROKU

### A. TUJUAN

- Mahasiswa memahami penggunaan **Heroku**.
- Mahasiswa dapat menggunakan membuat aplikasi menggunakan **Heroku**.

### B. TEORI SINGKAT

Heroku merupakan salah satu layanan *cloud* yang menyediakan layanan *platform as a service* (PaaS) yang mendukung beberapa bahasa pemrograman. Heroku adalah salah satu *platform cloud* pertama yang diciptakan sejak juni 2007, dimana pada awal Heroku hanya mendukung bahasa pemrograman ruby. Namun sekarang Heroku mendukung bahasa pemrograman Java, Node.js, Scala, Clojure, Python, PHP dan Go. Heroku dikatakan sebagai platform polyglot karena memungkinkan pengembang membangun, menjalankan, dan menskala aplikasi dengan cara yang sama di semua bahasa. Pengembang yang akan melakukan deploy aplikasi cukup melakukan konfigurasi aplikasi menurut bahasa pemrograman yang dipakai untuk dikembangkan, dijalankan dan dikelola tanpa kompleksitas membangun dan memelihara infrastruktur untuk pengembangan dan peluncuran aplikasi.

### C. PRAKTIK

1. Mendaftar akun pengguna di Heroku dengan mengakses pranala berikut : <https://signup.heroku.com/>, lalu mengisikan atribut pendaftaran sekaligus mencantumkan email sehingga nanti akan ada konfirmasi pada email untuk konfigurasi password akun Heroku tersebut.



First name \*

Last name \*

Email address \*

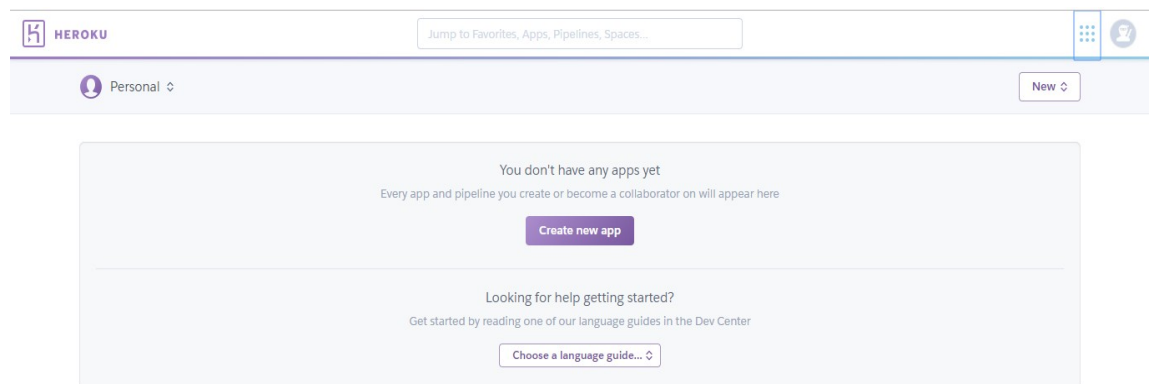
Company name

Role \*

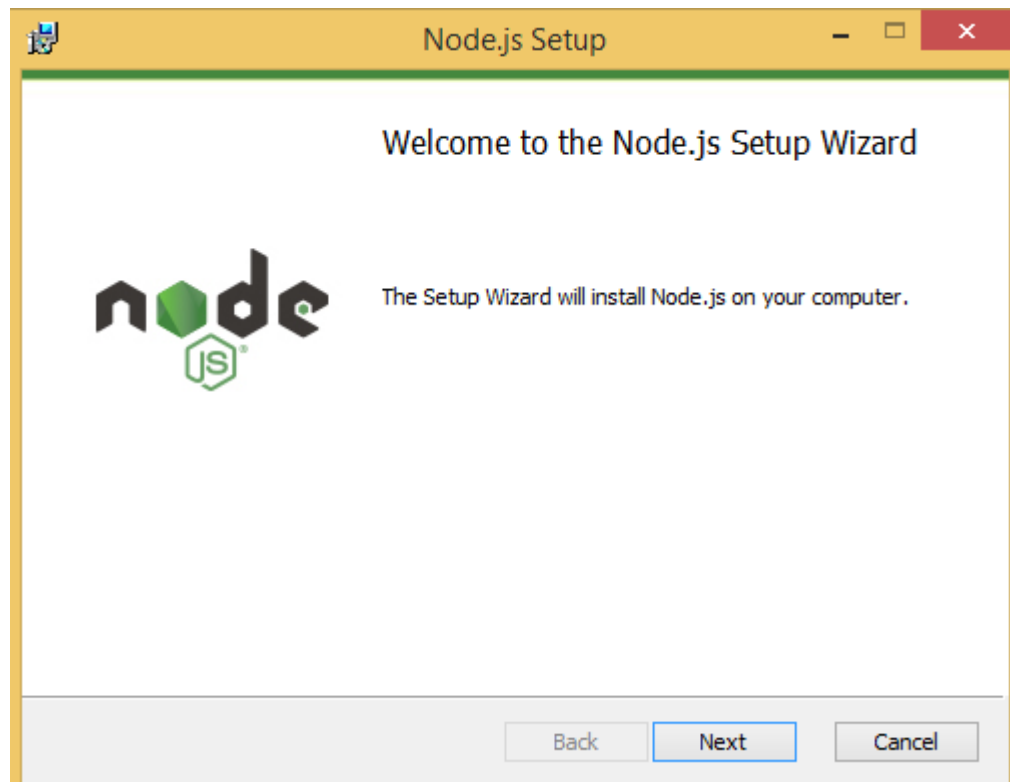
Country \*

Primary development language \*

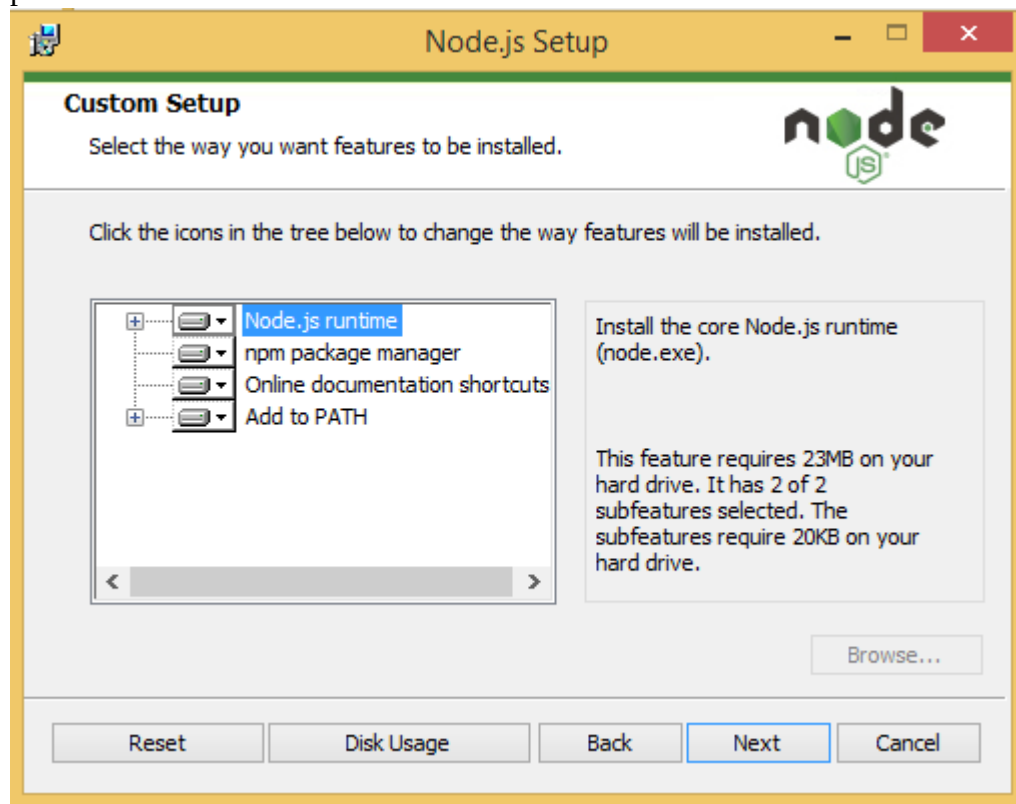
Kemudian terlihat tampilan dashboard Heroku seperti berikut :



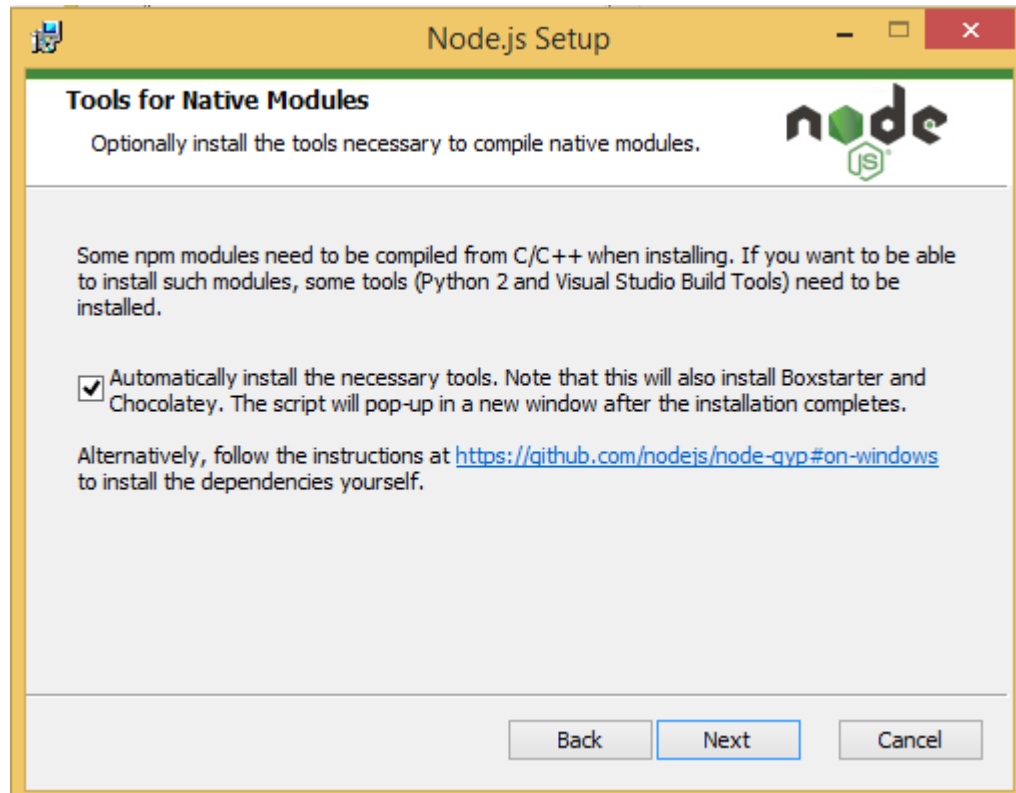
2. Memasang node.js pada perangkat komputer dengan mengunduh file node-v10.13.0-x64.msi. kemudian muncul tampilan berikut:



Berikutnya terdapat agreement klik centang, lalu next. Pada custom setup tidak perlu diubah. Setelah itu klik next.



Kemudian pilih centang pada automatically install the necessary tools nya



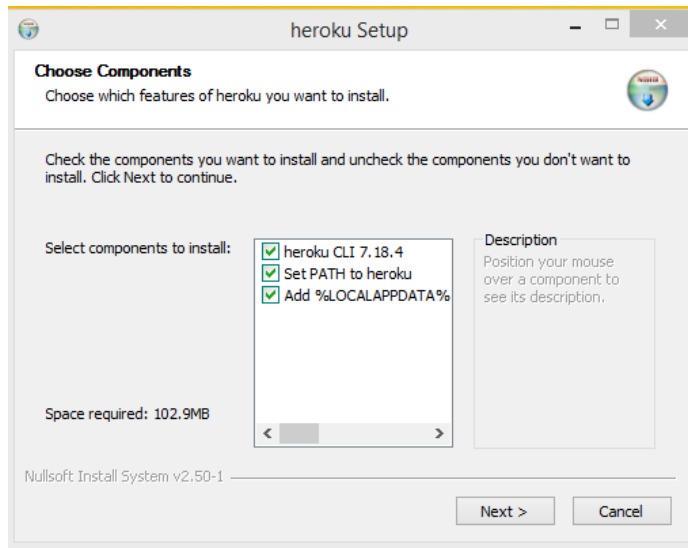
Setelah itu melakukan pengecekan apakah instalasi berjalan atau tidak, dengan perintah di cmd seperti berikut :

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

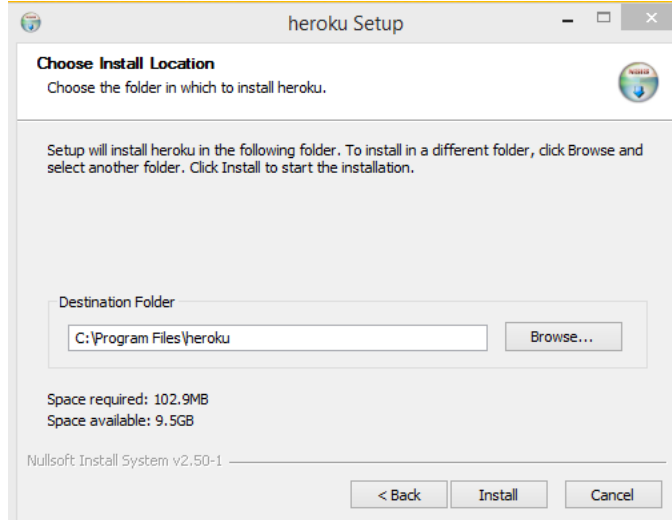
C:\Users\tamvan>node -v
v10.13.0

C:\Users\tamvan>npm -v
6.4.1
```

Melakukan instalasi Heroku pada perangkat computer  
Centang semua komponen yang akan diinstall pada tab choose components



Mengarahkan instalasi pada direktori Program Files yang berada di disk C:



Setelah proses instalasi selesai cek instalasi dengan perintah heroku -v pada cmd

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
```

```
C:\Users\tamvan>heroku -v
heroku/7.18.4 win32-x64 node-v11.0.0
```

### 3. Mengunduh aplikasi web

Clone project dengan git bash dengan perintah berikut:

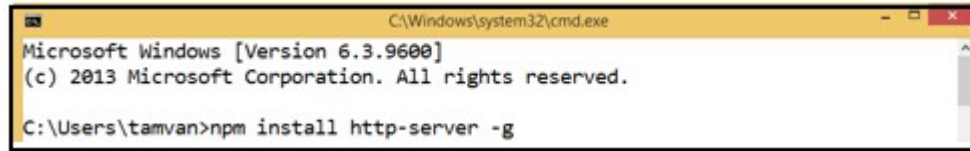
**git clone** [https://github.com/chessaAditirta/web\\_project](https://github.com/chessaAditirta/web_project)

### Catatan :

Ubah nama folder web\_project dengan nama diikuti nim karena nama folder dan nama aplikasi nanti digunakan sebagai subdomain aplikasi di Heroku. Contoh format :

**Chessa155610037**

4. Memasang http-server dengan npm dengan perintah :  
**npm install http-server -g**



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\tamvan>npm install http-server -g
```

5. Mengarahkan cursor menuju ke direktori projek dan menjalankan aplikasi di lokal dengan perintah seperti pada gambar di bawah ini :



```
C:\Users\tamvan>cd chessa155610037

C:\Users\tamvan\chessa155610037>http-server -p 8080 -a localhost -c 0
Starting up http-server, serving ./
Available on:
  http://localhost:8080
Hit CTRL-C to stop the server
```

Penjelasan:

**http-server** adalah potongan code di atas untuk menggunakan fungsi dengan http-server yang sebelumnya diinstalasi dan **-p 8080** adalah port aplikasi yang berjalan. Setelah itu **-a localhost** adalah address atau alamat akses aplikasi. Kemudian yang terakhir **-c 0** yang merupakan cache time dalam satuan detik yang berarti cache time bernilai 0.

6. Melakukan inisiasi npm dengan perintah **npm init -y** yang akan memunculkan file bernama package.json yang berisi nama aplikasi, diskripsi aplikasi, dependensi aplikasi dan lain sebagainya seperti file definisi aplikasi sejenis seperti pom.xml milik maven dan build.gradle milik gradle.

```
C:\Users\tamvan\chessa155610037>npm init -y
Wrote to C:\Users\tamvan\chessa155610037\package.json:

{
  "name": "chessa155610037",
  "version": "1.0.0",
  "description": "Tayo^2 Programming",
  "main": "modernizr-custom.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/chessaAдитirta/web_project.git"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/chessaAдитirta/web_project/issues"
  },
  "homepage": "https://github.com/chessaAдитirta/web_project#readme"
}
```

Lalu, install node static dengan perintah **npm install node-static --save**

```
C:\Users\tamvan\chessa155610037>npm install node-static --save
+ node-static@0.7.11
updated 1 package and audited 6 packages in 2.835s
found 0 vulnerabilities
```

Kemudian cek kembali dengan perintah `npm init -y` maka akan ada tambahan dependensi yaitu node static

```
C:\Users\tamvan\chessa155610037>npm init -y
Wrote to C:\Users\tamvan\chessa155610037\package.json:

{
  "name": "chessa155610037",
  "version": "1.0.0",
  "description": "Tayo^2 Programming",
  "main": "modernizr-custom.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/chessaAditirta/web_project.git"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/chessaAditirta/web_project/issues"
  },
  "homepage": "https://github.com/chessaAditirta/web_project#readme",
  "dependencies": {
    "node-static": "^0.7.11"
  },
  "devDependencies": {}
}
```

7. Membuat index.js untuk mengorganisasi jalannya aplikasi dengan node.js

Sebelumnya tambahkan perintah **"start": "node index.js"**, dan hapus beberapa code sehingga pada package.json akan terlihat seperti gambar di bawah:

```
C:\Users\tamvan\chessa155610037>npm init -y
Wrote to C:\Users\tamvan\chessa155610037\package.json:

{
  "name": "chessa155610037",
  "version": "1.0.0",
  "description": "Tayo^2 Programming",
  "main": "modernizr-custom.js",
  "scripts": {
    "start": "node index.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "node-static": "^0.7.11"
  },
  "devDependencies": {}
}
```

Kemudian buat file index.js dengan code seperti dibawah,

```
var static = require('node-static');
var file = new static.Server();

require('http').createServer(function(request, response) {
  request.addListener('end', function() {
    file.serve(request, response);
  }).resume();
}).listen(process.env.PORT || 5000);
```

Terakhir jalankan perintah **node index.js** pada cmd untuk menjalankan web statis dengan index.js. Jika tidak menemukan error log buka pada <http://localhost:5000> untuk tes aplikasi di local.

8. Menghubungkan direktori pada git lalu git local dihubungkan ke Heroku  
Sebelum melakukan inisiasi git hapus folder .git bila tidak menemukan temuan dengan show hidden folder pada view tab.



Menjalankan perintah **git init** untuk inisialisasi git untuk menggantikan .git yang tadi dihapus karena milik repository sebelumnya.

```
C:\Users\tamvan\chessa155610037>git init  
Initialized empty Git repository in C:/Users/tamvan/chessa155610037/.git/
```

Menjalankan perintah **heroku login** untuk masuk ke akun Heroku masing-masing melalui cmd

```
C:\Users\tamvan\chessa155610037>heroku login  
heroku: Enter your login credentials  
Email: masihsendiri999@gmail.com  
Password: *****  
Logged in as masihsendiri999@gmail.com
```

Membuat aplikasi diheroku dengan format perintah **heroku create name-app**.

Seperti perintah sebelumnya nama folder dan nama aplikasi yang digunakan adalah format namaNim maka perintahnya menjadi seperti berikut:

```
C:\Users\tamvan\chessa155610037>heroku create chessa155610037  
Creating ☐ chessa155610037... done  
https://chessa155610037.herokuapp.com/ | https://git.heroku.com/chessa155610037.git
```

Penjelasan :

Ada dua alamat yang tertera pada hasil pembuatan aplikasi dimana yang kiri adalah domain aplikasi dan kanan adalah alamat repo git yang terintegrasi ke Heroku. Dengan alamat repo tersebut dapat dimungkinkan untuk melakukan git clone app dengan alamat tersebut dengan syarat telah login ke Heroku atau perintah Heroku login. Pembuktian dapat dilihat bila saudara menjalankan perintah **git remote -v**.

```
C:\Users\tamvan\chessa155610037>git remote -v  
heroku https://git.heroku.com/chessa155610037.git (fetch)  
heroku https://git.heroku.com/chessa155610037.git (push)
```

Penjelasan:

Dimana menunjukkan push dan fetch beralamat di git.heroku, maka jangan takut kehilangan kode program bila setelah di push nanti karena sudah tersimpan di cloud. Saudara tinggal melakukan clone app dengan git clone alamat tersebut.

## 9. Push proyek ke Heroku sekaligus deploy aplikasi

Salah satu keunggulan Heroku adalah mendeploy aplikasi dan push code programnya dengan sekali push karena git local saudara telah terintegrasi ke akun Heroku saudara. Oleh karena itu istilah di Heroku bukan deploy apps melainkan push apps. Sebelum melakukan push perlu diingat bergitu push apps dijalankan Heroku pertama kali akan menjalankan perintah npm install dimana direktori aplikasi kita akan ditambahkan direktori dengan nama node\_modules. Mekanisme tersebut terjadi karena kita menggunakan ekosistem node.js dimana node\_modules adalah direktori yang berisi plugin, ekstensi atau kebutuhan resource aplikasi node.js. Hal ini seperti mekanisme SDK pada pemrograman Android studio. Singkatnya kita perlu melakukan ignore terhadap direktori node\_modules tersebut agar aplikasi web statis bisa dijalankan untuk itu perlu membuat file .gitignore. seperti kode di bawah:

**/node\_modules**

Git add dan git commit

```
C:\Users\tamvan\chessa155610037>git add .
warning: LF will be replaced by CRLF in about.html.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in modernizr-custom.js.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in package-lock.json.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in package.json.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in styles/main.css.
The file will have its original line endings in your working directory.
```

Kemudian lakukan perintah commit dengan format

**git commit -m “isi bebas lepas tanpa batas”**

```
C:\Users\tamvan\chessa155610037>git commit -m "1.1.3:Web apps push to Heroku"
[master (root-commit) c98d0ae] 1.1.3:Web apps push to Heroku
```

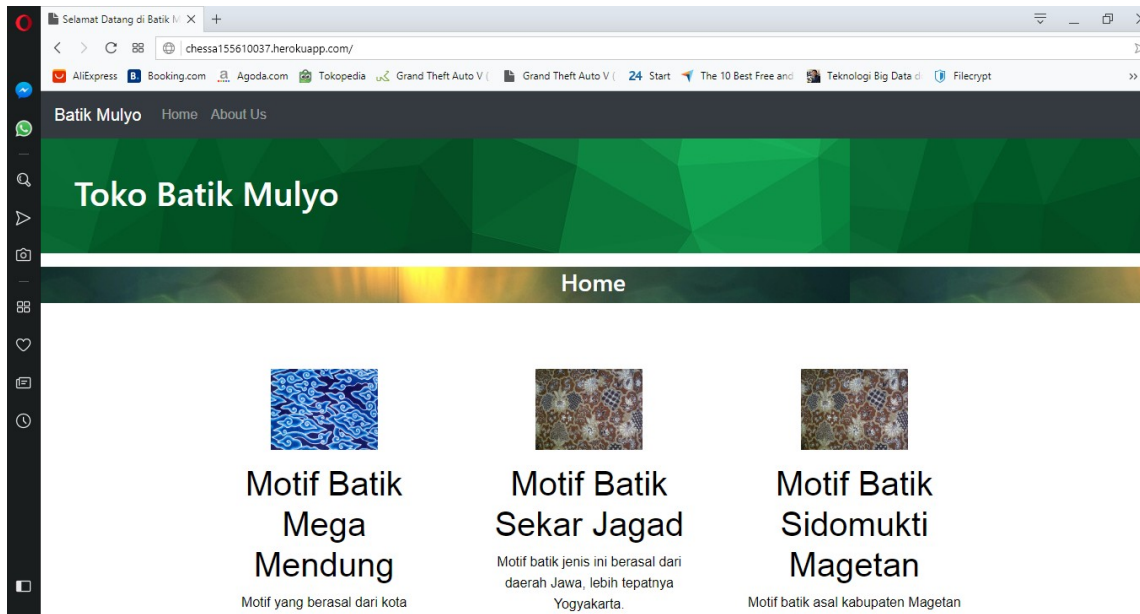
Terakhir memanjatkan do'a dan menjalankan perintah dengan format **git push heroku master**

**catatan:**

Jangan mengetik perintah git push origin master hayoo... ingat nama remote kita adalah heroku coba kembali kehalaman yang membahas **git remote - v**.

```
C:\Users\tamvan\chessa155610037>git push heroku master
Counting objects: 28, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (25/25), done.
Writing objects: 100% (28/28), 2.91 MiB | 243.00 KiB/s, done.
Total 28 (delta 0), reused 0 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Node.js app detected
remote:
remote: -----> Creating runtime environment
remote:
remote:           NPM_CONFIG_LOGLEVEL=error
remote:           NODE_ENV=production
remote:           NODE_MODULES_CACHE=true
remote:           NODE_VERBOSE=false
...
remote: -----> Caching build
remote:           - node_modules
remote:
remote: -----> Pruning devDependencies
remote:           audited 6 packages in 0.806s
remote:           found 0 vulnerabilities
remote:
remote:
remote: -----> Build succeeded!
remote: -----> Discovering process types
remote:           Procfile declares types      -> (none)
remote:           Default types for buildpack -> web
remote:
remote: -----> Compressing...
remote:           Done: 20.9M
remote: -----> Launching...
remote:           Released v3
remote:           https://chessa155610037.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/chessa155610037.git
 * [new branch]      master -> master
```

Menjalankan perintah **heroku open** untuk membuka aplikasi pada browser.



Penjelasan :

Aplikasi yang telah di push ke Heroku dan memperoleh alamat <https://chessa155610037.herokuapp.com/> .

#### D. TUGAS

1. Buat aplikasi web statis menurut **imajinasi masing-masing** memiliki beberapa halaman web dan salah satu halamannya tersusun atas nama, nim serta hal yang menunjukkan profil saudara seperti halaman web mengenai tentang , profil atau About pada umumnya.

##### Catatan:

Format nama folder dan nama aplikasi **namatugNim**. Contoh:  
**chessatug155610037**

2. Deploy aplikasi tersebut di Heroku.
3. Screenshoot file index.js, package.json, aplikasi hasil deploy ke Heroku dan lampirkan alamat deploy aplikasi Heroku saudara di laporan.

## DOCKER

### A. TUJUAN

1. Mahasiswa mampu melakukan instalasi docker.
2. Mahasiswa mampu memahami penggunaan dasar docker.

### B. TEORI SINGKAT

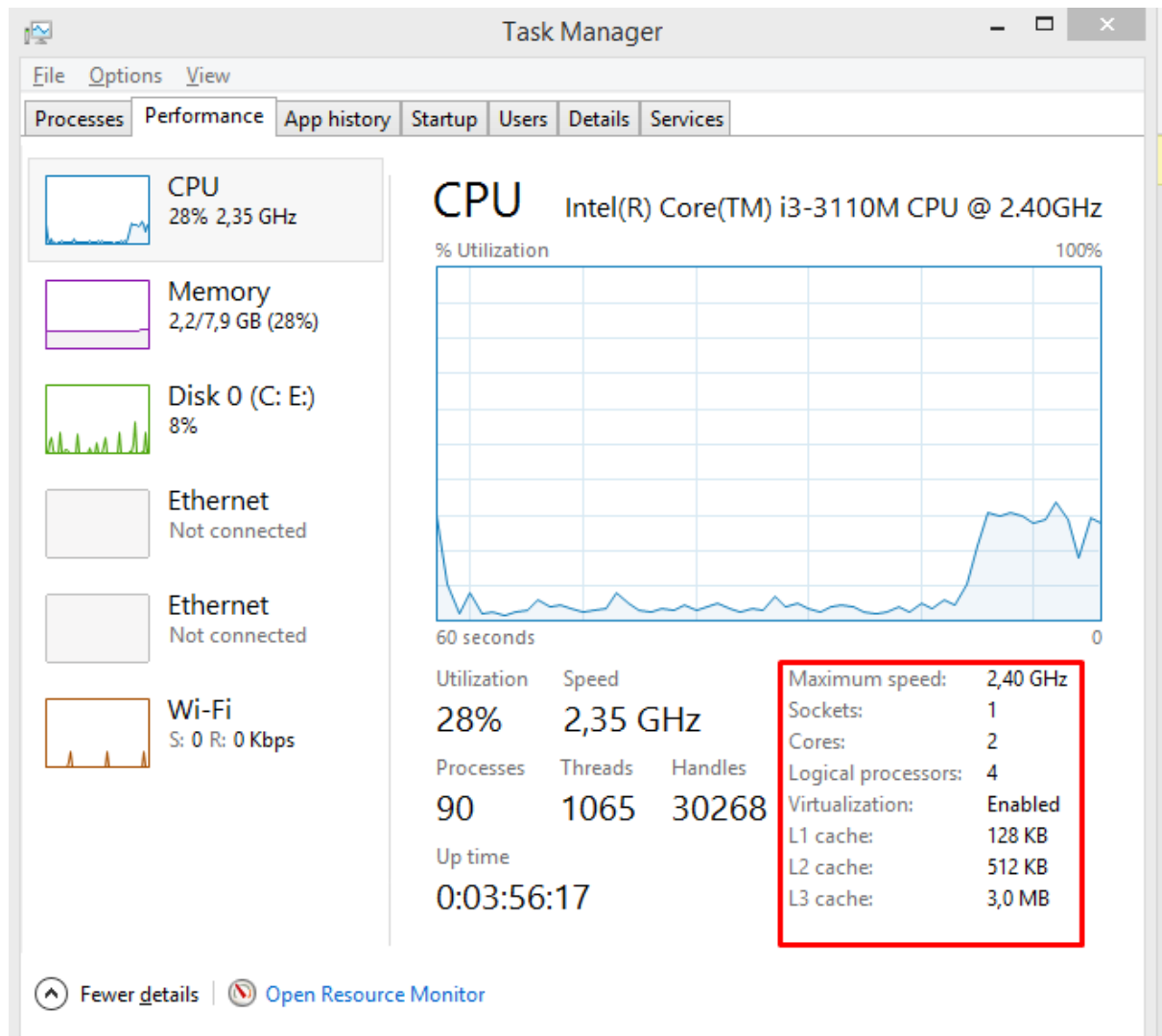
Docker adalah sebuah aplikasi yang bersifat open source yang berfungsi sebagai wadah/container untuk mengepak/memasukkan sebuah software secara lengkap beserta semua hal lainnya yang dibutuhkan oleh software tersebut dapat berfungsi. Pengaturan software beserta file/hal pendukung lainnya akan menjadi sebuah Image (istilah yang diberikan oleh docker). Kemudian sebuah instan dari Image tersebut kemudian disebut Container.

Aplikasi docker bisa dijalankan terutama pada sistem operasi linux seperti Debian, Ubuntu, Redhat, Centos dan banyak distro linux lainnya. Sedangkan untuk Mac OS dan Win OS untuk keperluan pemasangan docker adalah menggunakan Docker Toolbox.

### C. PRAKTIK

#### **Instalasi Docker Toolbox**

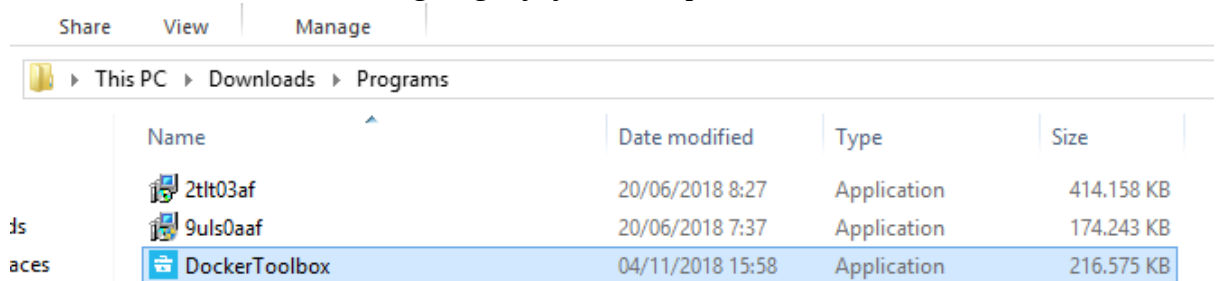
Sebelum melakukan instalasi, perlu kita ketahui untuk memasang aplikasi docker toolbox memerlukan sebuah teknologi yang dinamakan virtualization. Untuk mengetahui apakah komputer memiliki teknologi virtualization bisa dilihat pada task manager. Kemudian pada task manager buka pada tab *Performance*. Maka akan terlihat teknologi virtualization yang ada pada komputer seperti gambar dibawah ini



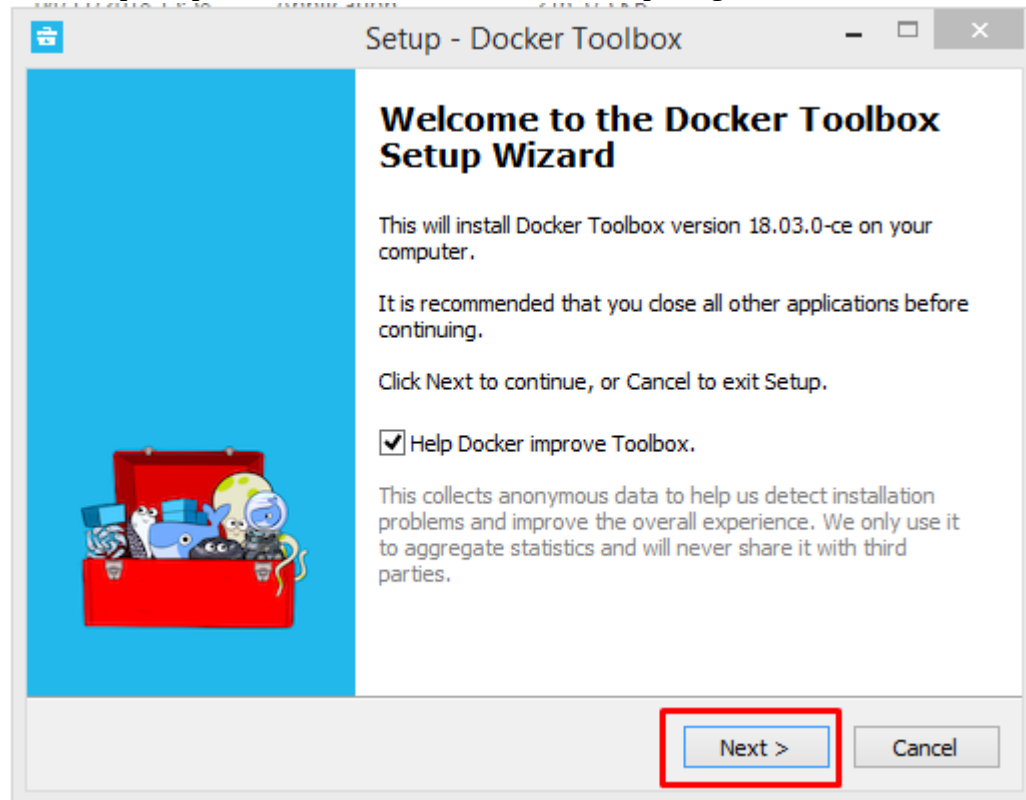
Seperti gambar diatas bahwa terdapat kotak merah sebelah kanan yang menunjukkan teknologi Virtualization support pada komputer. Setelah itu download docker toolbox di halaman web [https://docs.docker.com/toolbox/toolbox\\_install\\_windows/](https://docs.docker.com/toolbox/toolbox_install_windows/)



Setelah selesai terdownload langsung saja jalankan aplikasi docker toolbox

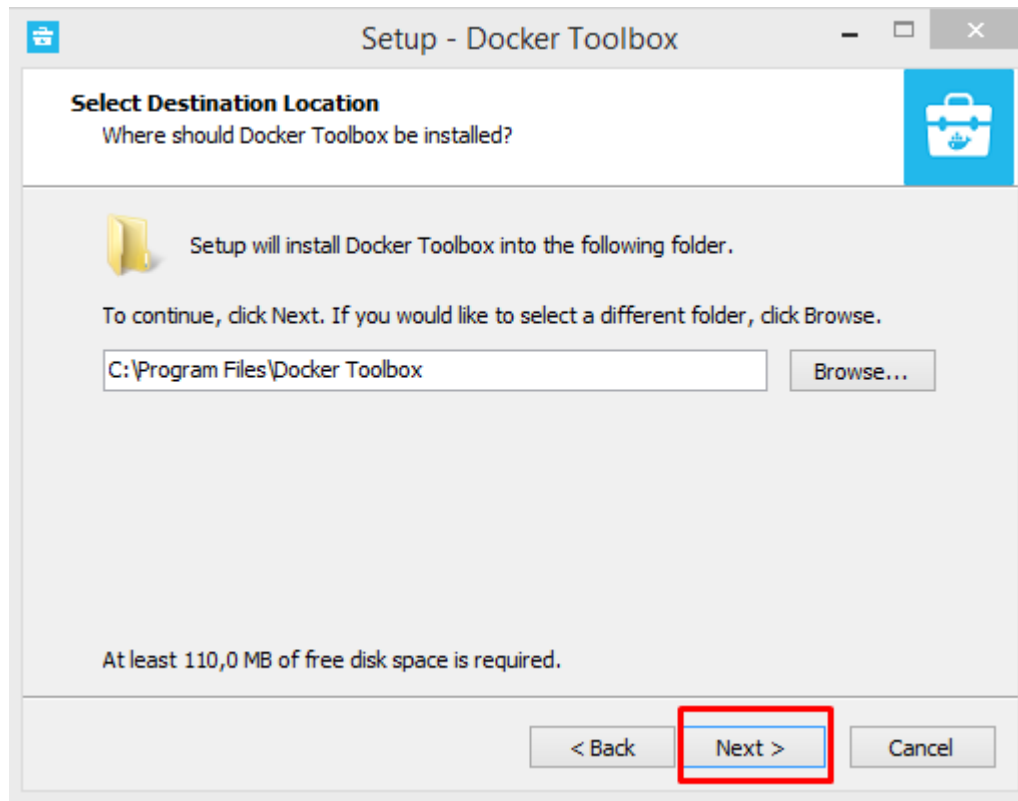


Tampilan pertama instalasi docker toolbox seperti gambar dibawah ini

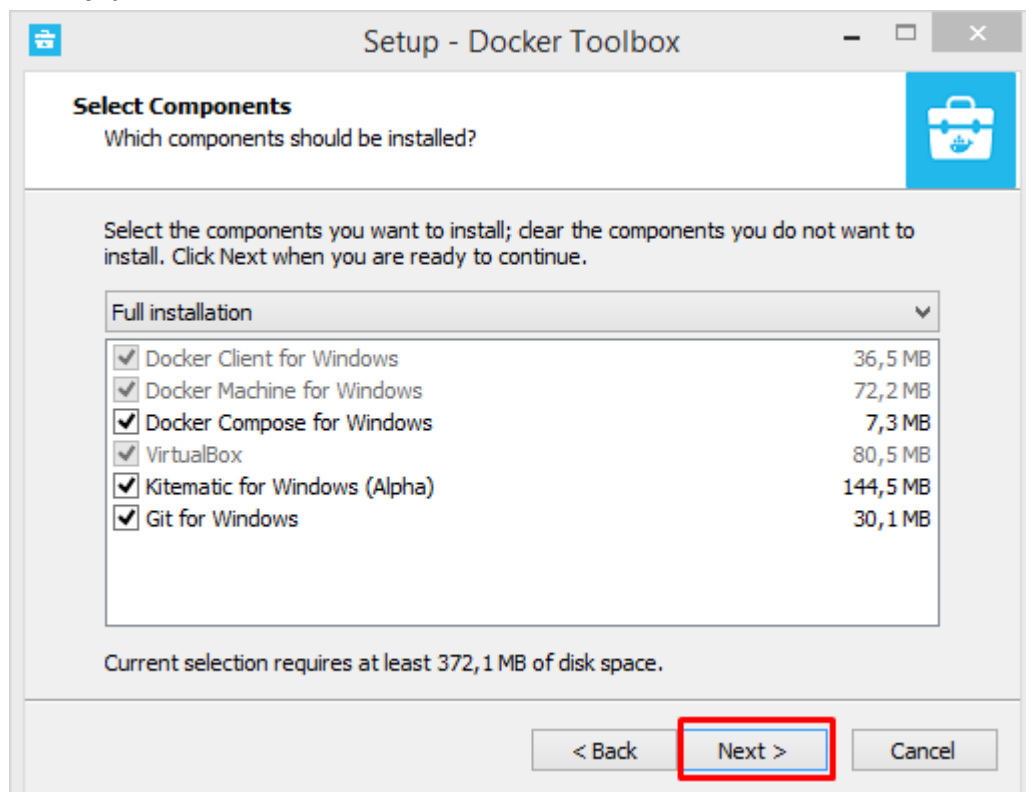


Klik Next untuk melanjutkan instalasi.

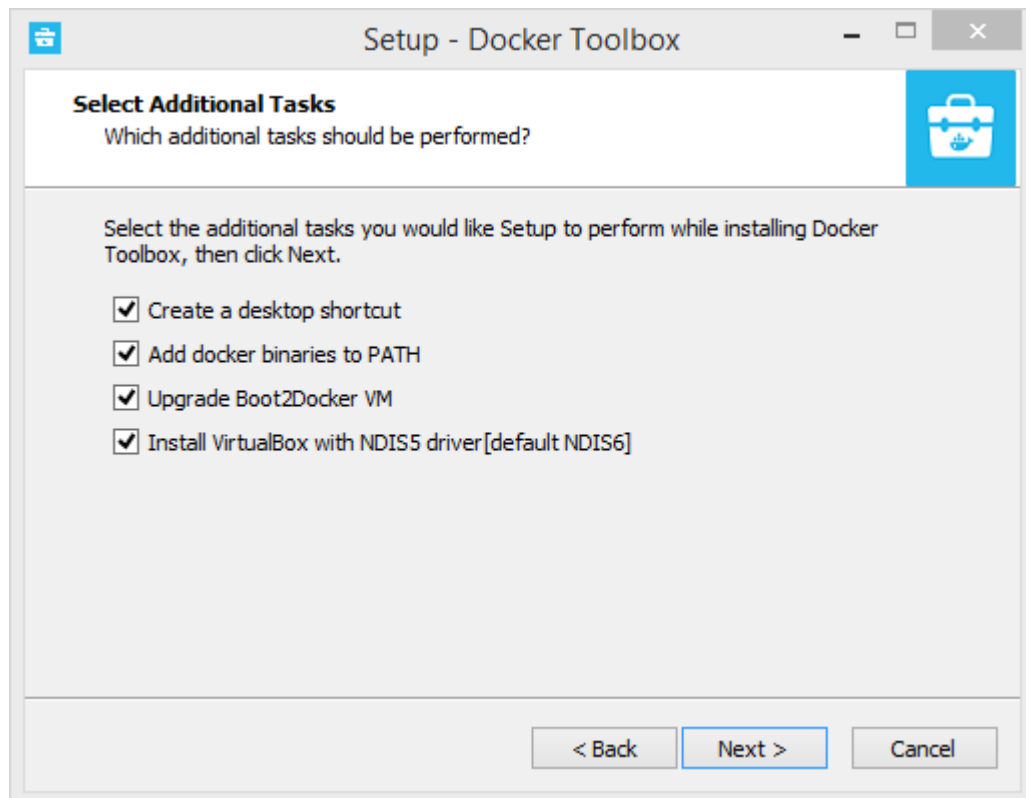




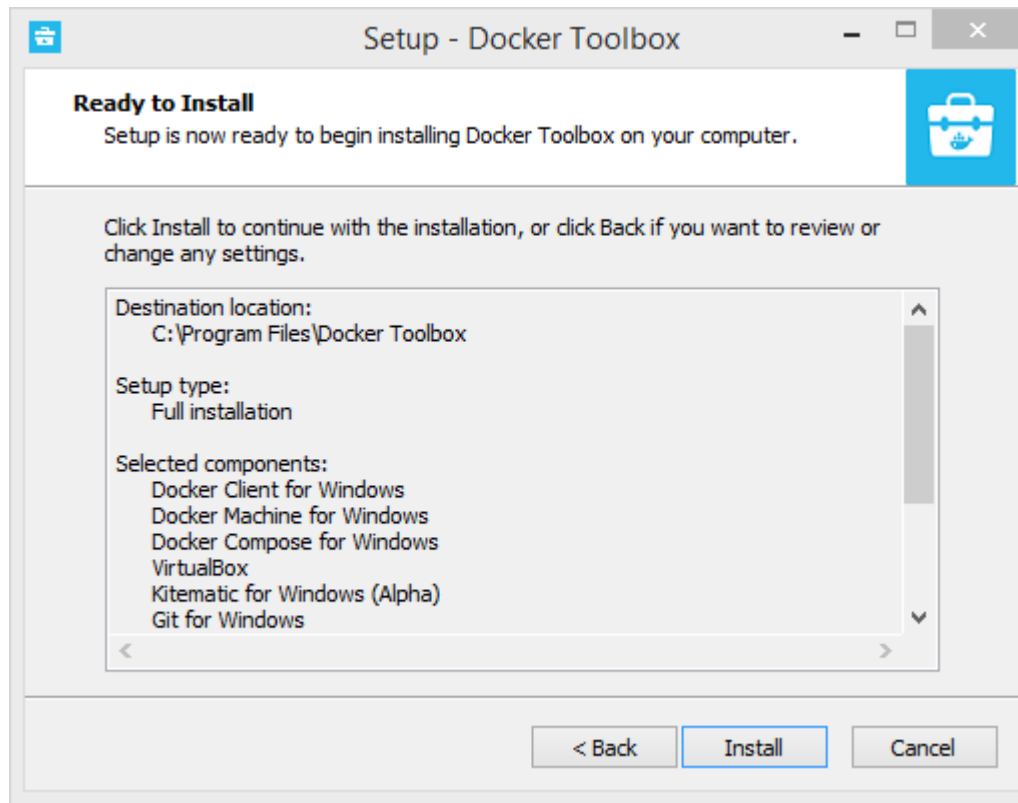
Pada step selanjutnya yaitu pengaturan direktori instalasi. Klik *Browse* untuk menentukan direktori instalasi. Jika tidak biarkan default direktori awal kemudian klik *Next*



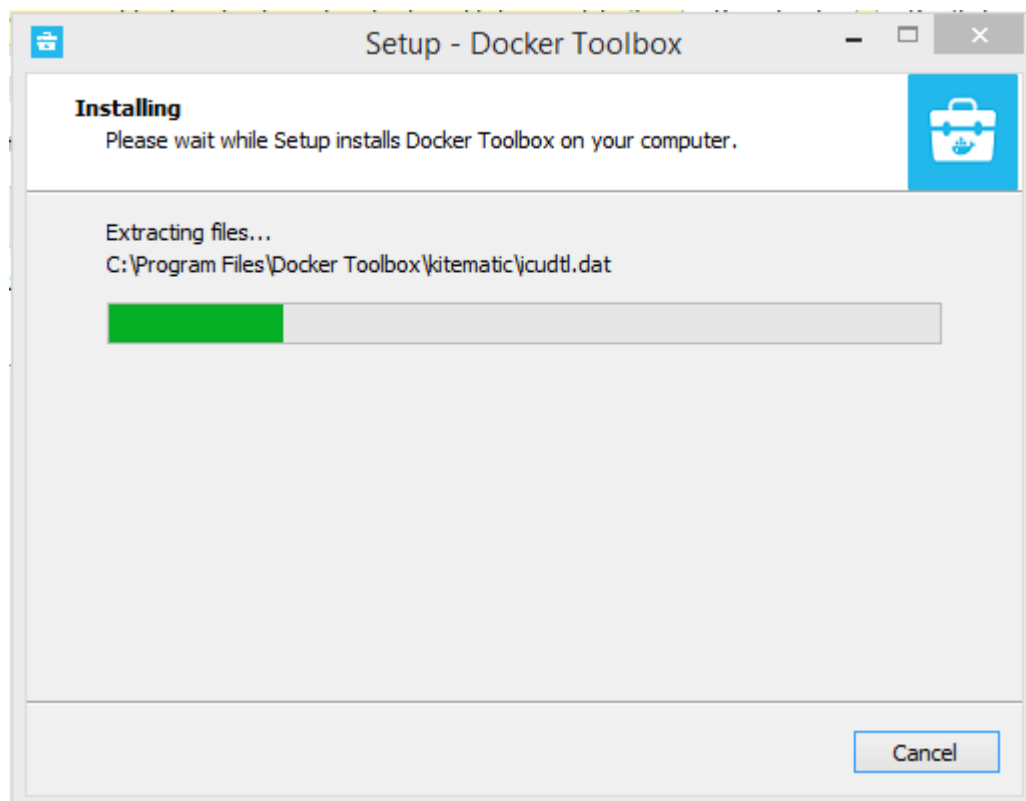
Pada step berikutnya memilih service apa saja yang di instal. Pada instalasi ini pilih *Full installation* kemudian klik *Next*.



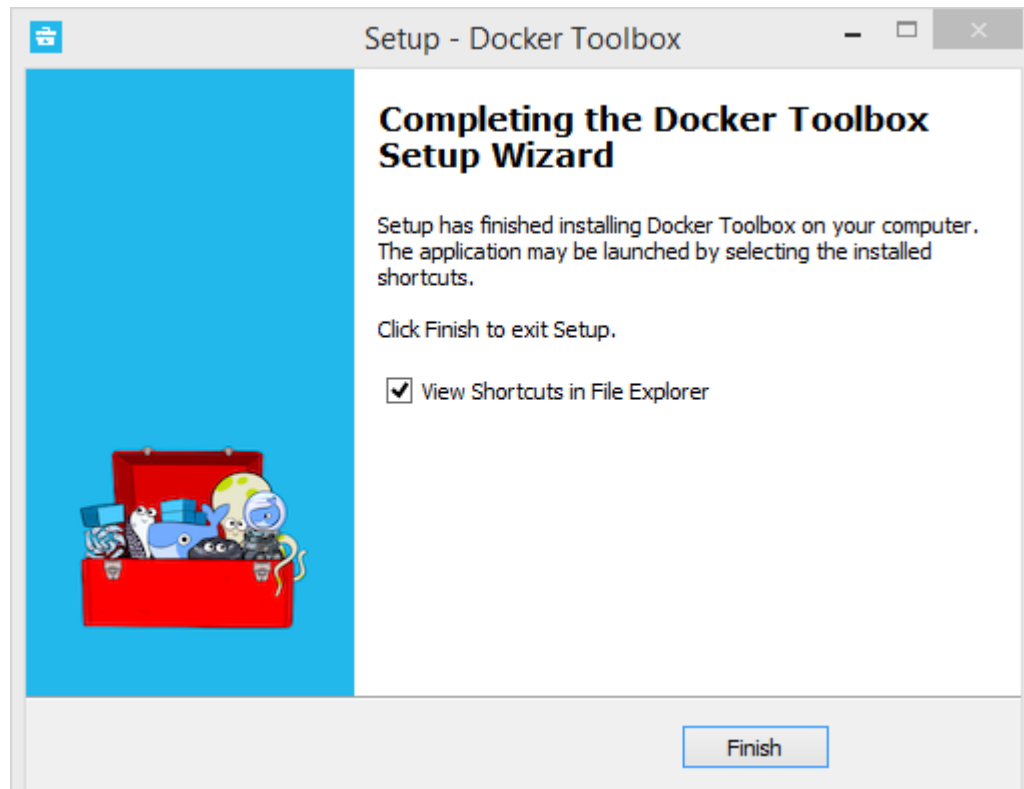
Pada step selanjutnya yaitu memilih additional task yang dibutuhkan. Biarkan default lalu klik *Next*



Klik *Install* untuk memulai instalasi docker toolbox



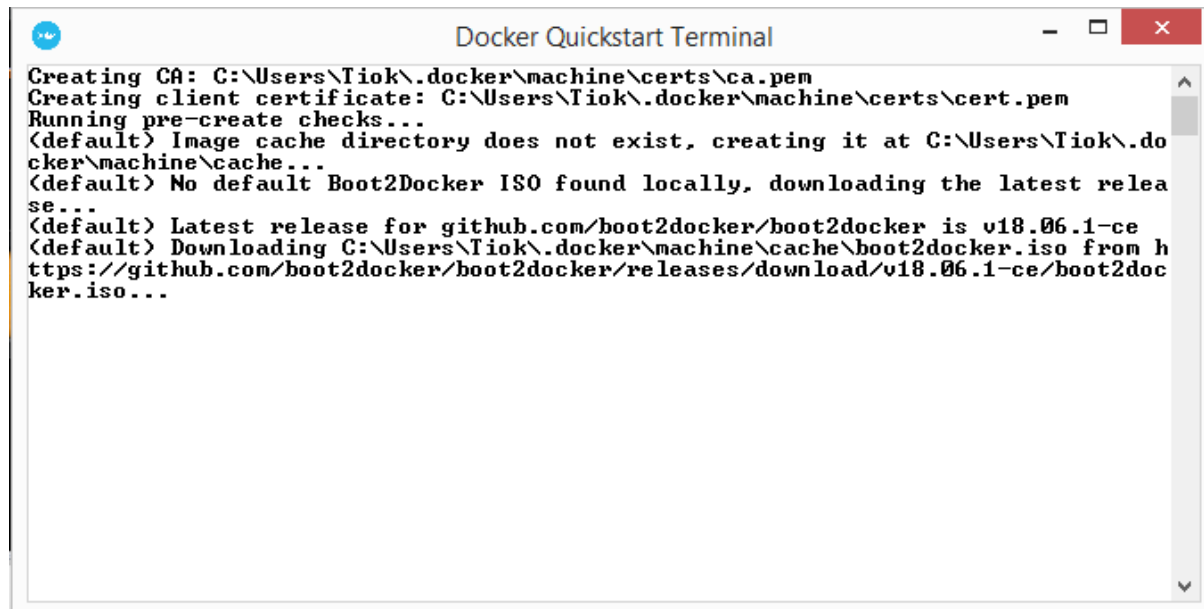
Tunggu proses instalasi docker toolbox hingga selesai seperti gambar dibawah ini



Klik *Finish*. Untuk menggunakan aplikasi docker toolbox klik icon *Docker Quickstart Terminal* seperti gambar dibawah ini

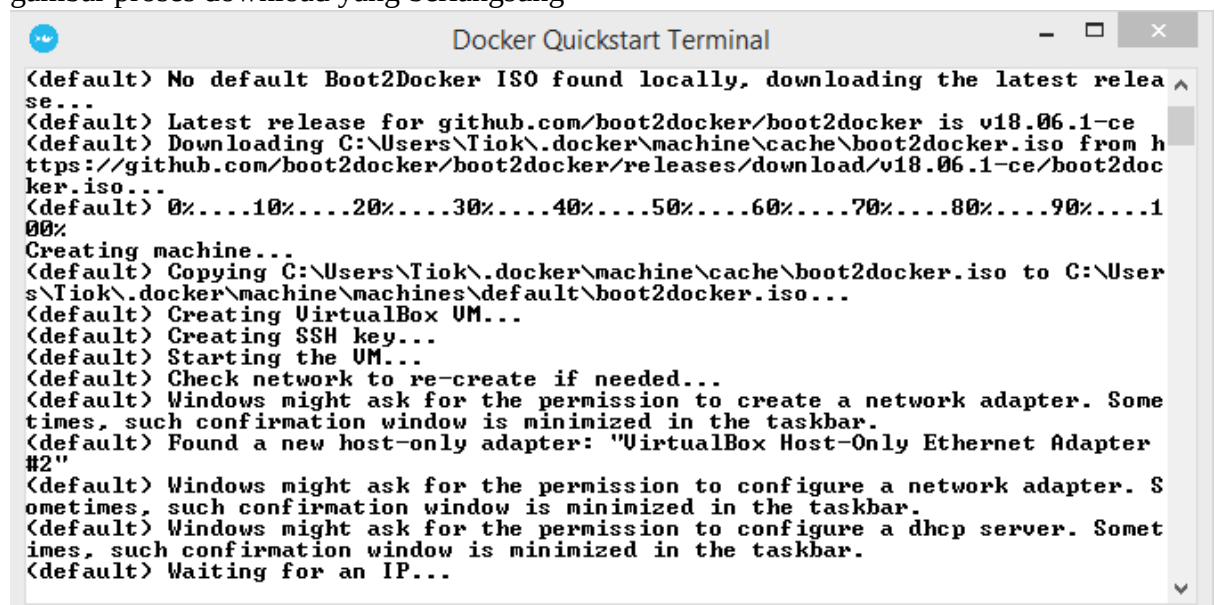


Setelah aplikasi Docker Quickstart Terminal dibuka akan muncul jendela terminal Docker Quickstart Terminal seperti gambar dibawah ini



```
Docker Quickstart Terminal
Creating CA: C:\Users\Tiok\.docker\machine\certs\ca.pem
Creating client certificate: C:\Users\Tiok\.docker\machine\certs\cert.pem
Running pre-create checks...
(default) Image cache directory does not exist, creating it at C:\Users\Tiok\.docker\machine\cache...
(default) No default Boot2Docker ISO found locally, downloading the latest release...
(default) Latest release for github.com/boot2docker/boot2docker is v18.06.1-ce
(default) Downloading C:\Users\Tiok\.docker\machine\cache\boot2docker.iso from https://github.com/boot2docker/boot2docker/releases/download/v18.06.1-ce/boot2docker.iso...
```

Pada step ini biarkan proses download berjalan hingga selesai. Berikut adalah beberapa gambar proses download yang berlangsung



```
Docker Quickstart Terminal
(default) No default Boot2Docker ISO found locally, downloading the latest release...
(default) Latest release for github.com/boot2docker/boot2docker is v18.06.1-ce
(default) Downloading C:\Users\Tiok\.docker\machine\cache\boot2docker.iso from https://github.com/boot2docker/boot2docker/releases/download/v18.06.1-ce/boot2docker.iso...
(default) 0%....10%....20%....30%....40%....50%....60%....70%....80%....90%....100%
Creating machine...
(default) Copying C:\Users\Tiok\.docker\machine\cache\boot2docker.iso to C:\Users\Tiok\.docker\machine\machines\default\boot2docker.iso...
(default) Creating VirtualBox VM...
(default) Creating SSH key...
(default) Starting the VM...
(default) Check network to re-create if needed...
(default) Windows might ask for the permission to create a network adapter. Sometimes, such confirmation window is minimized in the taskbar.
(default) Found a new host-only adapter: "VirtualBox Host-Only Ethernet Adapter #2"
(default) Windows might ask for the permission to configure a network adapter. Sometimes, such confirmation window is minimized in the taskbar.
(default) Windows might ask for the permission to configure a dhcp server. Sometimes, such confirmation window is minimized in the taskbar.
(default) Waiting for an IP...
```

Setelah download sudah selesai jendela terminal akan berubah seperti gambar dibawah ini







```
MINGW64:/c/Program Files/Docker Toolbox
Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
f17d81b4b692: Pull complete
d5c237920c39: Pull complete
a381f92f36de: Pull complete
Digest: sha256:b73f527d86e3461fd652f62cf47e7b375196063bbbd503e853af5be16597cb2e
Status: Downloaded newer image for nginx:latest
```

- Docker images. Docker images digunakan untuk melihat images yang ada. Contoh penggunaan seperti gambar dibawah ini

```
Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
nginx                latest             dbfc48660aeb       2 weeks ago
109MB
```

- “docker run -it image /bin/bash” digunakan untuk masuk kedalam sistem image. Contoh penggunaan seperti ini

```
Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker run -it nginx /bin/bash
root@6c619e4bbdaa:/#
```

Untuk test apakah kita masuk di sistem image bisa ketikkan perintah “cat /etc/os-release” seperti gambar dibawah ini

```
Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker run -it nginx /bin/bash
root@6c619e4bbdaa:/# cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 9 (stretch)"
NAME="Debian GNU/Linux"
VERSION_ID="9"
VERSION="9 (stretch)"
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
```

Ketikkan exit untuk keluar dari sistem image seperti gambar dibawah ini

```
root@a26a7adbcfa5:/# exit
exit
```

- “docker run -d -p 8080:80 -name nama-container image” perintah tersebut digunakan untuk membuat image menjadi container. Arti -p port yang digunakan untuk mengakses dan -name digunakan untuk memberika nama container yang akan dijalankan. Sebagai contoh seperti gambar dibawah ini

```
Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker run -d -p 8080:80 --name mynginx nginx
edd27bdc2e388d125196fa3a1de7134004385134a5f111ee4d44cdb21dd1e686
```

- “docker ps” perintah ini digunakan untuk melihat container yang sedang aktif atau berjalan. Sebagai contoh seperti gambar dibawah ini

```
Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
94643ab82369      nginx              "nginx -g 'daemon of" 11 seconds ago
Up 8 seconds      0.0.0.0:8080->80/tcp mynginx
```

- “docker ps -a” perintah ini digunakan untuk melihat semua container yang sedang aktif atau berjalan. Sebagai contoh seperti gambar dibawah ini

```
Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker ps -a
```

CONTAINER ID	IMAGE	PORTS	COMMAND	NAMES	CREATED
edd27bdc2e38	nginx	0.0.0.0:8080->80/tcp	"nginx -g 'daemon off'"	mynginx	4 minutes ago
a26a7adbcfa5	nginx		"/bin/bash"		8 minutes ago
6c619e4bbdaa	nginx		"/bin/bash"	modest_shirley	13 minutes ago
				mystifying_mclean	

- “docker exec -it nama-container /bin/bash”  
Setelah menjalankan perintah ini, anda akan berada dalam sistem operasi container. Dengan menggunakan command tersebut, dapat melakukan pengecekan file konfigurasi, data, dsb. Sebagai contoh seperti gambar dibawah ini

```
Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker exec -it mynginx /bin/bash
root@edd27bdc2e38:/#
```

Untuk melihat test apakah kita masuk pada sistem container jita berikan perintah “cat /etc/os-release” seperti gambar dibawah ini

```
Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker exec -it mynginx /bin/bash
root@edd27bdc2e38:/# cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 9 (stretch)"
NAME="Debian GNU/Linux"
VERSION_ID="9"
VERSION="9 (stretch)"
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
```

Ketikkan exit untuk keluar dari sistem image seperti gambar dibawah ini

```
root@a26a7adbcfa5:/# exit
exit
```

- “docker inspect nama-container/container-id”

Digunakan untuk melihat detail dari container yang berjalan seperti: IP Address dan lainnya Sebagai contoh seperti gambar di bawah ini

```
Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker inspect edd2
```

```
[
  {
    "Id": "edd27bdc2e388d125196fa3a1de7134004385134a5f111ee4d44cdb21dd1e686",
    "Created": "2018-11-05T08:48:37.158924775Z",
    "Path": "nginx",
    "Args": [
      "-g",
      "daemon off;"
    ],
    "State": {
```

- “docker stop id-container” perintah ini digunakan untuk menghentikan container yang berjalan. Id container yang dituliskan cukup 4 digit paling depan. Sebagai contoh ikuti langkah dibawah ini

```
Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker ps -l
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
edd27bdc2e38       nginx              "nginx -g 'daemon of" 26 minutes ago
Up 26 minutes      0.0.0.0:8080->80/tcp mynginx

Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker stop edd2
edd2

Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker ps -l
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
edd27bdc2e38       nginx              "nginx -g 'daemon of" 26 minutes ago
Exited (0) 5 seconds ago mynginx
```

- “docker start id-container” perintah ini digunakan untuk menjalankan container. Id container yang dituliskan cukup 4 digit paling depan. Sebagai contoh ikuti langkah dibawah ini.

```
Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker ps -l
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
edd27bdc2e38       nginx              "nginx -g 'daemon of" 26 minutes ago
Exited (0) 5 seconds ago mynginx

Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker start edd2
edd2

Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker ps -l
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
edd27bdc2e38       nginx              "nginx -g 'daemon of" 27 minutes ago
Up 4 seconds      0.0.0.0:8080->80/tcp mynginx
```

- “docker rm id-container/nama-container” perintah ini digunakan untuk menghapus container yang berjalan. Sebagai contoh ikuti langkah dibawah ini

```

Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker ps -l
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS             PORTS              NAMES
94643ab82369        nginx              "nginx -g 'daemon of" 7 minutes ago
Up 21 seconds       0.0.0.0:8080->80/tcp mynginx

Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker stop 9464
9464

Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker ps -l
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS             PORTS              NAMES
94643ab82369        nginx              "nginx -g 'daemon of" 8 minutes ago
Exited (0) 3 seconds ago          mynginx

Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker rm 9464
9464

Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox
$ docker ps -l
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS             PORTS              NAMES
a26a7adbcfa5        nginx              "/bin/bash"         About an hour ago
Exited (0) About an hour ago          modest_shirley

```

#### D. LATIHAN

Latihan diberikan oleh dosen pengampu

#### E. TUGAS

Tugas diberikan oleh dosen pengampu

## **DOCKER CONTAINER**

### **A. TUJUAN**

1. Mahasiswa mampu membuat image docker
2. Mahasiswa mampu menjalankan image sebagai container

### **B. TEORI SINGKAT**

Docker images adalah sebuah template yang bersifat read only. Template ini sebenarnya adalah sebuah OS atau OS yang telah diinstall berbagai aplikasi. Docker images berfungsi untuk membuat docker container, dengan hanya 1 docker images kita dapat membuat banyak docker container.

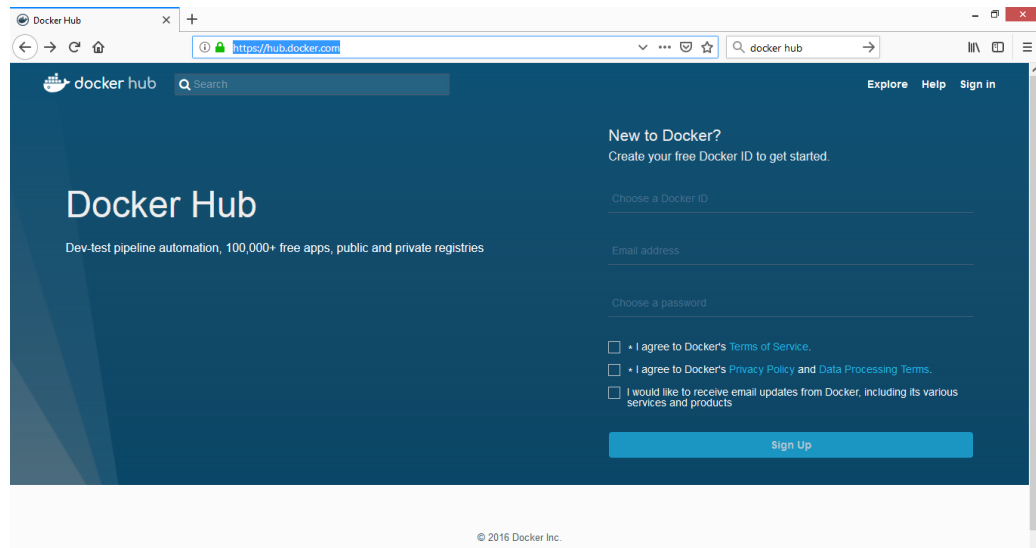
Docker container bisa dikatakan sebagai sebuah folder, dimana docker container ini dibuat dengan menggunakan docker daemon. Setiap docker container disimpan maka akan terbentuk layer baru tepat diatas docker images atau base image diatasnya. Contohnya misalkan kita menggunakan image ubuntu, kemudian kita membuat sebuah container dari image ubuntu tersebut dengan nama ubuntu, kemudian kita lakukan instalasi sebuah software misalnya nginx maka secara otomatis container ubuntu akan berada diatas layer image atau base image ubuntu. Anda dapat membuat banyak docker container dari 1 docker images. Docker container ini nantinya dapat dibuild sehingga akan menghasilkan sebuah docker images, dan docker images yang dihasilkan dari docker container ini dapat kita gunakan kembali untuk membuat docker container yang baru.

Docker registry adalah kumpulan docker image yang bersifat private maupun public yang dapat anda akses di [docker hub](https://hub.docker.com/). Dengan menggunakan docker registry, anda dapat menggunakan docker image yang telah dibuat oleh developer yang lain, sehingga mempermudah kita dalam pengembangan aplikasi.

### **C. PRAKTIK**

- **Membuat akun docker hub**

Buka halaman web <https://hub.docker.com/>



Buat akun Docker hub dengan mengisi Nama, Email dan Password seperti contoh dibawah ini

## New to Docker?

Create your free Docker ID to get started.

☐


\* I agree to Docker's [Terms of Service](#).


☐

\* I agree to Docker's [Privacy Policy](#) and [Data Processing Terms](#).

☐

I would like to receive email updates from Docker, including its various services and products

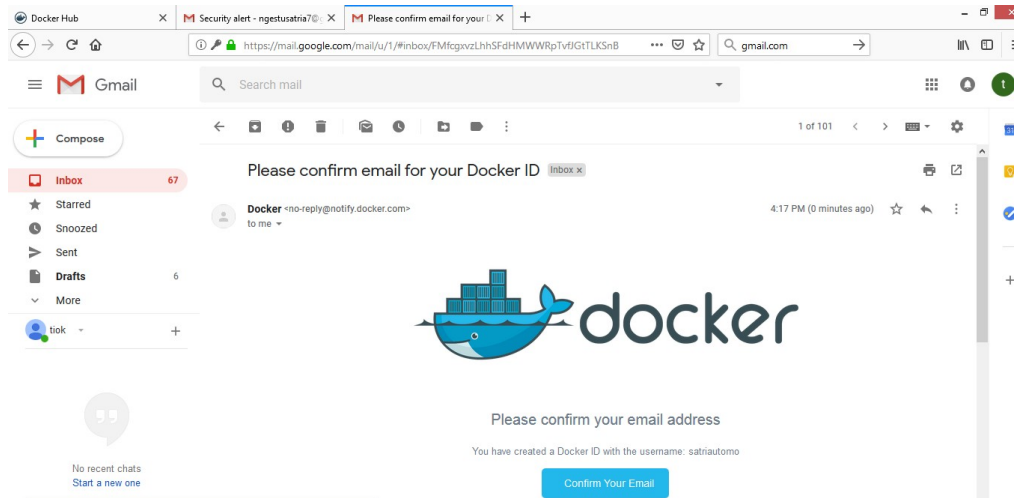
 I'm not a robot

  
reCAPTCHA  
[Privacy](#) - [Terms](#)

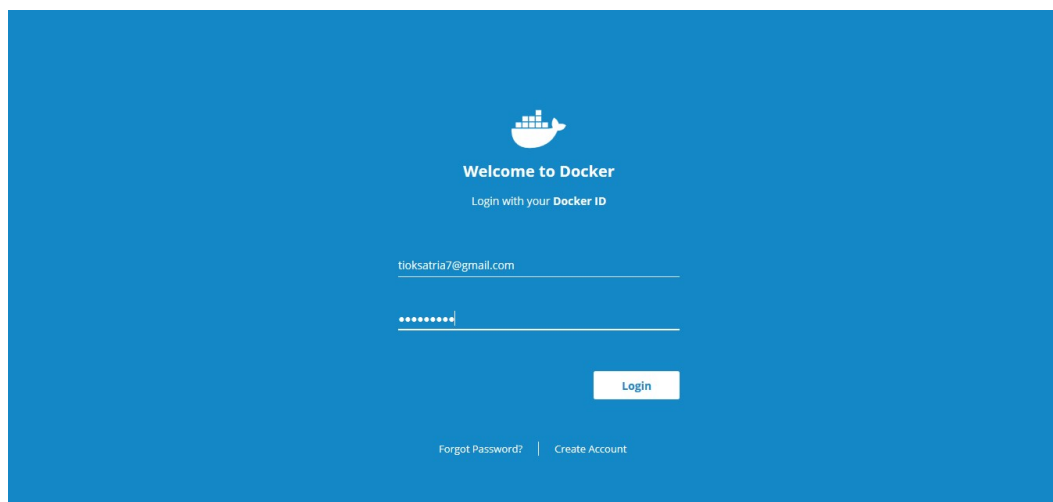
Sign Up



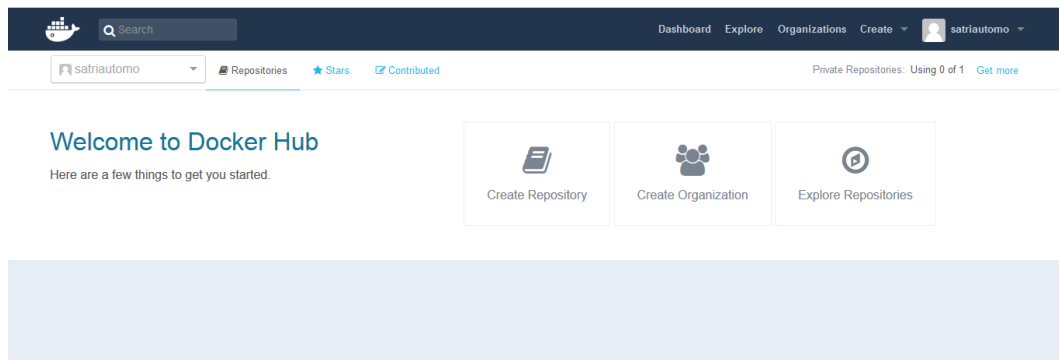
Verifikasi docker melalui email anda yang sudah di daftarkan pada docker hub



Setelah anda verifikasi silahkan anda login menggunakan nama email dan password yang sudah anda daftarkan

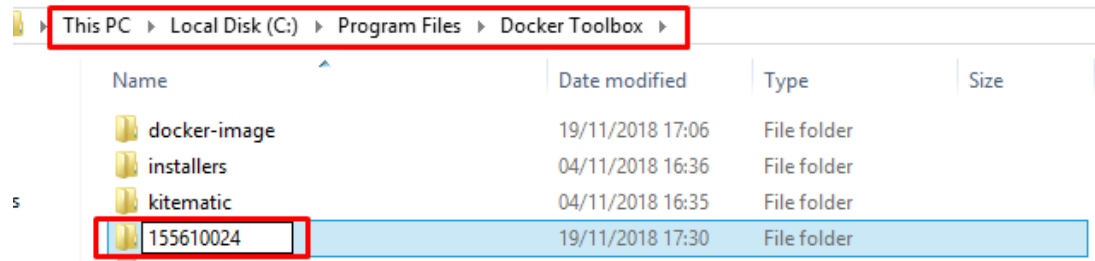


Maka berikut adalah tampilan home pada docker hub setelah anda login



- **Membuat Docker Container dengan docker file**

Buka aplikasi Docker Toolbox. Buat direktori dengan nama folder nim anda di dalam direktori C:\Program Files\Docker Toolbox. Contoh



Buat file dengan nama Dockerfile didalam direktori nim yang anda buat. Isikan kode berikut kedalam file Dockerfile

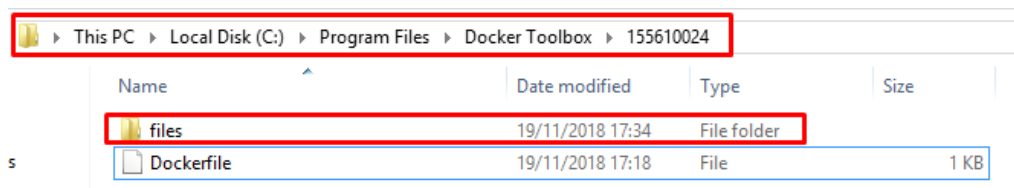
```

C:\Program Files\Docker Toolbox\155610024\Dockerfile - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
Dockerfile
1 FROM alpine:latest
2 RUN apk add --update nginx && \
3     rm -rf /var/cache/apk/* && \
4     mkdir -p /tmp/nginx/
5
6 COPY files/nginx.conf /etc/nginx/nginx.conf
7 COPY files/default.conf /etc/nginx/conf.d/default.conf
8 ADD files/index.html /usr/share/nginx/html/
9
10 EXPOSE 80/tcp
11
12 ENTRYPOINT ["nginx"]
13 CMD ["-g", "daemon off;"]
14

```

Bu

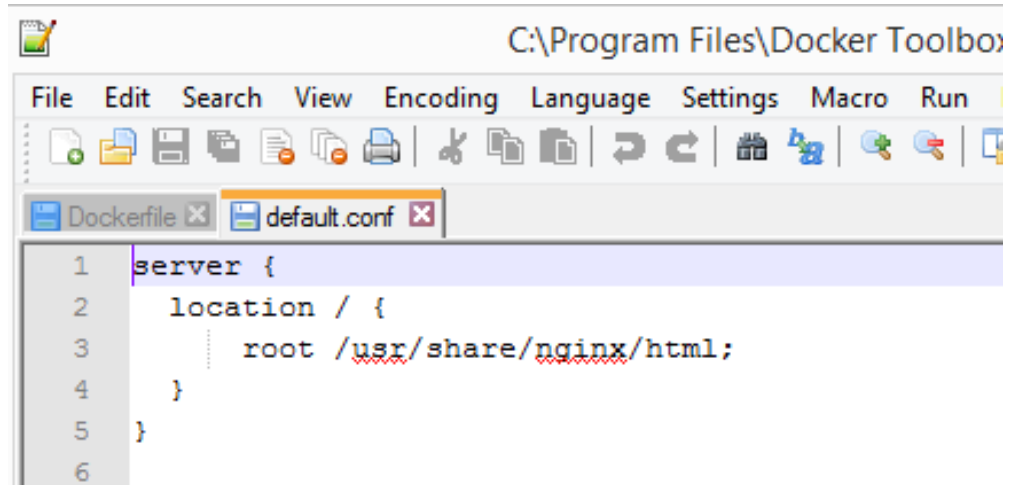
at direktori dengan nama files didalam direktori nim yang anda buat. Contoh



Bu

at file default.conf didalam direktori files. Isikan kode berikut kedalam file

default.conf

A screenshot of a text editor window titled "C:\Program Files\Docker Toolbox". The window has a menu bar with "File", "Edit", "Search", "View", "Encoding", "Language", "Settings", "Macro", and "Run". Below the menu bar is a toolbar with various icons. The editor shows two tabs: "Dockerfile" and "default.conf". The "default.conf" tab is active, displaying the following code:

```
1 server {  
2     location / {  
3         root /usr/share/nginx/html;  
4     }  
5 }  
6
```

Buat file index.html di dalam direktori files. Isikan kode untuk menampilkan nama dan nim anda ke dalam index.html

Buat file dengan nama nginx.conf di dalam direktori files. Isikan kode berikut kedalam file nginx.conf

S

etelah semua konfigurasi sudah dipersiapkan berikut struktur file yang ada di dalam direktori nim yang anda buat tadi

This PC > Local Disk (C:) > Program Files > Docker Toolbox > 155610024 >				
Name	Date modified	Type	Size	
files	19/11/2018 17:34	File folder		
Dockerfile	19/11/2018 17:18	File	1 KB	

This PC > Local Disk (C:) > Program Files > Docker Toolbox > 155610024 > files				
Name	Date modified	Type	Size	
default.conf	11/05/2018 8:15	CONF File	1 KB	
index	10/06/2017 9:54	HTML File	2 KB	
nginx.conf	11/05/2018 8:15	CONF File	1 KB	

Buka aplikasi Docker Toolbox kemudian masuk pada direktori nim yang anda buat sebelumnya seperti contoh pada gambar dibawah ini

Jalankan perintah “docker build -t nama-nimanda .” sepertigambar dibawah ini

Tunggu hingga proses download selesai

```
Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox/155610024
$ docker build -t satria-155610024 .
Sending build context to Docker daemon 7.168kB
Step 1/8 : FROM alpine:latest
--> 196d12cf6ab1
Step 2/8 : RUN apk add --update nginx && rm -rf /var/cache/apk/* &&
mkdir -p /tmp/nginx/
--> Running in e29f7eecb65e
fetch http://dl-cdn.alpinelinux.org/alpine/v3.8/main/x86_64/APKINDEX.tar.gz
fetch http://dl-cdn.alpinelinux.org/alpine/v3.8/community/x86_64/APKINDEX.tar.gz
(1/2) Installing pcre (8.42-r0)
(2/2) Installing nginx (1.14.1-r0)
Executing nginx-1.14.1-r0.pre-install
Executing busybox-1.28.4-r1.trigger
OK: 6 MiB in 15 packages
Removing intermediate container e29f7eecb65e
--> d83ad49ed2fb
Step 3/8 : COPY files/nginx.conf /etc/nginx/nginx.conf
--> ab85b11b7864
Step 4/8 : COPY files/default.conf /etc/nginx/conf.d/default.conf
--> cf15f6e9ab1b
Step 5/8 : ADD files/index.html /usr/share/nginx/html/
--> 148158fa8ce5
Step 6/8 : EXPOSE 80/tcp
--> Running in c71b33760f69
Removing intermediate container c71b33760f69
--> 4bc8293eba1c
Step 7/8 : ENTRYPOINT ["nginx"]
--> Running in 2118e4c9ecc9
Removing intermediate container 2118e4c9ecc9
--> 19653ff6ded5
Step 8/8 : CMD ["-g", "daemon off;"]
--> Running in 12ca12577883
Removing intermediate container 12ca12577883
--> 3702bb77741b
Successfully built 3702bb77741b
Successfully tagged satria-155610024:latest
SECURITY WARNING: You are building a Docker image from Windows against a non-Win-
dows Docker host. All files and directories added to build context will have '-r
wxr-xr-x' permissions. It is recommended to double check and reset permissions f
or sensitive files and directories.
```

Jalankan perintah “docker images” untuk melihat image yang ada

```
Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox/155610024
$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
satria-155610024    latest             3702bb77741b       About a minute ago
nginx               latest             dbfc40668aeb       4 weeks ago
alpine              latest             196d12cf6ab1       2 months ago
```

Ketika jalankan image tersebut menjadi container. Ketikkan perintah “docker run -d -p 8000:80 --name nim-nama namaimage” seperti contoh dibawah ini

```
Tiok@Satria MINGW64 /c/Program Files/Docker Toolbox/155610024
$ docker run -d -p 8000:80 --name 155610024-satria satria-155610024
35dachb76e409db54988647ff06c5ce9fda1ca7c58c3263288615bd238575d70e
```

Buka web browser anda ketikkan url <http://192.168.99.100:8000/>. Jika web yang tampil nama dan nim maka sukses di buat

- **Build image menggunakan github ke docker hub**  
Buat repositori di git hub anda dengan nama repositori tccminggu9

## Create a new repository

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

Owner

tioksatria

Repository name

tccminggu9

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

Description (optional)

☒ 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

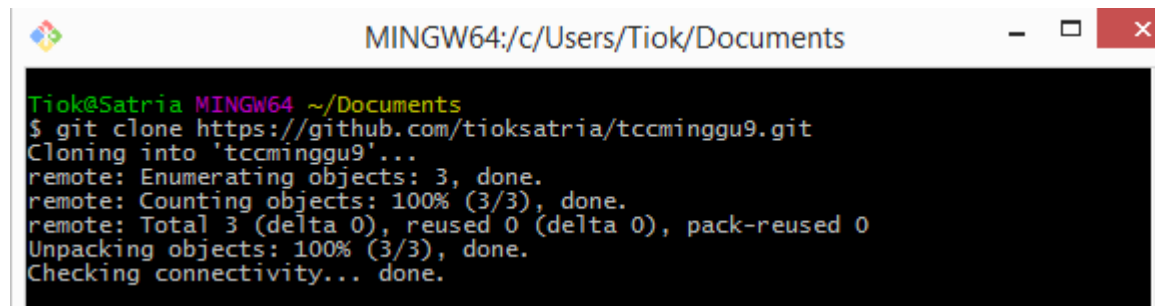
Buka Document kemudian klik kanan pilih Git Bash Here seperti gambar dibawah ini

This PC > Documents

Name	Date modified	Type	Size
CPY_SAVES	12/04/2018 17:27	File folder	
Criterion Games	22/04/2018 19:43	File folder	
Custom Office Templates	26/04/2018 8:25	File folder	
IISExpress	15/08/2018 15:11	File folder	
KONAMI	22/04/2018 9:56	File folder	
Lightshot	04/11/2018 16:08	File folder	
My ISO Files	12/04/2018 16:23	File folder	
My Shapes	12/04/2018 16:46	File folder	
My Web Sites	15/08/2018 15:11	File folder	
PCSX2	12/04/2018 17:42	File folder	
SQL Server Management Studio	16/08/2018 1:34	File folder	
Visual Studio 2015	27/08/2018 12:14	File folder	
Visual Studio 2017	24/08/2018 14:46	File folder	
ACTIVITY LOG	17/08/2018 0:30	File folder	17 KB

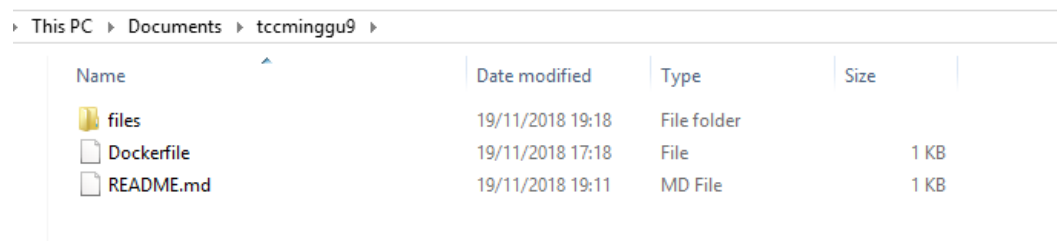
View  
Sort by  
Group by  
Refresh  
Customize this folder...  
Paste  
Paste shortcut  
Open in Visual Studio  
Git GUI Here  
Git Bash Here

Jalankan perintah “git clone alamat repositori anda” seperti contoh dibawah ini



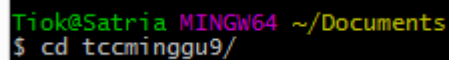
```
Tiok@Satria MINGW64 ~/Documents
$ git clone https://github.com/tioksatria/tccminggu9.git
Cloning into 'tccminggu9'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
Checking connectivity... done.
```

Setelah proses clone maka akan ada direktori tccminggu9 di Document. Copy dan paste semua file yang ada di direktori nim yang anda buat ke dalam direktori tccminggu9 seperti gambar dibawah ini



This PC > Documents > tccminggu9 >				
Name	Date modified	Type	Size	
files	19/11/2018 19:18	File folder		
Dockerfile	19/11/2018 17:18	File	1 KB	
README.md	19/11/2018 19:11	MD File	1 KB	

Kembali ke git bash kemudian masuk ke direktori tccminggu9



```
Tiok@Satria MINGW64 ~/Documents
$ cd tccminggu9/
```

Push ke repositori anda seperti gambar berikut

```
Tiok@Satria MINGW64 ~/Documents/tccminggu9 (master)
$ git init
Reinitialized existing Git repository in C:/Users/Tiok/Documents/tccminggu9/.git/

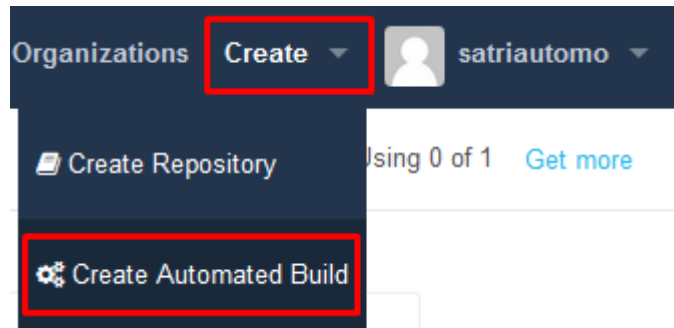
Tiok@Satria MINGW64 ~/Documents/tccminggu9 (master)
$ git add *
warning: LF will be replaced by CRLF in Dockerfile.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in files/default.conf.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in files/index.html.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in files/nginx.conf.
The file will have its original line endings in your working directory.

Tiok@Satria MINGW64 ~/Documents/tccminggu9 (master)
$ git commit -m "minggu 9"
[master 6d233c5] minggu 9
warning: LF will be replaced by CRLF in Dockerfile.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in files/default.conf.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in files/index.html.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in files/nginx.conf.
The file will have its original line endings in your working directory.
4 files changed, 71 insertions(+)
create mode 100644 Dockerfile
create mode 100644 files/default.conf
create mode 100644 files/index.html
create mode 100644 files/nginx.conf

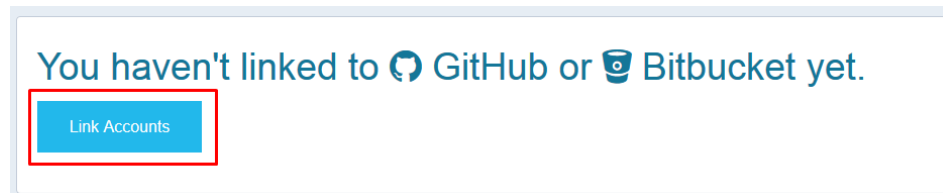
Tiok@Satria MINGW64 ~/Documents/tccminggu9 (master)
$ git push -u origin master
Fatal: AggregateException encountered.
Username for 'https://github.com': tioksatria
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 1.59 KiB | 0 bytes/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To https://github.com/tioksatria/tccminggu9.git
 25dffb1..6d233c5  master -> master
Branch master set up to track remote branch master from origin.
```

Cek ke repositori anda masing-masing apakah sudah benar-benar terupload. Jika sudah kita beralih ke docker hub dan silahkan login dengan akun docker hub anda masing-masing. Klik “create” kemudian pilih “Create Automated Build”

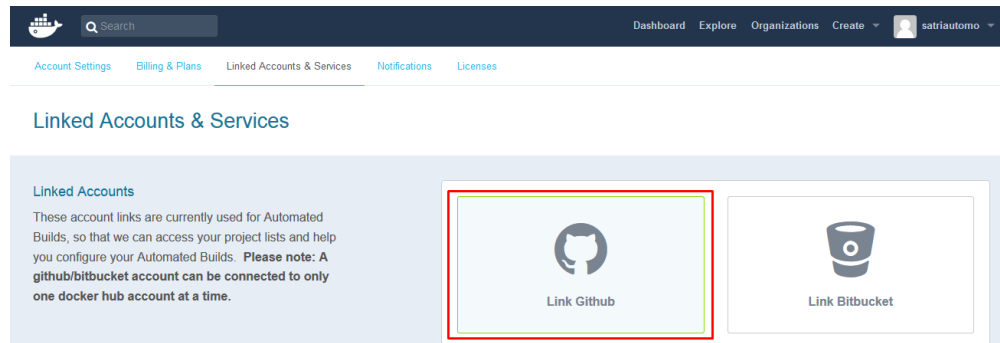




Klik “Link Accounts”



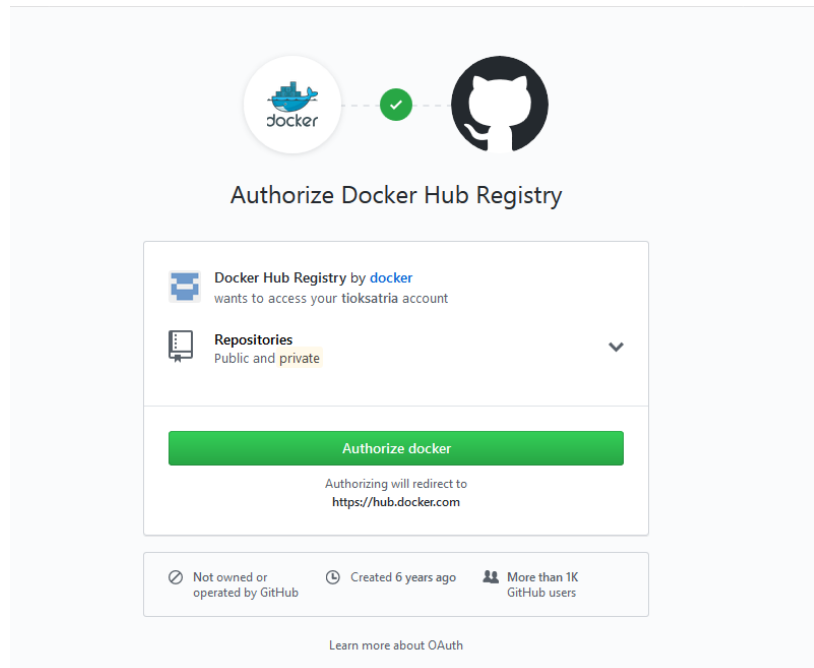
Pilih “Link Github”



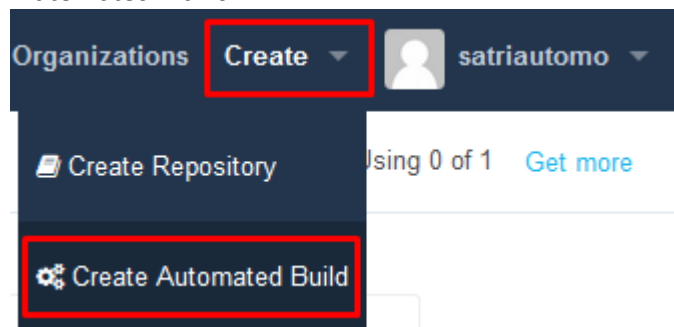
Sete

lah itu docker hub akan meminta Authorize ke akun github yang anda punya.

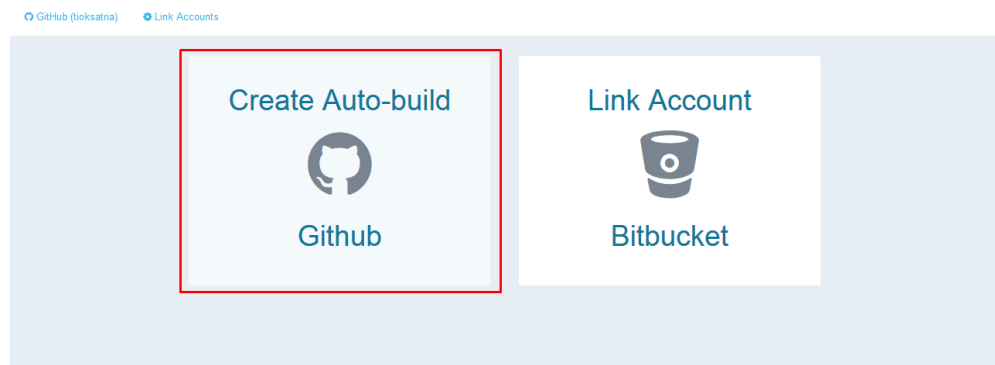
Klik pada “Authorize docker”



Selanjutnya kita kembali lagi pada menu “create” dan pilih pada “Create Automated Build”



Setelah itu pilih “Create Auto-build Github”



Kemudian akan muncul repositori yang ada di akun github anda. Pilih repositori yang tccminggu9 seperti contoh dibawah ini.

GitHub (tioksatria) Link Accounts

Users/Organizations

tioksatria

Type to filter

tccminggu9

Selanjutnya akan masuk pada tahap “Create Automated Build”. Pada visibility pilih “public”. kemudian pada Short Description silahkan diisi sesuai yang anda inginkan. Sementara pada modul ini di isi “minggu 9” seperti contoh dibawah

## Create Automated Build

Repository Namespace & Name\*

satriautomo tccminggu9

Visibility

public

Short Description\*

minggu 9

By default Automated Builds will match branch names to Docker build tags. [Click here to customize](#) behavior.

Create

Klik Create kemudian akan muncul tampilan seperti dibawah ini. Klik tab Build Settings

Repo Info Tags Dockerfile Build Details Build Settings Collaborators Webhooks Settings

Short Description

minggu 9

Docker Pull Command

docker pull satriautomo/tccminggu9

Full Description

Full description is empty for this repo.

Owner

satriautomo

Setelah kita berada pada Tab Build Settings kemudian klik pada tombol Trigger seperti contoh dibawah ini .

Repo Info Tags Dockerfile Build Details Build Settings Collaborators Webhooks Settings

### Build Settings

☒ When active, builds will happen automatically on pushes.

The build rules below specify how to build your source into Docker images. The name can be a string or a regex. The Docker Tag name may contain variables. We currently support {sourcerefs}, which refers to the source branch/tag name. [Show more](#)

Source Repository: tioksatria/tccminggu9

Type	Name	Dockerfile Location	Docker Tag Name		
Branch	master	/	latest	+	Trigger
Branch	All branches except master	/	Same as branch	-	

Save Changes

Kemudian klik Save Changes untuk menyimpan pengaturan seperti contoh dibawah ini

Repo Info Tags Dockerfile Build Details Build Settings Collaborators Webhooks Settings

### Build Settings

☒ When active, builds will happen automatically on pushes.

The build rules below specify how to build your source into Docker images. The name can be a string or a regex. The Docker Tag name may contain variables. We currently support {sourcerefs}, which refers to the source branch/tag name. [Show more](#)

Source Repository: tioksatria/tccminggu9

Type	Name	Dockerfile Location	Docker Tag Name		
Branch	master	/	latest	+	Triggered
Branch	All branches except master	/	Same as branch	-	

Save Changes

kemudian kita beralih pada tab Build Details seperti contoh dibawah ini.

Repo Info Tags Dockerfile Build Details Build Settings Collaborators Webhooks Settings

### Build Settings

☒ When active, builds will happen automatically on pushes.

The build rules below specify how to build your source into Docker images. The name can be a string or a regex. The Docker Tag name may contain variables. We currently support {sourcerefs}, which refers to the source branch/tag name. [Show more](#)

Source Repository: tioksatria/tccminggu9

Type	Name	Dockerfile Location	Docker Tag Name		
Branch	master	/	latest	+	Trigger
Branch	All branches except master	/	Same as branch	-	

Set

elah kita berada pada tab Build Details maka terdapat status build image dari github. Disini jika status berubah menjadi “Success” maka build image dari github sudah selesai

Repo Info	Tags	Dockerfile	Build Details	Build Settings	Collaborators	Webhooks	Settings
Status	Actions	Tag	Created	Last Updated	Source Repository		
Building		latest	2 minutes ago	a few seconds ago	tioksatria/tccminggu9		

Repo Info	Tags	Dockerfile	Build Details	Build Settings	Collaborators	Webhooks	Settings
Status	Actions	Tag	Created	Last Updated	Source Repository		
Success		latest	2 minutes ago	a few seconds ago	tioksatria/tccminggu9		

Selanjutnya bisa kita lihat dengan klik pada home docker hub maka akan ada image yang sudah di build dari github seperti contoh dibawah ini.

## Repositories

Create Repository +

Type to filter repositories by name			
satriaautom/tccminggu9 public   automated build	0 STARS	2 PULLS	DETAILS

Untuk mengetahui detail dari image yang anda buat klik “Detail” pada image. Pada bagian sebelah kanan terdapat alamat jika image ingin didownload seperti contoh dibawah ini

PUBLIC | AUTOMATED BUILD

satriaautom/tccminggu9 ☆

Last pushed: 6 minutes ago

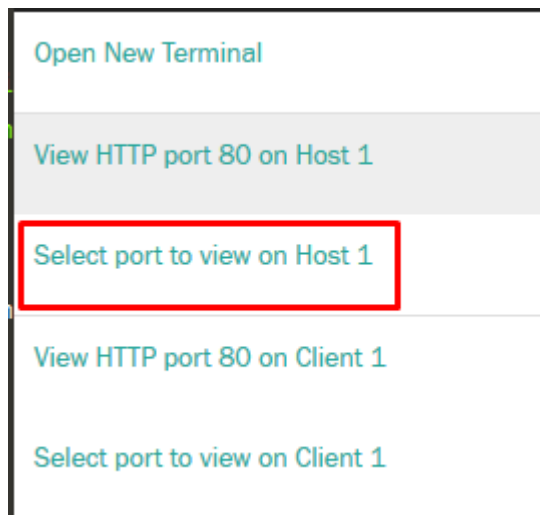
Repo Info	Tags	Dockerfile	Build Details	Build Settings	Collaborators	Webhooks	Settings
Short Description	Docker Pull Command			Owner			
minggu 9	docker pull satriaautom/tccminggu9			satriaautom			
Full Description							
tccminggu9							

Untuk test apakah image bisa dijalankan, kita menggunakan enviroment yang lainnya menggunakan “katacoda.com”. Jalankan perintah “docker pull idnama/tccminggu9” dan jalankan perintah “docker run -d -p 8000:80 --name nim-anda nama-image” seperti contoh dibawah ini

```
Terminal +
Your Interactive Bash Terminal.
A good starting point is executing `docker`

$ docker pull satriautomo/tccminggu9
Using default tag: latest
latest: Pulling from satriautomo/tccminggu9
4fe2ade4980c: Pull complete
566295c3aa68: Pull complete
d783c5330d24: Pull complete
1d1529015f4f: Pull complete
905b2e0d629c: Pull complete
Digest: sha256:809fde4112c44c4d2c0aeda3e514cf21e974c17ca55670752dc801d1b797cdc6
Status: Downloaded newer image for satriautomo/tccminggu9:latest
$ docker run -d -p 8000:80 --name 155610024-satria satriautomo/tccminggu9
bff237c48bfc095aa6e2305aea7ca878a7cb8beca3a356dcbb063869f355630
```

ik tanda “+” pada terminal kemudian pilih “Select port to view on Host 1” seperti contoh dibawah ini



Setelah itu ketikkan kode port “8000” kemudian klik “Display Port” seperti contoh dibawah ini.

---

## Web Preview Port Selector

To display a HTTP server, please enter the port number below

Display Port

Jika muncul nama dan nim anda masing-masing maka container berhasil dijalankan seperti contoh dibawah ini



Nama : Satria.

Nim : 155610024

#### **D. LATIHAN**

Latihan diberikan oleh dosen pengampu

#### **E. TUGAS**

Tugas diberikan oleh dosen pengampu

## DOCKER COMPOSE

### A. TUJUAN

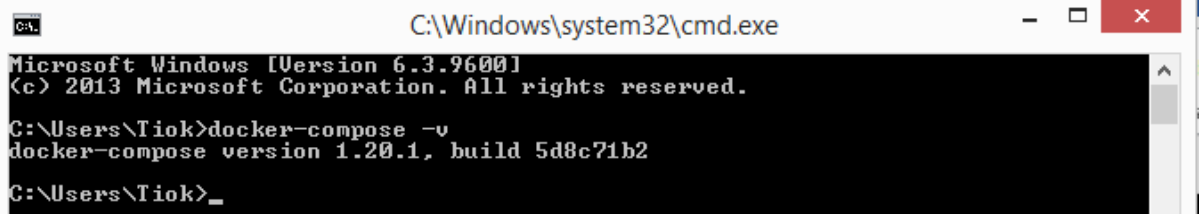
- Mahasiswa mengerti tentang kegunaan docker compose
- Mahasiswa dapat menggunakan docker compose sesuai dengan kebutuhan

### B. TEORI SINGKAT

Compose adalah salah satu tool untuk mendefinisikan dan menjalankan aplikasi Docker multi-container. Dengan Compose, untuk menjalankan menggunakan file Compose untuk mengonfigurasi layanan aplikasi. Kemudian, dengan menggunakan satu perintah, dapat membuat dan memulai semua layanan dari suatu konfigurasi.

### C. PRAKTIK

Sebelum memulai praktikum, kita perlu memastikan apakah docker compose sudah terinstall di komputer apa belum. Cara mengetahui yaitu dengan cara buka aplikasi command prompt dan ketikkan perintah “docker-compose -v” seperti contoh dibawah ini



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Tiok>docker-compose -v
docker-compose version 1.20.1, build 5d8c71b2

C:\Users\Tiok>_
```

Buatlah direktori dengan nama “docker-compose”. Dalam direktori “docker-compose” buat direktori dengan “product”. Pada direktori “product” buat dokumen dengan nama file “api.py”. Pada dokumen “api.py” ketikkan kode-kode seperti gambar dibawah ini



```
api.py x
1 # Product Service
2
3 # Import framework
4 from flask import Flask
5 from flask_restful import Resource, Api
6
7 # Instantiate the app
8 app = Flask(__name__)
9 api = Api(app)
10
11 class Product(Resource):
12     def get(self):
13         return {
14             'products': ['Ice cream', 'Chocolate', 'Fruit', 'Eggs']
15         }
16
17 # Create routes
18 api.add_resource(Product, '/')
19
20 # Run the application
21 if __name__ == '__main__':
22     app.run(host='0.0.0.0', port=80, debug=True)
23
```

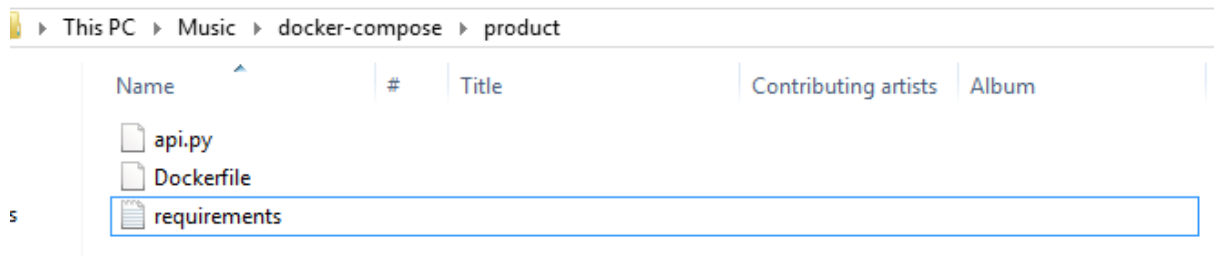
Setelah itu buatlah dokumen dengan nama file “Dockerfile” dan isikan kode-kode dibawah ini ke dalam “Dockerfile”.

```
api.py x Dockerfile x
1 FROM python:3-onbuild
2 COPY . /usr/src/app
3 CMD ["python", "api.py"]
4
```

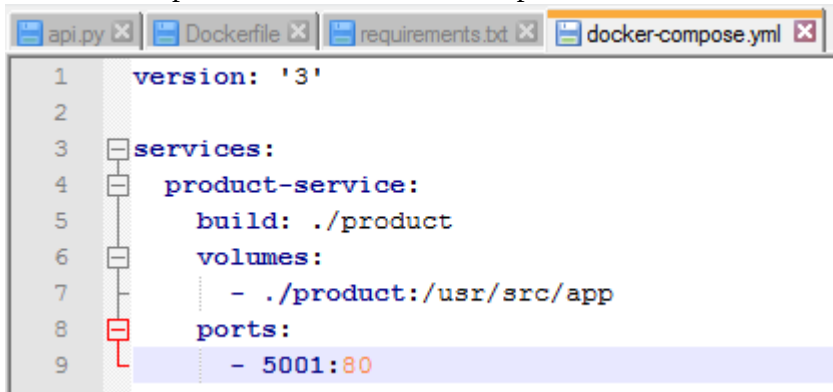
Setelah itu buatlah dokumen dengan nama file “requirements.txt” dan isikan kode-kode dibawah ini kedalam “requirements.txt”.

```
api.py x Dockerfile x requirements.txt x
1 Flask==0.12
2 flask-restful==0.3.5
3
```

Maka pada direktori “product” akan ada file-file seperti pada contoh dibawah ini

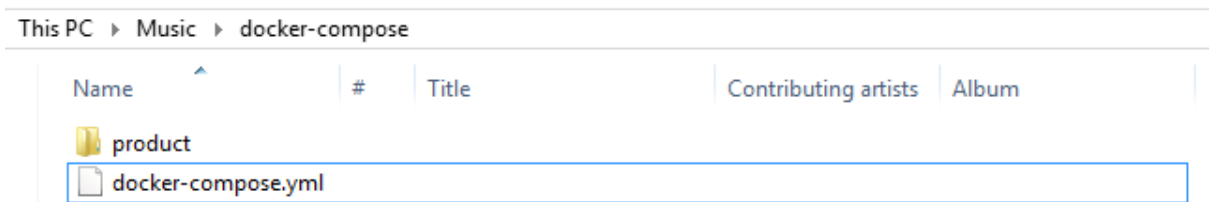


Setelah itu buatlah dokumen dengan nama file “docker-compose.yml” pada direktori “docker-compose”. Isikan kode-kode seperti dibawah ini

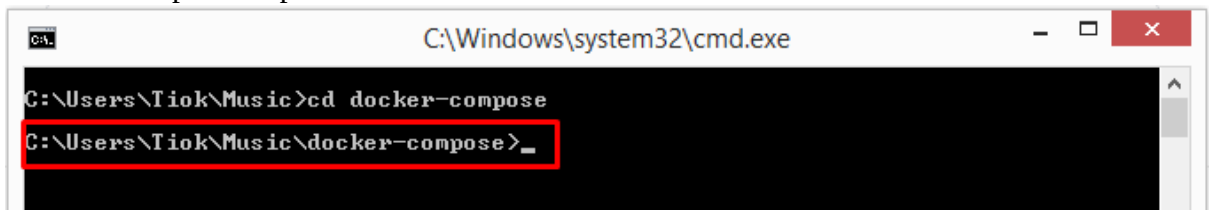


```
1 version: '3'
2
3 services:
4   product-service:
5     build: ./product
6     volumes:
7       - ./product:/usr/src/app
8     ports:
9       - 5001:80
```

Maka file yang ada di dalam direktori “docker-compose” seperti contoh dibawah ini



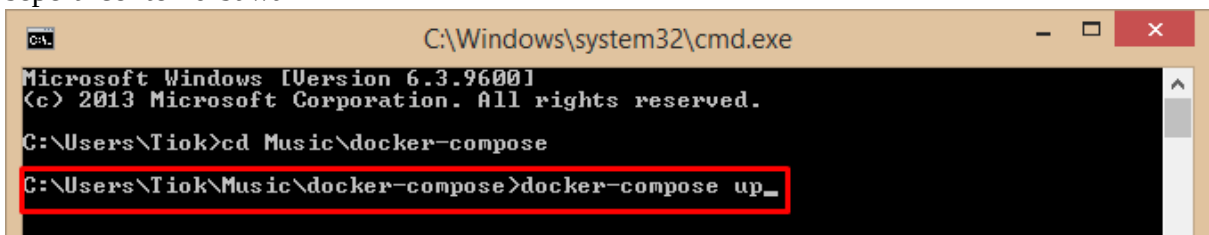
Buka kembali command prompt dan pastikan posisi direktori berada di dalam direktori “docker-compose” seperti contoh dibawah ini



```
C:\Windows\system32\cmd.exe

C:\Users\Tiok\Music>cd docker-compose
C:\Users\Tiok\Music\docker-compose>_
```

Kemudian ketikkan perintah “docker-compose up” untuk memulai docker compose seperti contoh dibawah ini



```
C:\Windows\system32\cmd.exe

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

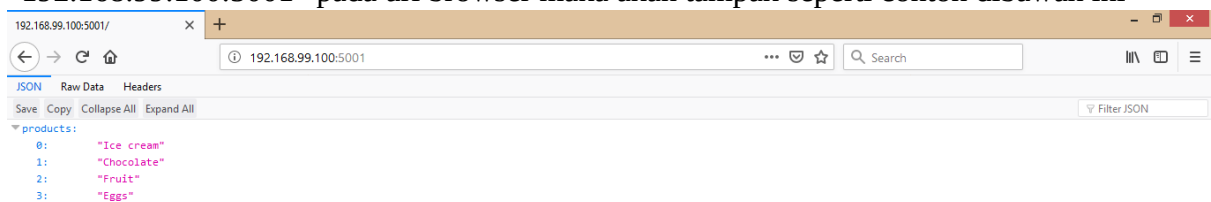
C:\Users\Tiok>cd Music\docker-compose
C:\Users\Tiok\Music\docker-compose>docker-compose up_
```

Selanjutnya yaitu proses download dan tunggu hingga selesai

```
C:\Windows\system32\cmd.exe

1c7fe136a31e: Downloading [==>]
3.218MB/54.25MBwloading [=====>]
ece825d3308b: Downloading [=====>]
ece825d3308b: Downloading [=====>]
4.5MB/17.59MB
ece825d3308b: Downloading [=====>]
ece825d3308b: Downloading [=====>]
1c7fe136a31e: Downloading [===>]
3.751MB/54.25MBwloading [=====>]
ece825d3308b: Downloading [=====>]
ece825d3308b: Downloading [=====>]
ece825d3308b: Downloading [=====>]
5.414MB/17.59MB
ece825d3308b: Downloading [=====>]
1c7fe136a31e: Downloading [===>]
4.291MB/54.25MB
ece825d3308b: Downloading [=====>]
5.954MB/17.59MB
ece825d3308b: Downloading [=====>]
ece825d3308b: Downloading [=====>]
1c7fe136a31e: Downloading [====>]
4.832MB/54.25MB
ece825d3308b: Downloading [=====>]
6.679MB/17.59MB
ece825d3308b: Downloading [=====>]
```

Jika sudah selesai kita cek dengan membuka browser dan ketikkan  
“192.168.99.100:5001” pada url browser maka akan tampak seperti contoh dibawah ini



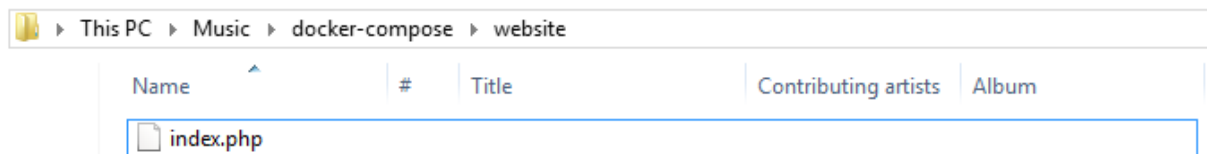
Pada langkah diatas yaitu menggunakan docker compose tetapi hanya membuat 1 image. Untuk selanjutnya yaitu kita coba menggunakan docker compose untuk membuat 2 image dan 2 image tersebut bisa saling terintegrasi. Sebelum memulai pastikan menghentikan docker compose yang sedang berjalan dengan cara mengetikkan perintah “docker-compose stop” seperti contoh dibawah ini

```
C:\Users\Tiok\Music\docker-compose>docker-compose stop
Stopping dockercompose_product-service_1 ... done
```

Masih menggunakan file-file yang terdapat pada direktori “docker-compose”. Tambahkan direktori “website” dan buat file dengan nama file “index.php”. Isikan kode-kode berikut pada file “index.php”.

```
api.py x Dockerfile x requirements.txt x docker-compose.yml x index.php x
1 <html>
2   <head>
3     <title>My Shop</title>
4   </head>
5
6   <body>
7     <h1>Welcome to my shop</h1>
8     <ul>
9       <?php
10
11         $json = file_get_contents('http://product-service/');
12         $obj = json_decode($json);
13
14         $products = $obj->products;
15
16         foreach ($products as $product) {
17           echo "<li>$product</li>";
18         }
19
20       ?>
21     </ul>
22   </body>
23 </html>
24
```

Maka pada direktori “website” terdapat file sebagai berikut



Setelah itu buka kembali file “docker-compose.yml” pada direktori “docker-compose”. Setelah itu tambahkan beberapa kode-kode seperti contoh dibawah ini

```
api.py x Dockerfile x requirements.txt x docker-compose.yml x
1 version: '3'
2
3 services:
4   product-service:
5     build: ./product
6     volumes:
7       - ./product:/usr/src/app
8     ports:
9       - 5001:80
10
11   website:
12     image: php:apache
13     volumes:
14       - ./website:/var/www/html
15     ports:
16       - 5000:80
17     depends_on:
18       - product-service
19
```

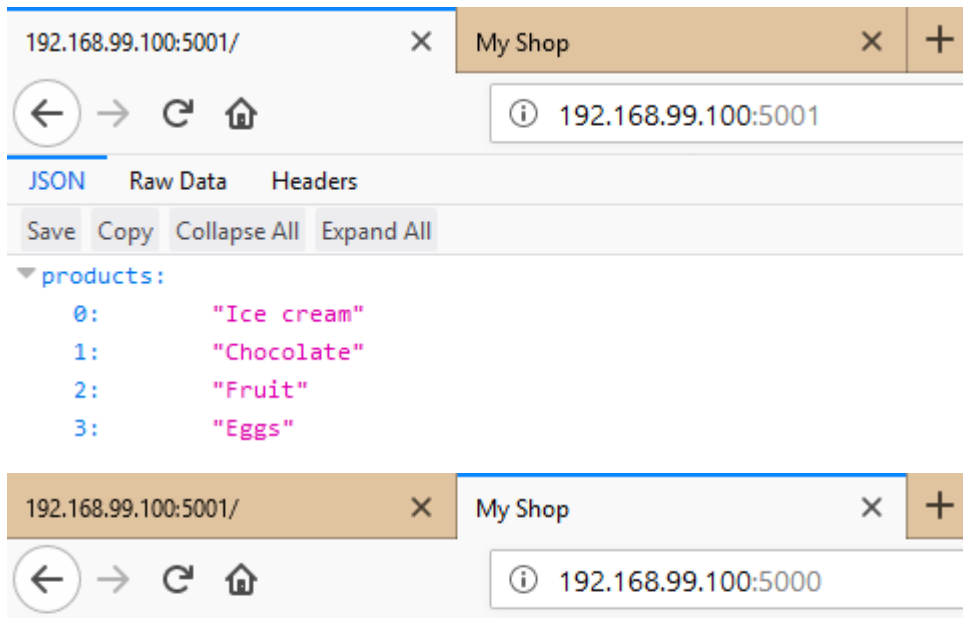
Kemudian buka kembali command prompt dan jalankan lagi “docker-compose up” seperti contoh dibawah ini

```
C:\Users\Tiok\Music\docker-compose>docker-compose up
Pulling website (php:apache)...
```

Maka selanjutnya proses download dan tunggu hingga selesai

```
C:\Windows\system32\cmd.exe - docker-compose up
Successfully tagged dockercompose_product-service:latest
WARNING: Image for service product-service was built because it did not already
exist. To rebuild this image you must use 'docker-compose build' or 'docker-comp
a5a6f2f73cd8: Downloading [=====>]
fcdfdf7118ba: Downloading [=====>]
16.06MB/67.43MBwloading [=====>]
a5a6f2f73cd8: Downloading [=====>]
14.68MB/22.49MBwloading [=====>]
a5a6f2f73cd8: Downloading [=====>]
14.9MB/22.49MBwloading [=====>]
a5a6f2f73cd8: Downloading [=====>]
c425447c8835: Downloading [=====>]
13.95MB/17.13MBwloading [=====>]
c425447c8835: Downloading [=====>]
14.13MB/17.13MBwloading [=====>]
a5a6f2f73cd8: Downloading [=====>]
14.45MB/22.49MBwloading [=====>]
14.47MB/67.43MBwloading [=====>]
c425447c8835: Downloading [=====>]
13.41MB/17.13MBwloading [=====>]
12.71MB/17.13MBiting
33ed51bc30e8: Waiting
7c4215700bc4: Waiting
ef55a760eb7a: Waiting
d982e3946ac5: Waiting
```

Setelah download selesai kita coba buka browser dan ketikkan “192.168.99.100:5001” dan “192.168.99.100:5000”. Maka jika berhasil kedua alamat tersebut akan tampak seperti contoh dibawah ini



## Welcome to my shop

- Ice cream
- Chocolate
- Fruit
- Eggs

Dari ilustrasi diatas menunjukkan container php mengambil data dari container python. Disini fungsi docker compose sebenarnya yaitu untuk membuat dan mengintegrasikan beberapa container agar bisa saling terhubung.

### D. LATIHAN

Latihan diberikan oleh dosen pengampu

### E. TUGAS

Tugas diberikan oleh dosen pengampu