

---

# POLITEKNIK NEGERI LHOKSEUMAWE

JURUSAN TEKNOLOGI INFORMASI DAN KOMPUTER

TEKNOLOGI REKAYASA KOMPUTER JARINGAN


---




---


## TRKJ-6435 — CLOUD COMPUTING

---

DOSEN PENGAMPU: Muhammad Davi, S.Kom., M.Cs. 

Gedung Terpadu Jurusan Teknologi Informasi dan Komputer 

muhammad.davi@pnl.ac.id 

2023-2024 Genap 

## **Daftar Isi**

<b>1</b>	<b>Lab 1: Docker Image "hello-world"</b>	<b>2</b>
<b>2</b>	<b>Lab 2: Shell</b>	<b>3</b>
<b>3</b>	<b>Lab 3: Membuat Akun di Docker Hub</b>	<b>4</b>

# 1 Lab 1: Docker Image "hello-world"

Pertama periksa dulu versi docker yang digunakan dengan perintah berikut:

```
1 $ docker version
```

Tuliskan versi docker disini: \_\_\_\_\_

Dari output diatas, Docker diimplementasikan dengan bahasa pemrograman: \_\_\_\_\_

Anda ingin menjalankan sebuah Image yang bernama hello-world. Image ini tersedia di Docker Hub. Namun sebelum menjalankan instruksi tersebut, lebih dahulu periksa apakah ada container dan image yang aktif.

```
1 $ docker images
```

Jika ternyata ada images yang sudah terdaftar, maka hapus lebih dahulu dengan instruksi dibawah ini.

```
1 // hentikan proses docker yang masih aktif
2 $ docker stop $(docker ps -a -q)
3
4
5 // hapus semua kontainer yang ada
6 $ docker rm $(docker ps -a -q)
7
8 // hapus semua images
9 $ docker rmi $(docker images -a -q)
10
11
12 // atau hapus semua image yang ada dan kontainer yg stop
13 $ docker system prune -a
```

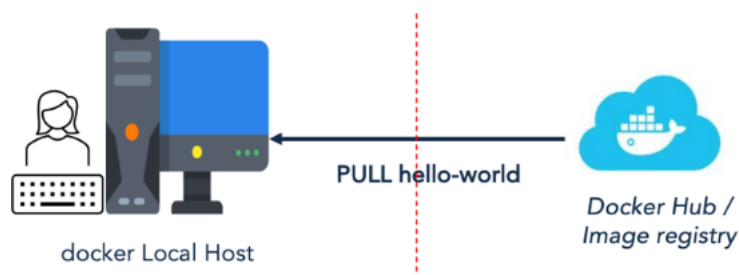
Opsi -a menampilkan semua elemen, sedangkan -q menampilkan ID nya saja. Ketiga instruksi ini dilakukan hanya jika ada proses tersebut yang aktif, urutannya penting, mulai dari proses, kontainer dan image. Selanjutnya lakukan instruksi berikut:

```
1 $ docker run hello-world
```

Pesan pertama yang muncul adalah:

Unable to find image 'hello-world:latest' locally

Docker berusaha untuk menjalankan image yang bernama hello-world, tapi tidak ditemukan di localhost. Oleh karena itu docker secara otomatis mencari hello-world di Docker Hub. Image tersebut ditemukan disana, dan selanjutnya docker mengambil (pull) image tersebut ke localhost, kemudian menjalankannya setelah selesai menarik image tersebut.



Bila Anda menjalankan untuk kedua kali, maka reaksi dari localhost lebih cepat karena image hello-world sudah berada di localhost, sehingga tidak lagi perlu mengunduh (pull) dari docker hub.

```
1 $ docker images
2 $ docker run hello-world
```

Tuliskan Image ID disini: \_\_\_\_\_  
Kalau punya \_\_\_\_\_ Image ID-nya: \_\_\_\_\_

## 2 Lab 2: Shell

Anda ingin mengaktifkan Image seperti Lab 1, tapi kali ini Anda ingin masuk ke kontainer tersebut untuk bekerja diatas kontainer seperti bekerja diatas sebuah Sistem Operasi yang dengan menggunakan Shell.

Sebelumnya pilih dulu Sistem Operasi yang akan digunakan, umumnya versi Linux. Ada 3 versi yang populer, yaitu RedHat, Ubuntu, Suse dan turunan lainnya seperti CentOS, Alpine, Linux Mint, Manjaro dan lainnya. Masing-masing punya kelebihan atau konsentrasi di segment tertentu.

Kita akan coba Linux yang relatif paling ringan, yaitu Alpine. Cari dulu di Docker Hub, apakah image alpine tersedia.

```
1 $ docker search alpine
```

Ada banyak image alpine terdaftar disitu, lengkap dengan utilitas lainnya. Yang kita butuhkan adalah alpine.

```
1 $ docker pull alpine
2 $ docker images
```

Perhatikan bahwa Tag adalah versi, dan bila tidak diberikan, nilai default adalah "latest". Sehingga instruksi berikut ekuivalen dengan sebelumnya.

```
1 $ docker pull alpine:latest
```

Jalankan sebagai kontainer dan bekerja dengan shell (bash):

```
1 $ docker run -it alpine
```

Opsi -it artinya jalankan sebagai interactive terminal (sebagai terminal). Jalankan beberapa instruksi shell untuk mencoba:

```
1 # id
2 # date
3 # hostname
4 # netstat -r
5 # exit
```

Anda bisa melihat proses aktif di kontainer dengan ps (proses status)

```
1 $ docker ps
```

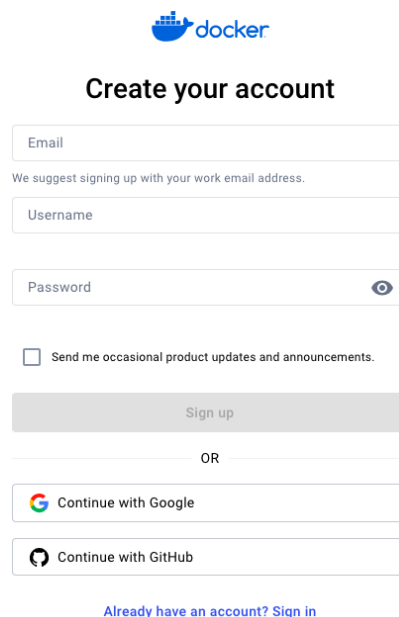
Tapi tidak ada kontainer yang aktif. Untuk melihat jejak proses sebelumnya, gunakan opsi -a (all).

```
1 $ docker ps -a
```

Banyak proses yang sudah berjalan dan selesai dengan status Exited.

### 3 Lab 3: Membuat Akun di Docker Hub

Persiapkan email yang valid yang dapat segera diakses. Validasi token akan diberikan melalui akun email tersebut. Arahkan browser ke: <https://hub.docker.com>. Pilih Sign up untuk membuat akun baru.







This site is protected by reCAPTCHA and the Google [Privacy Policy](#) and [Terms of Service](#) apply.

By creating an account I agree to the [Subscription Service Agreement](#), [Privacy Policy](#), [Data Processing Terms](#).

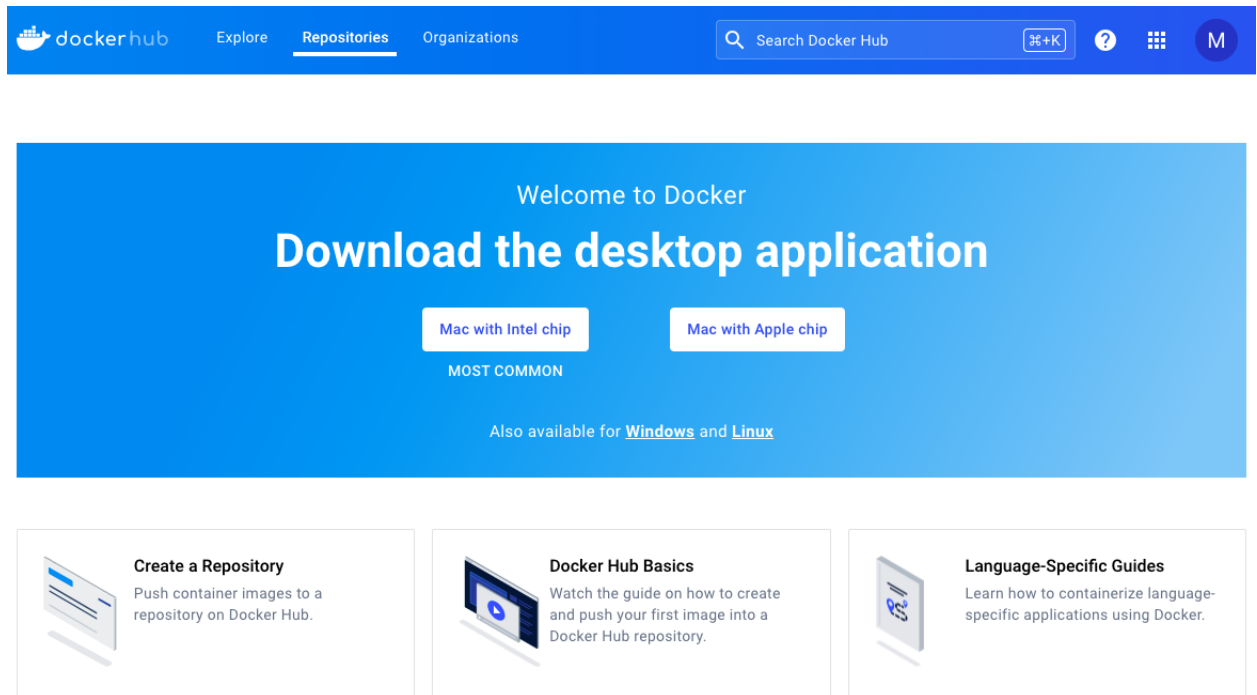
Tulis identitas, alamat email, dan password untuk login ke docker hub. Ikuti instruksi untuk validasi dan verifikasi (biasanya melalui pertanyaan pada sebuah gambar yang diberikan). Pilih Skema yang Free (\$0), "**Continue with Personal**".

#### Choose your Docker Core subscription

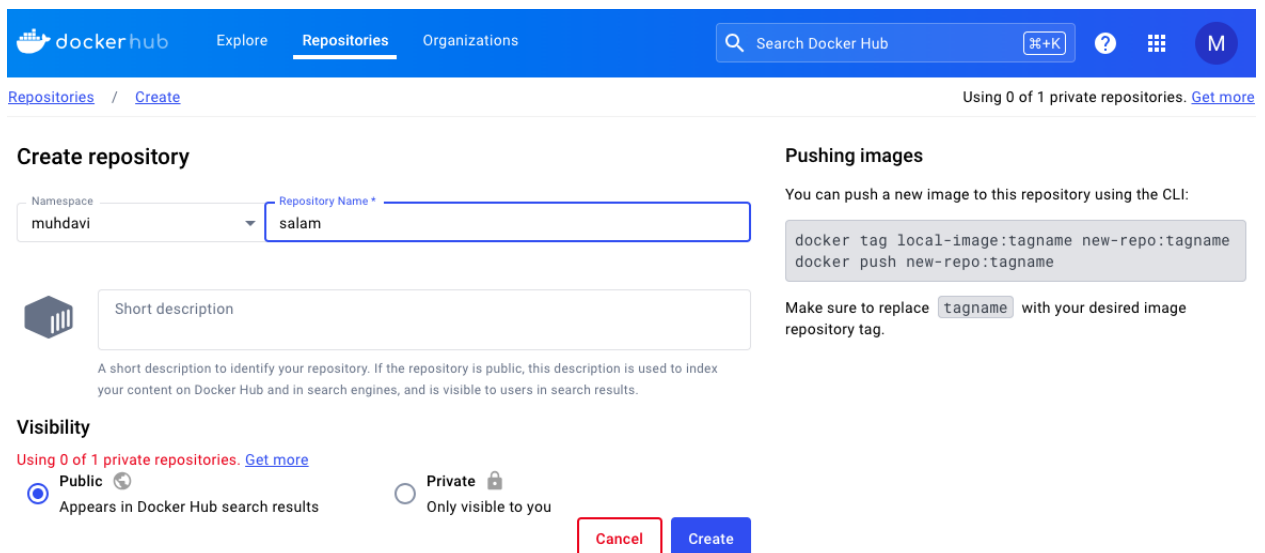
Use of Docker Desktop at a company of more than 250 employees OR more than \$10 million in annual revenue requires a paid subscription (Pro, Team, or Business).

Personal	Pro	Team	Business
For new developers and/or students getting started with containers.	For professional developers who want to accelerate innovation.	For development teams looking to increase collaboration and agility.	For businesses seeking to establish an enterprise-grade development approach.
<b>\$0</b>	<b>\$5</b> / month	<b>\$9</b> / seat / month <sup>⓪</sup>	<b>\$24</b> / seat / month <sup>⓪</sup>
<ul style="list-style-type: none"><li>✓ Docker Desktop Personal <sup>⓪</sup></li><li>✓ Unlimited public repositories</li><li>✓ Docker Engine + Kubernetes <sup>⓪</sup></li><li>✓ 200 image pulls per 6 hours</li><li>✓ 3 Scout-enabled repositories</li><li>✓ Local Scout analysis</li></ul>	<p>← Everything in Personal, plus:</p> <ul style="list-style-type: none"><li>✓ Docker Desktop Pro <sup>⓪</sup></li><li>✓ Unlimited private repositories</li><li>✓ 5,000 image pulls per day</li><li>✓ 5 concurrent builds <sup>⓪</sup></li><li>✓ Synchronized file shares</li><li>✓ Local Scout analysis and remediation</li><li>✓ 5 day support response</li></ul>	<p>← Everything in Pro, plus:</p> <ul style="list-style-type: none"><li>✓ Docker Desktop Team <sup>⓪</sup></li><li>✓ Unlimited teams</li><li>✓ 15 concurrent builds <sup>⓪</sup></li><li>✓ Add users in bulk</li><li>✓ Audit logs <sup>⓪</sup></li><li>✓ Up to 100 seats</li><li>✓ Role-based access control</li><li>✓ 2 day support response</li></ul>	<p>← Everything in Team, plus:</p> <ul style="list-style-type: none"><li>✓ Docker Desktop Business <sup>⓪</sup></li><li>✓ Hardened Docker Desktop <sup>⓪</sup></li><li>✓ Single Sign-On (SSO)</li><li>✓ SCIM user provisioning</li><li>✓ VDI support</li><li>✓ Private Extensions Marketplace</li><li>✓ Image and Registry Access Management</li><li>✓ Purchase via invoice</li><li>✓ Priority case routing</li><li>✓ Proactive case monitoring</li><li>✓ 24-hour support response</li></ul>
			
Included minutes: 400 mins / month Included cache: 100 GB	Included minutes: 400 mins / month Included cache: 100 GB	Included minutes: 400 mins / month Included cache: 100 GB	Included minutes: 800 mins / month Included cache: 200 GB
<a href="#">Continue with Personal</a>	<a href="#">Buy Now</a>	<a href="#">Buy Now</a> ▾	<a href="#">Buy Now</a> ▾

Abaikan Download the desktop application. Pilih **Create a Repository**.



Pada Menu **Create repository**, tulis nama repository - pada contoh adalah "salam". Berikan deskripsi terkait. Visibility adalah "public", yang mana repository ini dapat diakses oleh siapa saja di Internet.



Hasil Pembuatan repository. Disebelah kanan adalah informasi untuk melakukan "push" yaitu menyimpan docker image di docker hub. Instruksi tersebut dilakukan dari komputer client setelah membuat image docker baru.

