**Dockerfile**

1. **Pengenalan Docker File**

Pada materi sebelum membahas tentang Docker dasar, sekarang bagaimana jika ingin membuat Docker Image sendiri?Pembuatan Docker Image bisa dilakukan dengan menggunakan instruksi-intruksi yang kita simpan didalam file DockerFile(Membuat Image).

**Docker File**

Jadi Sederhananya Dockerfile adalah file text yang berisi semua perintah yang bisa kita gunakan untuk membuat sebuah Docker Image. Anggap saja semua instruksi sama seperti Bahasa pemrograman namun tidak kompleks, instruksi tersebut untuk menjalankan aplikasi kita, CRUD container, dll selanjutnya kita simpan didalam Dockerfile nanti Dockerfile tersebut akan dieksekusi sebagai perintah untuk membuat Docker Image.

**Docker Build**

Untuk membuat Docker Image dari Dockerfile, kita bisa menggunakan perintah docker build. Saat membuat Docker Image dengan docker build, nama image secara otomatis akan dibuat random dan biasanya kita ingin menambahkan nama/tag pada imagenya, kita bisa mengubahnya dengan perintah -t.

Misal berikut adalah contoh cara menggunakan docker build :

**docker build -t namaimage/tagversion folderDockerfile**

docker build -t khannedy/app:1.0.0 folder-dockerfile

docker build -t anommudita/app:1.0.0 -t anommudita/app:latest folder-dockerfile (membuat satu image dengan 2 nama berbeda)

1. **Dockerfile File Format**

Seperti Namanya, Dockerfile biasanya dibuat dalam sebuah file dengan nama Dockerfile, tidak memiliki extension apapun, walaupun sebenarnya bisa saja kita membuat dengan nama lain, namun direkomendasikan menggunkan nama Dockerfile.

1. **Instuction Format**

Instruksi untuk membuat Dockerfile, sama halnya seperti Bahasa pemrograman tanda #(untuk komentar) dan huruf besarkecil bisa sensitif, secara sederhana berikut adalah format untuk file Dockerfile :

1. # Komentar
2. INSTRUCTION arguments(paramternya)

* **# 🡺** digunakan menambahkan komentar, kode dalam baris tersebut secara otomatis dianggap komentar atau disable
* **INSTRUCTION** adalah untuk peritah yang digunakan di Dockerfile, ada banyak perintah yang tersedia, dan penulisan perintahnya case sensitive, sehingga kita bisa gunakan huruf besar atau kecil, Namun rekomendasi menggunakan UPPER CASE
* **Arguments** adalah data argument untuk INSTRUCTION, yang menyesuaikan dengan jenis INSCTRUCTION yang digunakan.

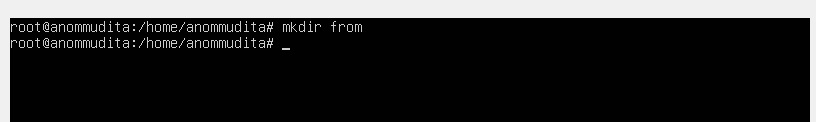
1. **From Instruction**

Pada saat membuat Docker Image, biasanya perintah pertama adalah melakukan build stage dengan instruksi FROM. FROM digunakan untuk membuat build stage dari image yang kita tentukan, biasanya jarang sekali kita akan membuat Docker Image dari scratch(kosongan), biasanya kita akan membuat Docker Image dari Docker Image lain yang sudah ada, misalnya kita ingin membuat image JAVA berarti kita menggunakan image asal dari JAVA yang sudah ada. Untuk menggunakan FROM, kita bisa gunakan perintah **“FROM image:version”**

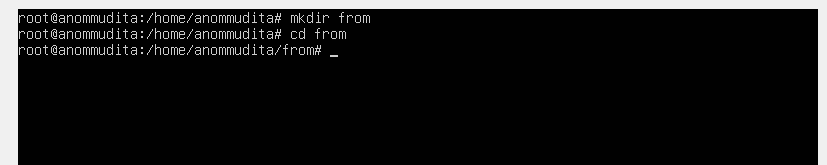
Dicontoh saya mengunakan alpine karena alpine sebuah OS linux yang paling kecil ukurannya.

**Tahap Membuat Image**

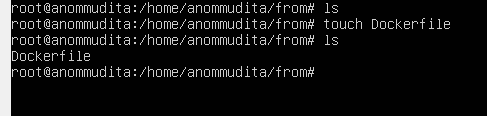
1. Buatlah directory bernama from, untuk pemberian nama directory bebas saja dengan perintah “**mkdir from**” jika ingin menghapusnya gunakan perintah “**rm -rf namedirectory**”



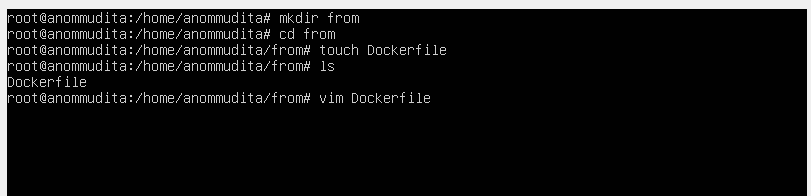
1. Masuk ke directory yang kalian sudah buat sebelumnya

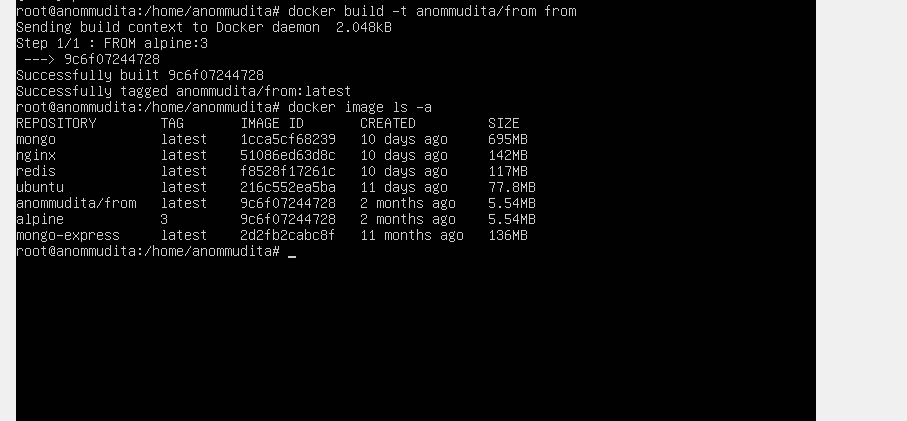


1. Jika sudah masuk ke directory maka buat file Dockerfilenya dengan perintah **“touch Dockerfile”** tanpa format



1. Setelah sudah membuatnya, masuk ke Dockerfile dengan perintah “**vim Dockerfile**”



1. Masuk ke file Dockerfile dan berikan perintah instruksinya “**FROM alpine:3**”, setelah sudah maka kalian keluar dari texteditor tersebut dengan **Esc + : , lalu wq! + enter**
2. Setelah sudah membuat Dockerfile, sekarang saatnya menggunakan perintah Docker Build sehingga image yang kita buat akan dibangun oleh docker “**docker build -t anommudita/from from**” dan lihat image yang kita buat apakah berhasil dibuat. Catatan jika ingin build image, maka kalian harus berada diluar folder Dockerfile yang kalian buat. Gunakan perintah **“cd ..”**
3. **Run Instruction**

Run adalah sebuah instruksi untuk mengeksekusi perintah didalam image pada saat build stage(tahap membangun). Hasil perintah RUN akan di commit dalam perubahan image tersebut, jadi perintah RUN akan dieksekusi pada saat proses docker build saja, setelah menjadi Docker Image, perintah tersebut tidak akan dijalankan lagi. Jadi ketika menjalankan Docker Container dari Image tersebut, maka perintah RUN tidak akan dijalankan lagi.

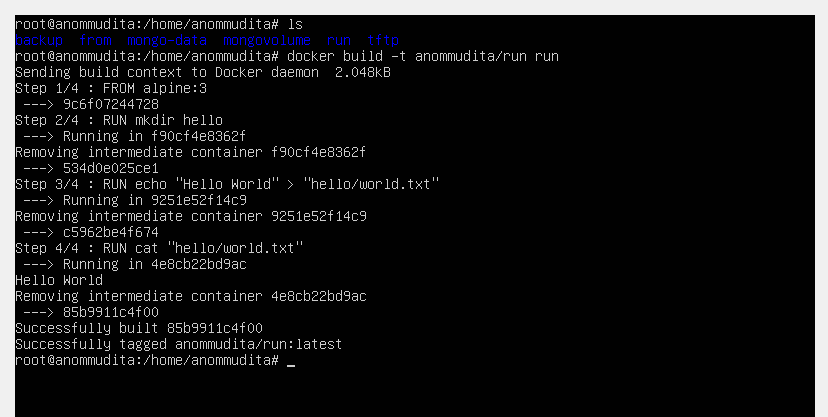
**Run Instruction Format**

Perintah RUN memiliki 2 format :

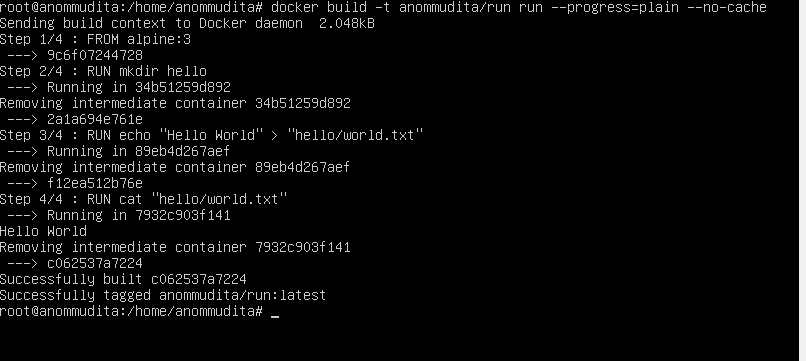
* RUN Command
* RUN [“executable”, “argument”, “…”]

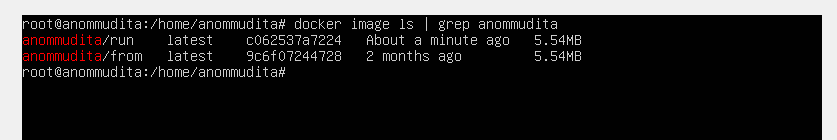
**Tahapan RUN Instruction**

1. Buat directory berikan nama bebas, lalu buat file dengan nama Dockerfile tanpa format dan buka file tersebut, ikuti perintah dibawah.
2. Setelah membuat Dockerfile, ikuti perintah dibawah. Instruksi yang saya buat ialah membuat folder, membuat file yang berisikan output Hello World dan menjalankan file world.txt, untuk keluar dari text editor tersebut dengan **Esc + : , lalu wq! + enter**
3. Kemudian jalankan perintah docker build untuk membuat Image dan hasil output yang kita buat di Dockerfile terlihat diterminal. Catatan jika ingin build image, maka kalian harus berada diluar folder Dockerfile yang kalian buat. Gunakan perintah **“cd ..”**



1. **Display Output**

****Secara default, di docker terbaru tidak akan menampilkan tulisan detail dari build-nya, jika kita ingin menampilkan detailnya, kita bisa gunakan perintah **--progress=plain**. Selain itu juga docker build melakukan cache, jika kita ingin mengulangi lagi tanpa menggunakan cache, kita bisa gunaka perintah **–no-cache**

****Untuk melihat Image Docker yang kita buat tadi gunakan perintah **“docker image ls | grep anommudita”**

1. **Command Instruction**

CMD atau Command merupakan instruksi yang digunakan ketika Docker Container berjalan. CMD tidak akan berjalan ketika proses build, namun dijalankan ketika Docker Container berjalan. Dalam Dockerfile, kita tiba bisa menambahkan lebih dari satu instruksi CMD, jika kita tambahkan lebih dari satu instruksi CMD, maka yang akan digunakan untuk menjalankan Docker Container adalah instruksi CMD yang terakhir.

**Command Instruction Format**

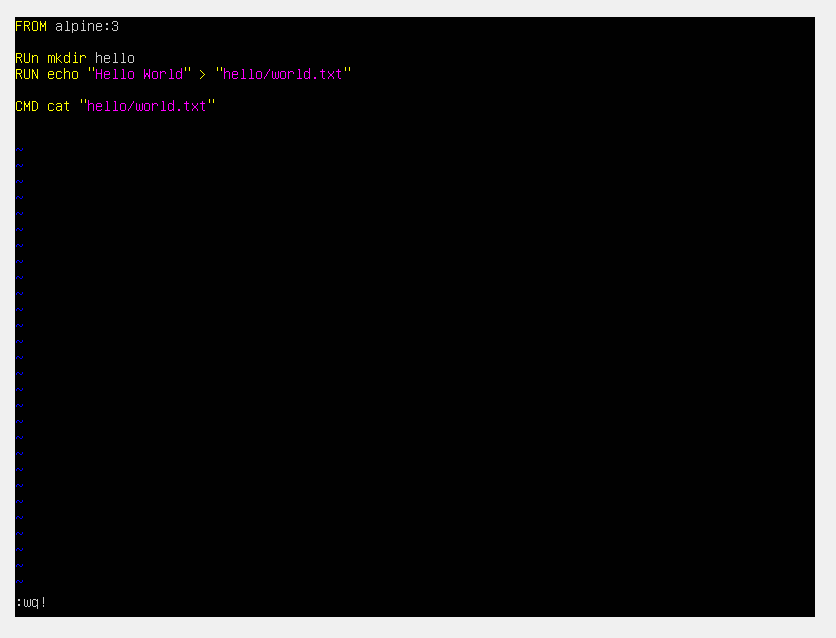
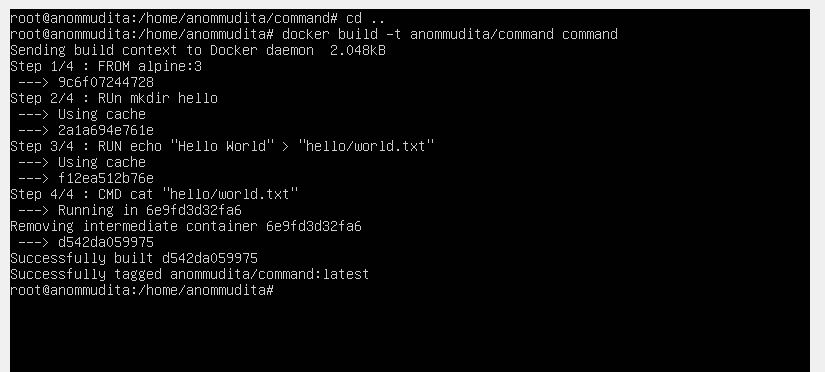
Perintah CMD memiliki beberapa format :

1. CMD command param param
2. CMD [“executable”, “param”, “param”]
3. CMD [“param”, “param”], akan menggunakan executable ENTRY POINT

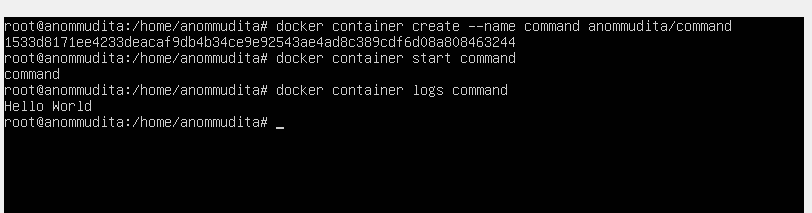
**Tahapan Command Instruction**

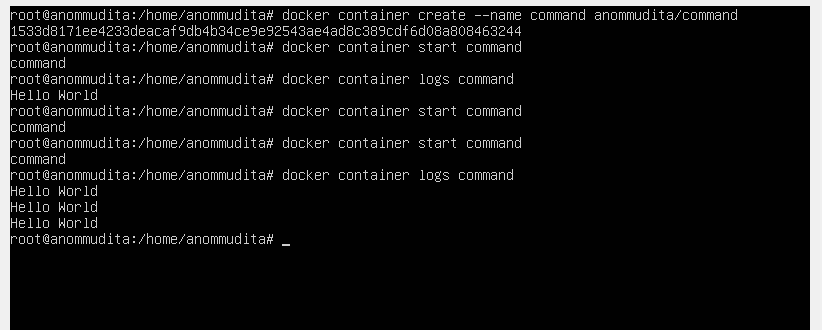
1. Buat directory berikan nama bebas, lalu buat file dengan nama Dockerfile tanpa format dan buka file tersebut, ikuti perintah dibawah.



1. Setelah membuat Dockerfile, ikuti perintah dibawah. Instruksi yang saya buat ialah membuat folder, membuat file yang berisikan output Hello World dan menjalankan file world.txt jika container sudah start atau dijalankan, untuk keluar dari text editor tersebut dengan **Esc + : , lalu wq! + enter**
2. Kemudian jalankan perintah docker build untuk membuat Image dan hasil output yang kita buat di Dockerfile tidak terlihat diterminal, karena instruksi CMD hanya berjalan jika container sudah dijalankan, untuk melihatnya kalian bisa menggunakan fitur docker inspect. Catatan jika ingin build image, maka kalian harus berada diluar folder Dockerfile yang kalian buat. Gunakan perintah **“cd ..”**

Menggunakan Fitur Docker Inspect untuk melihat outputnya

1. Membuat Docker Container untuk menjalankan hasil outputnya, lalu lihat logs pada Container tersebut dan terbukti bahwa tiap kali Container dijalankan akan mengeksekusi file world.txt atau menampilkan output ke terminal

Instuksi CMD ini, relevan sekali untuk pembuatan website

1. **Label Instruction**

Instruksi LABEL merupakan instruksi yang digunakan untuk menambahkan metadata ke dalam Docker Image yang kita buat. Metadata adalah informasi tambahan, misal seperti nama aplikasi, pembuat, website, perusahaan, lisensi dan lain-lain. Metadata hanya berguna sebagai informasi saja, tidak akan digunakan ketika kita menjalankan Docker Container.

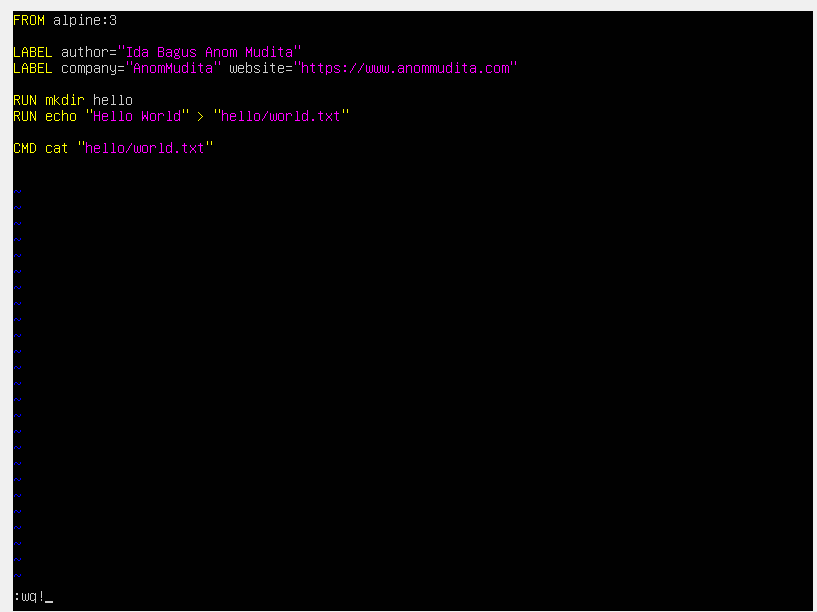
**Label Instruction Format**

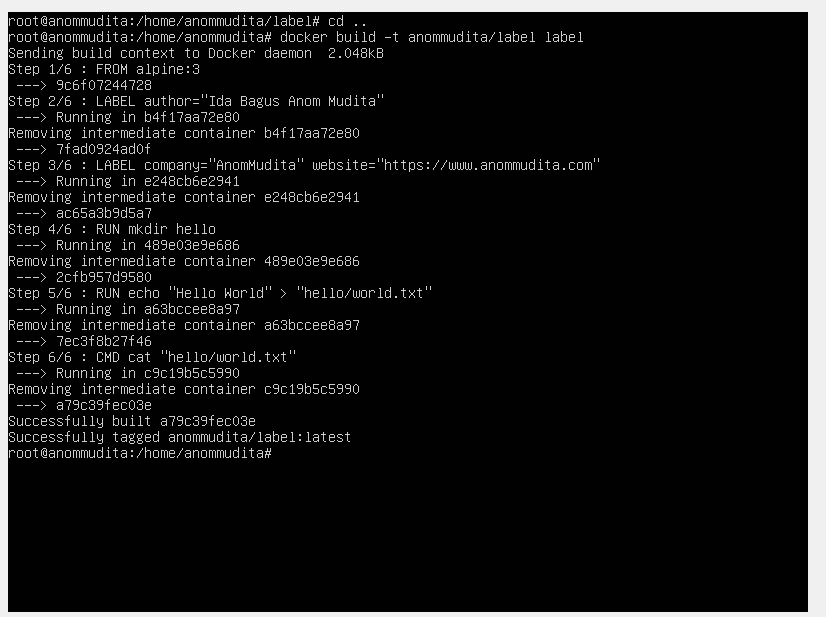
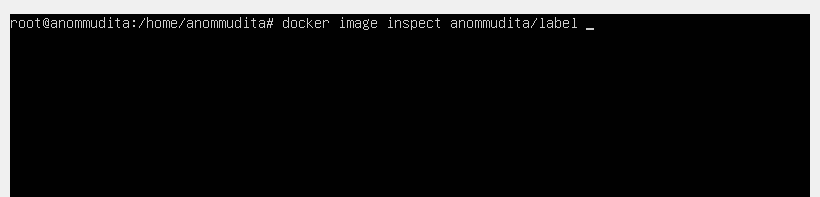
Berikut adalah format instruksi LABEL :

* LABEL<key> =<value>
* LABEL<key1>=<value1><key2>=<value2> 🡺 jika lebih satu sangat menyarankan menggunakan perintah ini

**TAHAPAN LABEL**

1. ****Seperti biasa buat directory berikan nama bebas, lalu buat file dengan nama Dockerfile tanpa format dan buka file tersebut, ikuti perintah dibawah.
2. Setelah membuat Dockerfile, ikuti perintah dibawah. Instruksi yang saya buat ialah membuat folder, membuat file yang berisikan output Hello World ,menjalankan file world.txt dan memberikan informasi(LABEL) terkait author,company, dan link website. Untuk keluar dari text editor tersebut dengan **Esc + : , lalu wq! + enter.** Untuk melihat label yang dibuat tadi maka kita bisa lihat dengan menggunakan fitur Docker Inspect.



1. Kemudian jalankan perintah docker build untuk membuat Image dan hasil label yang kita buat di Dockerfile tidak terlihat diterminal, karena instruksi Label hanya terlihat dengan menggunakan fitur docker inspect. Catatan jika ingin build image, maka kalian harus berada diluar folder Dockerfile yang kalian buat. Gunakan perintah **“cd ..”**
2. Gunakan perintah docker inspect

Hasilnya terlihat, label yang kita buat Dockerfile

1. **Add Instruction**

ADD adalah instruksi yang dapat digunakan untuk menambahkan file dari source ke dalam folder destination di Docker Image(dari laptop kita ke destination). Perintah ADD bisa mendeteksi apakah sebuah file source merupakan file compress seperti tar.gz,gzip, dan lain-lain. Jika mendeteksi file source adalah berupa file kompres, maka secara otomatis file tersebut akan di extract dalam folder destination. Perintah ADD juga bisa mendukung banyak penambahan file sekaligus. Penambahan banyak file sekaligus di Instruksi ADD menggunakan Pattern Go-Lang(Relevan untuk Programmer Go-lang ).

**ADD Instruction Format**

Instuksi ADD memiliki format sebagai berikut :

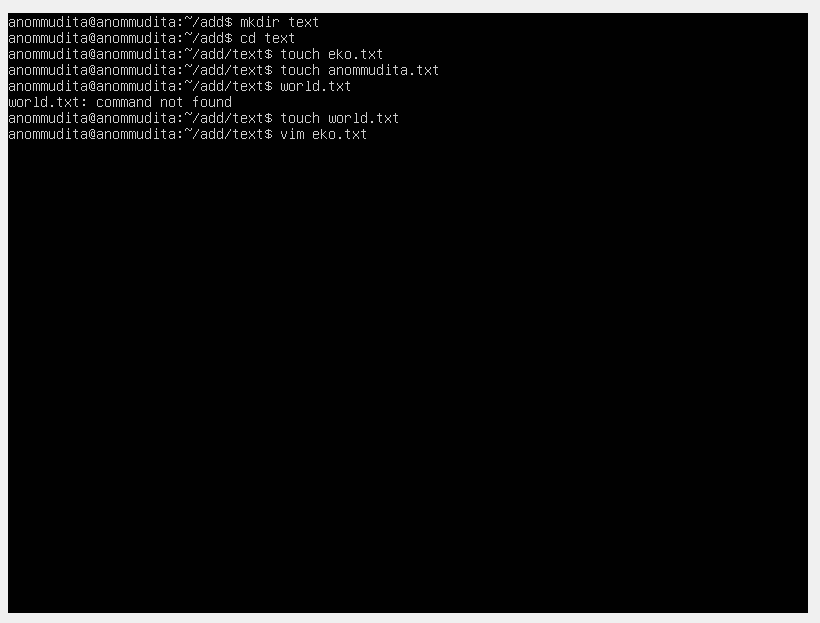
* ADD Source Destination

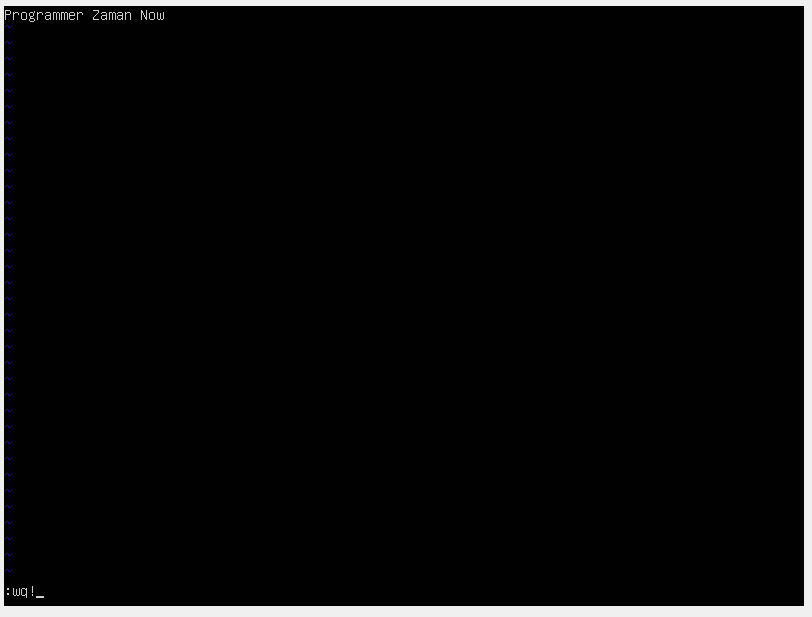
Contohnya :

ADD world.txt hello 🡺 menambahkan file world.txt ke folder hello

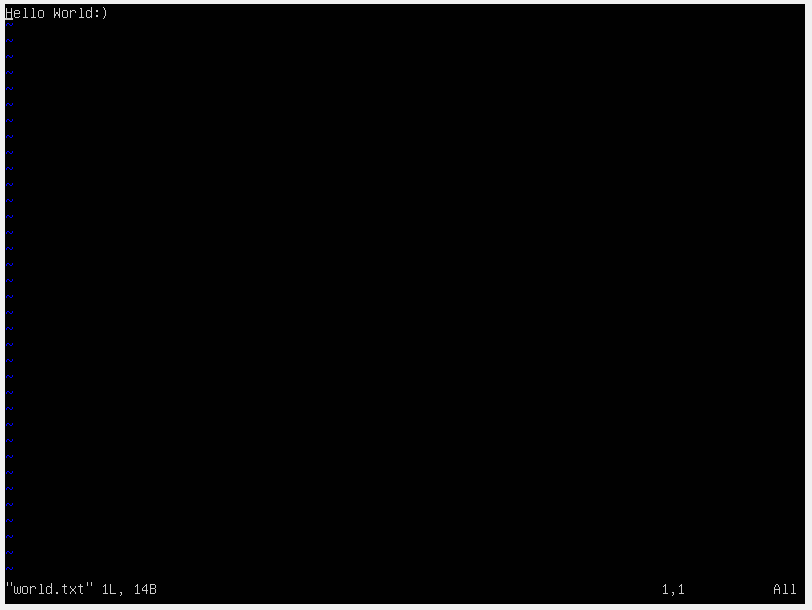
ADD \*.txt hello 🡺 menambahkan semua file .txt ke folder hello

**TAHAPAN ADD Instruction**

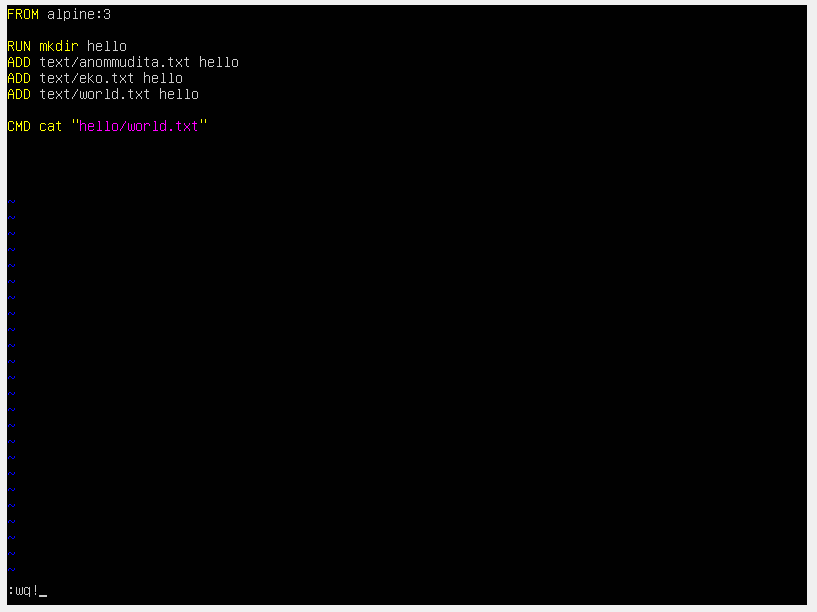
1. Buatlah Directory sebagai Sourcenya dan file berformat .txt, lalu gunakan vim untuk membuka masing-masing file yang sudah dibuat tadi.

file eko.txt berisi

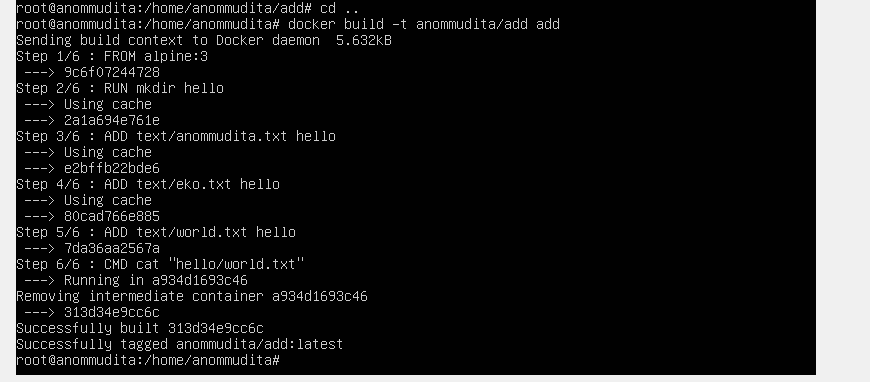
file anommudita.txt berisi

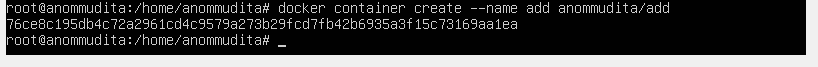
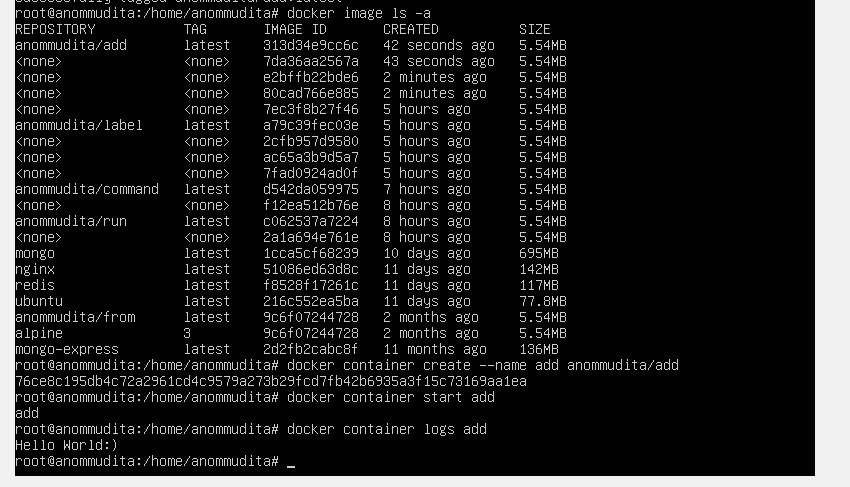
 file world.txt berisi

1. Seperti biasa buat directory dan berikan nama bebas, lalu buat file dengan nama Dockerfile tanpa format dan buka file tersebut, ikuti perintah dibawah
2. Kemudian pada tahapan kali ini kita akan menggunakan ADD Instruction, Jadi file yang didalam folder text akan dipindahkan ke folder hello dan ditampilkan diterminal ketika container dijalankan. . Untuk keluar dari text editor tersebut dengan **Esc + : , lalu wq! + enter**



1. Setelah Dockerfile dibuat maka kita akan membangun Image tersebut dengan perintah dibawah ini. Catatan jika ingin build image, maka kalian harus berada diluar folder Dockerfile yang kalian buat. Gunakan perintah **“cd ..”**



1. Buat container dengan menggunakan Image ADD yang sudah dibuat
2. Hidupkan Container dan lihat logs Container tersebut, karena kita mengeksekusi menggunakan CMD, pada akhirnya instruksi yang terakhir itu yang akan dijalankan oleh CMD.