

*Berdo'alah sebelum mengerjakan. Dilarang berbuat curang.  
Tugas ini untuk mengukur kemampuan anda, jadi kerjakan dengan sepenuh hati.  
Selamat belajar, semoga sukses !*

|  |                           |                        |
|--|---------------------------|------------------------|
| <b>Nama Mahasiswa:</b><br>Javiar Fasyah        | <b>NIM:</b><br>1301164477 | <b>Nilai:</b><br>..... |
| <b>Nama Mahasiswa:</b><br>Fahrur Rozi Syarbini | <b>NIM:</b><br>1301164213 | <b>Nilai:</b><br>..... |
| <b>Nama Mahasiswa:</b><br>Hilmi Triandi N      | <b>NIM:</b><br>1301164286 | <b>Nilai:</b><br>..... |

**Siapkan tools berikut sebelum mengerjakan:**

1. Harus menggunakan linux dengan distro fedora (<https://getfedora.org/id/workstation/>).
2. Buatlah git repository pada <https://github.com/> kemudian push semua kode dan hasil laporan anda ke dalam repository github yang sudah anda buat.
3. Gunakan docker (<https://www.docker.com/>) dan kubernetes (<https://kubernetes.io/id/>) untuk membuat container.
4. Kumpulkan link repository github tersebut sebagai tanda bahwa anda mengerjakan tugas modul ini.
5. Link repository harus berbeda untuk setiap tugasnya. Buatlah markdown yang rapi di setiap repository tugas yang anda kumpulkan.
6. Printscreen program harus dari desktop kelompok anda sendiri, dan harus dari linux yang sudah diinstall. Jika tidak, maka harus mengulang pengerjaan tugasnya.
7. Jangan lupa untuk menuliskan NAMA dan NIM pada laporan.
8. Laporan berbentuk PDF dan dikumpulkan pada link repository github beserta kodenya.
9. Walaupun tugas berkelompok tapi pengumpulan link github harus individu, jika tidak mengumpulkan maka dianggap tidak mengerjakan.

|       |      |        |
|-------|------|--------|
| Nama: | NIM: | Nilai: |
|-------|------|--------|

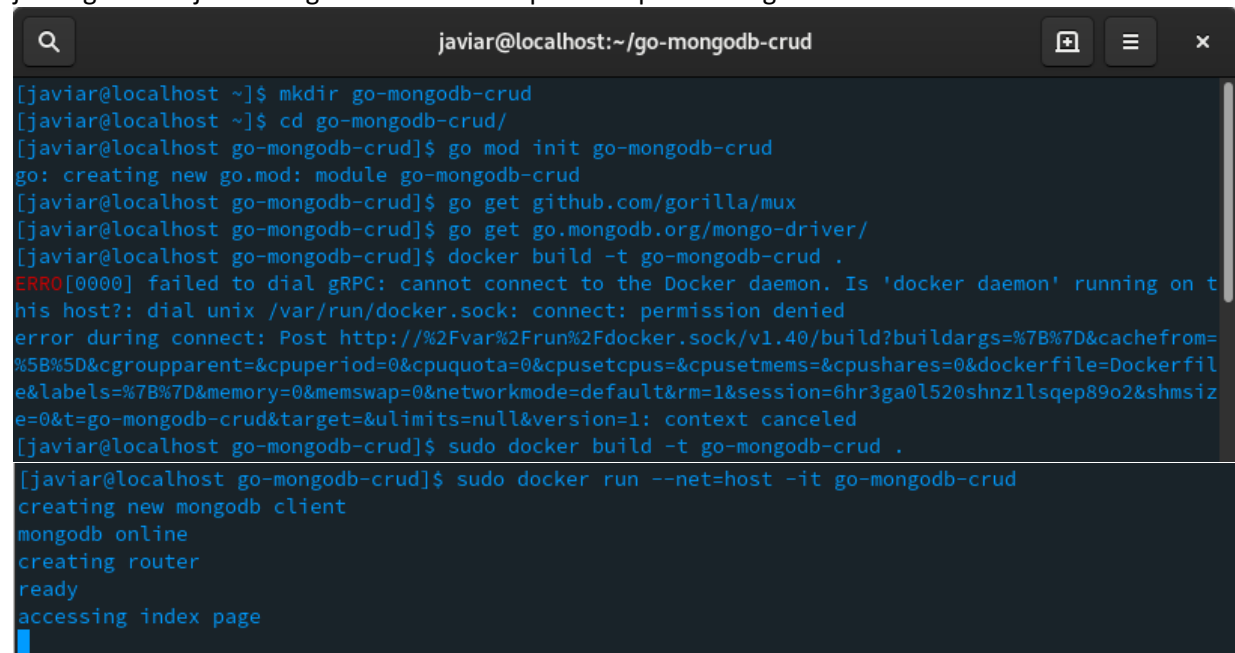
### Soal No 1 (Create container from web application)

Buatlah container dari docker dan kubernetes untuk web aplikasi yang telah anda buat pada tugas modul 4 soal nomor 1.

Jawaban:

Untuk bisa membuat container docker pada aplikasi dengan bahasa Go, aplikasi perlu dibuat dalam lingkungan module, tidak di dalam lingkungan GOPATH. Caranya adalah dengan membuat sebuah direktori baru pada direktori HOME, berpindah ke direktori baru tersebut, dan ketik “go mod init <nama aplikasi>” lalu enter. Perintah tersebut menghasilkan sebuah file go.mod, jangan lupa untuk download kembali package pihak ketiga yang dibutuhkan (go get <link package>). Setelah selesai download, akan terdapat file go.sum yang mencantumkan package-package pihak ketiga yang digunakan aplikasi.

Setelah itu, jangan lupa membuat Docker file sebelum membuat image pada docker, aplikasi akan di-build didalam image tersebut. Buat image dengan perintah “docker build -t <nama image> .” dan semua tahapan-tahapan yang ada di Dockerfile akan dilaksanakan (termasuk membuild aplikasinya). Setelah selesai, container bisa menggunakan image tersebut dengan perintah “docker run --net=host -it <nama image>.” Parameter “--net=host” digunakan agar aplikasi bisa berkomunikasi dengan database yang masih berada di host, nomor port tidak perlu dispesifikasikan jika menggunakan parameter tersebut. Parameter “-it” berarti ia berjalan tidak di background, bisa diganti dengan “-d” jika ingin bekerja di background. Container pun beroperasi dengan baik.



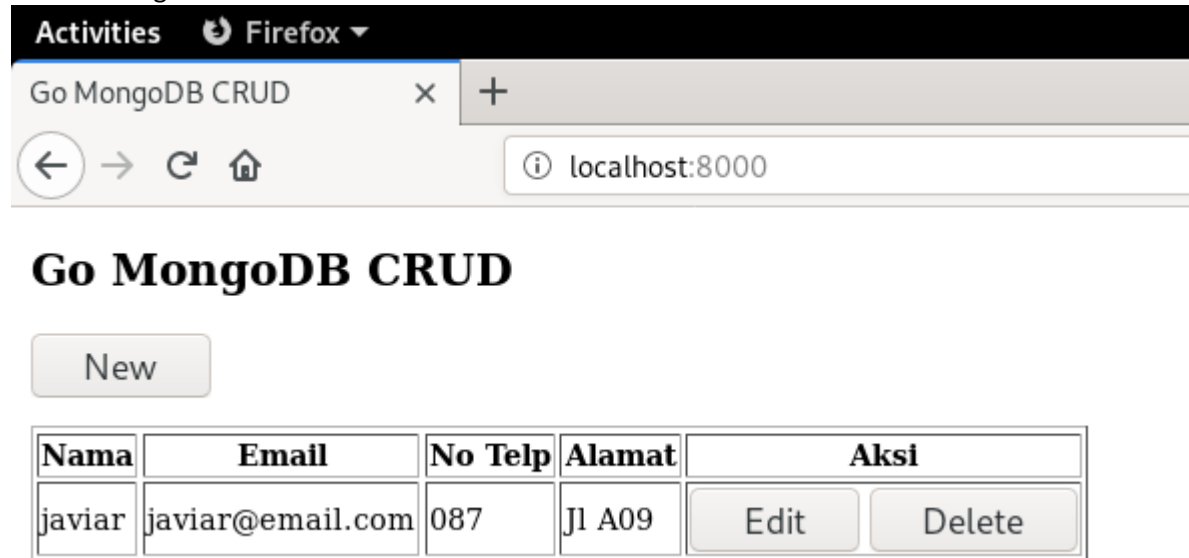
```

javiar@localhost: ~/go-mongodb-crud
[javiar@localhost ~]$ mkdir go-mongodb-crud
[javiar@localhost ~]$ cd go-mongodb-crud/
[javiar@localhost go-mongodb-crud]$ go mod init go-mongodb-crud
go: creating new go.mod: module go-mongodb-crud
[javiar@localhost go-mongodb-crud]$ go get github.com/gorilla/mux
[javiar@localhost go-mongodb-crud]$ go get go.mongodb.org/mongo-driver/
[javiar@localhost go-mongodb-crud]$ docker build -t go-mongodb-crud .
ERRO[0000] failed to dial gRPC: cannot connect to the Docker daemon. Is 'docker daemon' running on this host?: dial unix /var/run/docker.sock: connect: permission denied
error during connect: Post http://%2Fvar%2Frun%2Fdocker.sock/v1.40/build?buildargs=%7B%7D&cachefrom=%5B%5D&cgroupparent=&cpuquota=0&cpusetcpus=&cpusetmems=&cpushares=0&dockerfile=Dockerfile&labels=%7B%7D&memory=0&memswap=0&networkmode=default&rm=1&session=6hr3ga0l520shnzllsqep89o2&shmsize=0&t=go-mongodb-crud&target=&ulimits=null&version=1: context canceled
[javiar@localhost go-mongodb-crud]$ sudo docker build -t go-mongodb-crud .
[javiar@localhost go-mongodb-crud]$ sudo docker run --net=host -it go-mongodb-crud
creating new mongodb client
mongodb online
creating router
ready
accessing index page

```

|       |      |        |
|-------|------|--------|
| Nama: | NIM: | Nilai: |
|-------|------|--------|

Hasil running:



Pada kubernetes, container bisa dibuat secara local dengan menggunakan minikube yang memanfaatkan virtual machine yang telah terinstall pada host. Setelah minikube sudah terinstall pada virtual machine dan berjalan, maka pembuatan image lokal bisa dilakukan dengan memasuki lingkungan virtual machine minikube "\$eval(minikube docker-env)." Setelah itu kita bisa membuat image dengan "docker build -t <nama aplikasi> ." dan menunggu proses selesai.

Sebelum membuat deployment dan service, tag terlebih dahulu image dengan "docker tag <id repository> <nama aplikasi>:<versi>" untuk menghindari error pulling image. Barulah kita mendeploy dengan "kubectl create deployment <nama deployment> --image=<nama image aplikasi>:<versi>." Untuk bisa mengaksesnya, kita perlu mengeksposnya dengan "kubectl expose deployment <nama deployment> --type=NodePort --port=<port>," dan selanjutnya "kubectl get pod." Dari sini bisa diketahui alamat untuk mengakses deployment dengan "minikube <nama deployment> --url," namun sayang terjadi error yang tidak diketahui saat ingin membuka alamat aksesnya.