



Nama: **Muhammad Qomarudin (120140116)**
Mata Kuliah: **Sistem Operasi (IF2223)**

Tugas Ke: **03**
Tanggal: **01/05/2022**

1 Tujuan HandsOn

Tujuan dari HandsOn kali ini adalah untuk membuat mahasiswa memahami tentang Docker. Terutama tentang perintah-perintah dasar yang digunakan dalam menggunakan Docker dan apa fungsi perintah tersebut, dan juga tentang istilah-istilah yang sering digunakan dalam penggunaan Docker.

2 Instalasi

Pada HandsOn kali ini saya menggunakan Docker yang dijalankan pada sistem operasi Windows

2.1 Requirements

WSL 2 / Windows Subsystem for Linux

WSL 2 atau Windows Subsystem for Linux adalah fitur yang dikembangkan oleh Microsoft yang memungkinkan sistem operasi Windows dapat menjalankan GNU/Linux untuk. Untuk menginstall WSL 2, dapat kita lakukan melalui Windows PowerShell, dengan perintah berikut:

```
1 wsl --install #wsl default (ubuntu)
2
3 wsl --install [dirto name] #install wsl untuk dirto khusus.
```

```
> pwsh 19s 943ms
>> wsl --install -d ubuntu
Downloading: Ubuntu
Installing: Ubuntu
Ubuntu has been installed.
Launching Ubuntu ...
> pwsh 8s 459ms
>>
Enter new UNIX username: qomar21
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.10.102.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sun Apr 24 13:43:57 WIB 2022

System load:  0.03          Processes:      8
Usage of /:   0.4% of 250.98GB Users logged in: 0
Memory usage: 7%           IPv4 address for eth0: 172.27.73.209
Swap usage:   0%

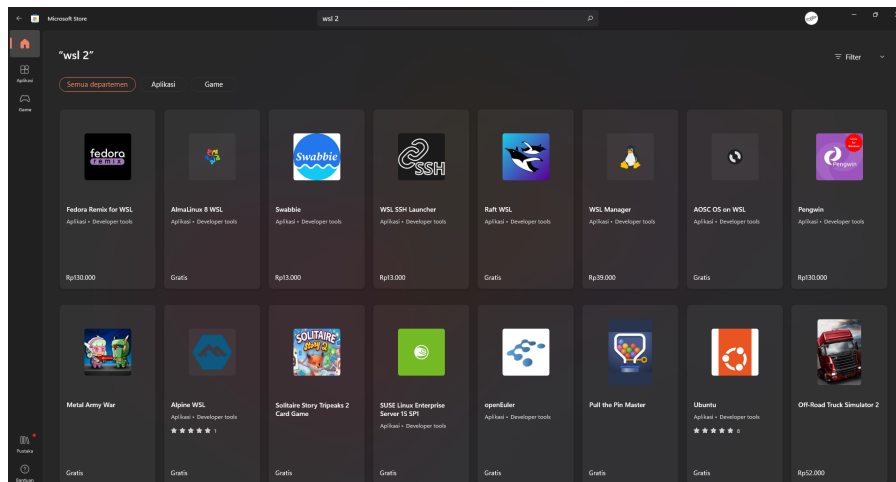
0 updates can be installed immediately.
0 of these updates are security updates.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

This message is shown once once a day. To disable it please create the
/home/qomar21/.hushlogin file.
qomar21@MASQOMAR-21:~$
```

Gambar 1: Install WSL 2 melalui Windows PowerShell

selain menggunakan PowerShell, WSL juga dapat diinstall melalui Microsoft Store. dengan cara memasukan kata kunci "wsl" pada kolom Search. kemudian memilih salah satu distro yang akan di install, kemudian klik pada "get"

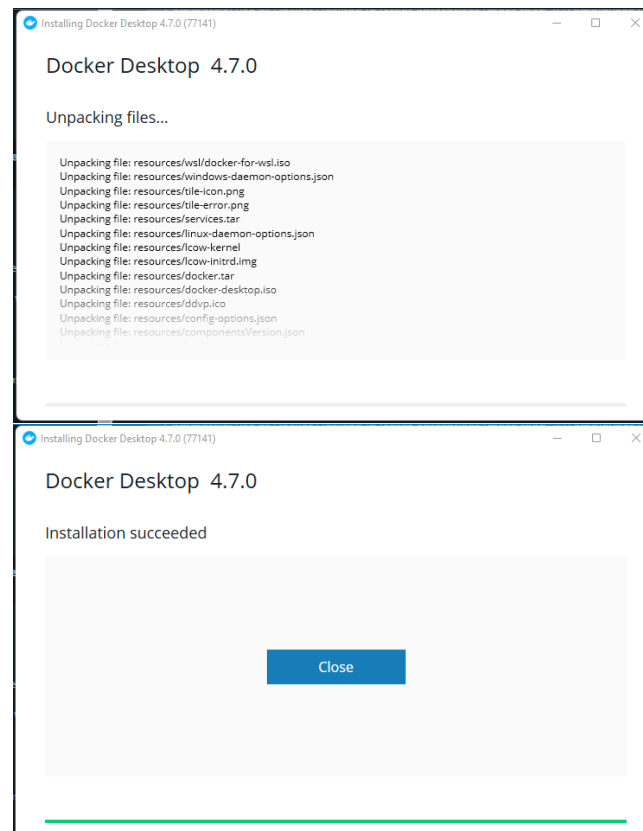


Gambar 2: Install WSL 2 melalui Microsoft Store

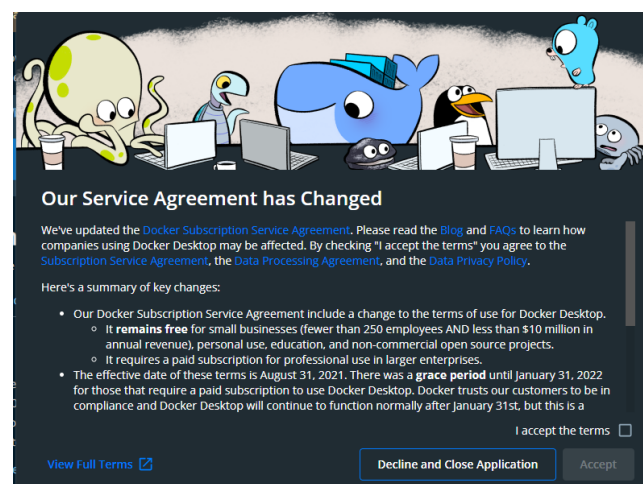
2.2 install Docker

Setelah WSL 2 diinstall, kita bisa menginstall Docker. Dengan cara kita mengunduh file instalasi Docker melalui official website resminya [Docker for Windows](https://docs.docker.com/docker-for-windows/). setelah selesai mengunduh, jalankan file unduhan yang telah kita unduh tersebut. maka akan muncul Window seperti gambar di bawah ini, kemudian klik "Ok" maka mulai mengunduh package-package yang diperlukan untuk menginstall Docker.





setelah docker terinstall, jalankan Docker tersebut.



kemudian klik "Accept". dan Docker siap di gunakan.

3 percobaan

3.1 Hello-World

setelah Docker berhasil diInstall, maka kita dapat menjalankan perintah perintah Docker tersebut pada PowerShell atau Command Prompt. Untuk mengeceknya dapat kita ketikkan perintah berikut pada PowerShell

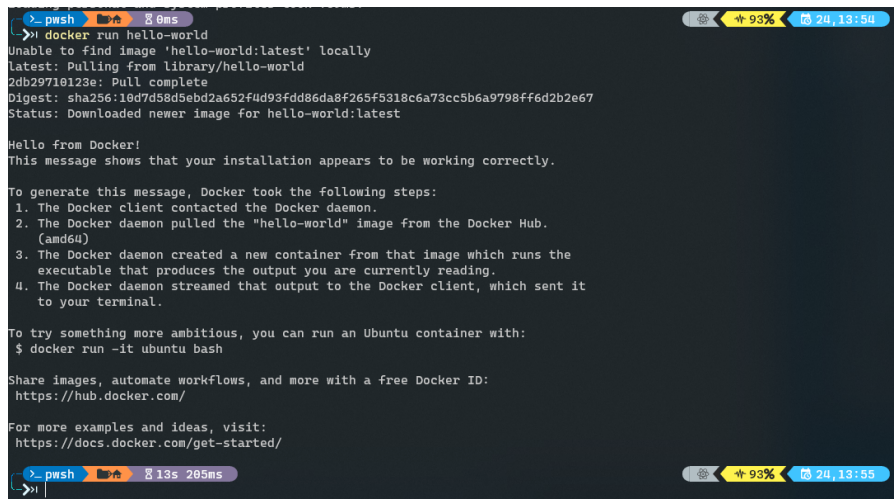
```
1 docker run Hello-World
```

maka outpunnya akan keluar sebagai berikut

```

1  Unabel to find image 'hello-world:latest' locally
2  latest: Pulling from library/hello-world
3  2db29710123e: Pull complete
4  Digest: sha256:10d7d58d5ebd2a652f0d93fdd86da8f265f5318c6a73cc5b6a9798ff6d2b2067
5  Status: Downloaded newer image for hello-world:latest
6
7  Hello from Docker!
8  This message shows that your installation appears to be working correctly
9  ....

```



Gambar 4: Docker run hello-world

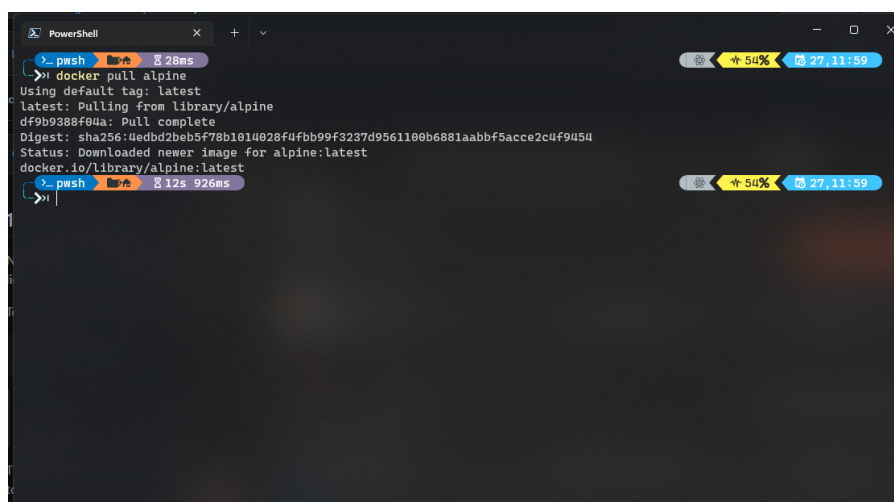
3.2 Alpine linux

selanjutnya kita akan menjalankan image pertama yaitu *alpine linux*. Untuk menjalankan image tersebut kita perlu mengunduh/pulling dengan cara mengetikan perintah berikut

```

1  Docker Pull Alpine

```



Gambar 5: Pulling Alpine Container

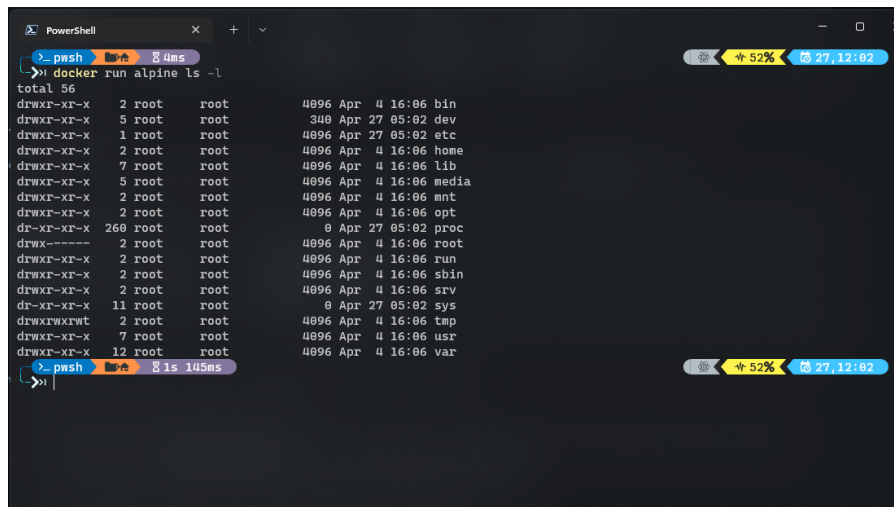
selanjutnya kita akan mencoba menjalankan salah Container dengan mengetikan perintah berikut

```

1  Docker run alpine ls -l

```

maka outputnya sebagai berikut



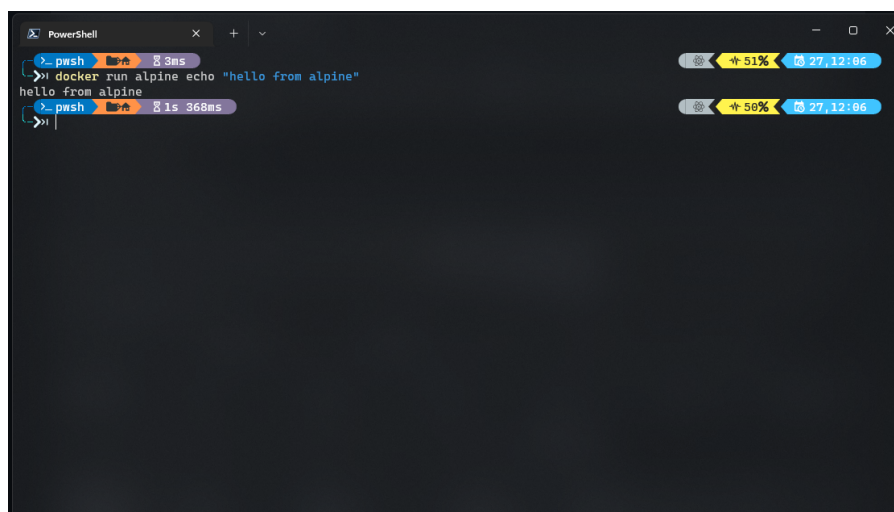
```
PowerShell
> pwsh 8 ms
>> docker run alpine ls -l
total 56
drwxr-xr-x 2 root root 4096 Apr 4 16:06 bin
drwxr-xr-x 5 root root 340 Apr 27 05:02 dev
drwxr-xr-x 1 root root 4096 Apr 27 05:02 etc
drwxr-xr-x 2 root root 4096 Apr 4 16:06 home
drwxr-xr-x 7 root root 4096 Apr 4 16:06 lib
drwxr-xr-x 5 root root 4096 Apr 4 16:06 media
drwxr-xr-x 2 root root 4096 Apr 4 16:06 mnt
drwxr-xr-x 2 root root 4096 Apr 4 16:06 opt
dr-xr-xr-x 260 root root 0 Apr 27 05:02 proc
drwx----- 2 root root 4096 Apr 4 16:06 root
drwxr-xr-x 2 root root 4096 Apr 4 16:06 run
drwxr-xr-x 2 root root 4096 Apr 4 16:06/sbin
drwxr-xr-x 2 root root 4096 Apr 4 16:06/srv
dr-xr-xr-x 11 root root 0 Apr 27 05:02/sys
drwxrwxrwt 2 root root 4096 Apr 4 16:06/tmp
drwxr-xr-x 7 root root 4096 Apr 4 16:06/usr
drwxr-xr-x 12 root root 4096 Apr 4 16:06/var
> pwsh 81s 145ms
>>
```

Gambar 6: Docker run alpine ls -l

setelah itu kita akan mencoba mengetika "hello from alpine" melalui Container tersebut dengan perintah sebagai berikut

```
1 Docker run alpine echo "hello from alpine"
```

maka outputnya sebagai berikut.



```
PowerShell
> pwsh 8 ms
>> docker run alpine echo "hello from alpine"
hello from alpine
> pwsh 81s 368ms
>>
```

Gambar 7: Docker run alpine echo

selanjutnya kita akan mencoba masuk kedalam bash atau Command line dari Container alpine tersebut. Untuk masuk kedalam kontaner Alpine kita tidak bisa hanya menjalankan Container tersebut dan memberi path untuk bashnya saja. Tetapi kita juga harus manambah option ketika menjalankan Container tersebut. perintah yang dapat kita gunakan sebagai berikut.

```
1 docker run alpine /bin/sh #tidak bisa masuk kedalam bash
2
3 docker run alpine -it /bin/sh
```

seperti tangkap layar berikut.

```
PowerShell
> docker run alpine /bin/sh
> docker run alpine -it /bin/sh
/ # ls
bin  dev  etc  home  lib  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
/ # ps -a
PID  USER  TIME  COMMAND
1  root   0:00  /bin/sh
9  root   0:00  ps -a
/ # uname -a
Linux 2c8e2c38679d 5.10.102.1-microsoft-standard-WSL2 #1 SMP Wed Mar 2 00:30:59 UTC 2022 x86_64 Linux
/ # echo "hello world"
hello world
/ # exit
>
```

Gambar 8: masuk bash alpine

Option -it berguna untuk masuk dan memberikan perintah pada bash Container tersebut. -i hanya untuk memberikan perintah saja, tetapi tidak akan masuk kedalam bash Container tersebut. sedangkan -t adalah option untuk masuk kedalam bash Container saja namun tidak dapat memberikan perintah ke dalamnya.

selanjutnya kita akan mengeceknya Container apa saja yang sudah pernah kita jalankan dengan cara mengetikan perintah berikut.

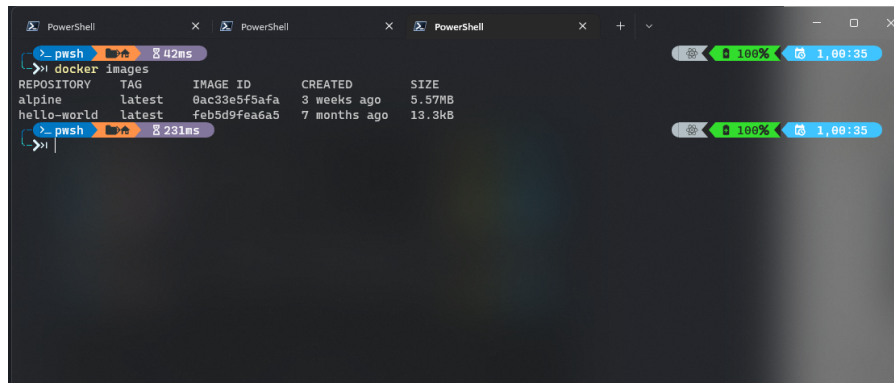
```
1 docker ps -a
```

```
PowerShell
> docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
2c8e2c38679d   alpine   "/bin/sh"  4 minutes ago  Exited (0)  About a minute ago  awesome_lehmann
55a7511196ff   alpine   "/bin/sh"  4 minutes ago  Exited (0)  4 minutes ago    elegant_ferhat
fdd42079329b   alpine   "/bin/sh"  4 minutes ago  Exited (0)  4 minutes ago    hopeful_bardeen
7d249539068c   alpine   "\\bin\\sh"  5 minutes ago  Created                                clever_jennings
c29abae138b9   alpine   "echo 'hello from al..."  6 minutes ago  Exited (0)  6 minutes ago    relaxed_lehmann
9d03e1u72beu   alpine   "ls -l"    11 minutes ago  Exited (0)  11 minutes ago   distracted_sammet
eff6b6f119da   alpine   "ls -l"    11 minutes ago  Exited (0)  11 minutes ago   agitated_chandrasekhar
f8f08febed26   alpine   "ls -l"    11 minutes ago  Exited (0)  11 minutes ago   youthful_volhard
da841dd9ef00   hello-world  "/hello"    2 days ago    Exited (0)  2 days ago       vibrant_kare
>
```

Gambar 9: cek container yang pernah di jalankan

selain itu, kita juga dapat mengecek image apa saja yang pernah kita unduh (pull). dengan mengetikan perintah berikut.

```
1 docker images
```

Gambar 10: cek images

4 Jawaban pertanyaan HandsOn

Apa itu Docker

Docker merupakan salah satu bentuk virtualisasi yang memungkinkan developer untuk membuat, mengemas, dan menjalankan aplikasi dalam sebuah Container. Docker sebenarnya mirip dengan Virtual Machine (VM), karena keduanya sama-sama memerlukan *resource isolation* yang artinya sama-sama mengisolasi sumber daya untuk dapat dijalankan. Namun, Docker dan VM memiliki perbedaan yang mencolok, yaitu pada resource yang diperlukan. Sebagai ilustrasi, pada saat menjalankan sebuah aplikasi dengan menggunakan Virtual Machine, maka kita perlu menyiapkan sistem operasi yang lengkap yang mana akan membutuhkan resource yang cukup besar. Berbeda dengan menggunakan Docker, kita tidak perlu menyiapkan seluruh sistem operasi yang lengkap, kita hanya perlu menyiapkan hal-hal yang kita perlukan dalam aplikasi tersebut, sehingga resource yang kita perlukan juga akan lebih sedikit. Hal ini dapat terjadi karena Docker memanfaatkan Kernel Linux pada level Host OS untuk digunakan secara bersama-sama (Share) oleh container.

apa fungsi perintah "docker run"

perintah "docker run" berfungsi untuk menjalankan sebuah Container. ketika "docker run" diketikkan oleh user maka docker akan mencari image yang sesuai dengan Container tersebut secara lokal kemudian menjalankannya. jika tidak terdapat image yang sesuai dengan Container tersebut di lokal, docker akan mencarinya di global dan akan mengunduh (pulling) image tersebut ke dalam lokal, dan kemudian menjalankan Container tersebut.

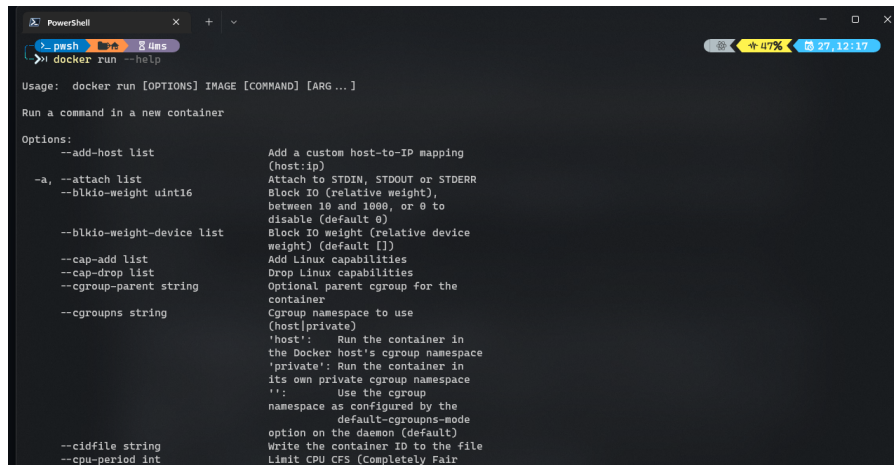
berikut Command yang dapat digunakan untuk menjalankan sebuah Container

```

1  docker run [image] #hanya menjalankan Containernya
2
3  docker run [image] [Command] #menjalankan Container sekaligus Commandnya (contoh Command = echo)
4
5  docker run [option] [image] [Command] #menjalankan image beserta option yang terdapat pada docker
   (Contoh option = -i)

```

untuk dapat mencari option yang sesuai dengan kebutuhan kita dapat kita ketikkan "docker run -help". di dalamnya terdapat banyak macam-macam option yang disediakan oleh Docker



```

PowerShell
> docker run --help

Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run a command in a new container

Options:
  --add-host list          Add a custom host-to-IP mapping
                           (host:ip)
  -a, --attach list        Attach to STDIN, STDOUT or STDERR
  --blkio-weight uint16    Block IO (relative weight),
                           between 10 and 1000, or 0 to
                           disable (default 0)
  --blkio-weight-device list
                           Block IO weight (relative device
                           weight) (default [])
  --cap-add list           Add Linux capabilities
  --cap-drop list          Drop Linux capabilities
  --cgroup-parent string   Optional parent cgroup for the
                           container
  --cgroupns string        Cgroup namespace to use
                           (host|private)
                           'host': Run the container in
                           the Docker host's cgroup namespace
                           'private': Run the container in
                           its own private cgroup namespace
                           '': Use the cgroup
                           namespace as configured by the
                           default-cgroupns-mode
                           option on the daemon (default)
  --cidfile string         Write the container ID to the file
  --cpu-period int         Limit CPU CFS (Completely Fair

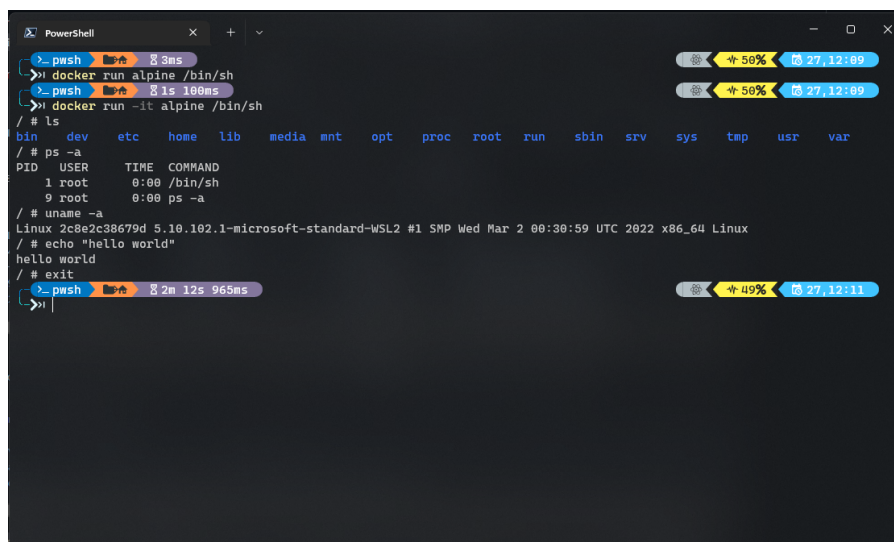
```

Gambar 11: help untuk docker run

Apa itu Conatainer

apa fungsi perintah "docker run -it"

sama halnya dengan perintah "docker run" biasa, perintah "docker run -it" juga berfungsi untuk menjalankan sebuah Container baru, tetapi terdapat penambahan pada bagian [Command]. Command "-it" meemungkinkan docker dapat mengakses Command line yang ada pada Container tersebut. sebenarnya comannnd "-it" itu terdiri dari 2 buah commadn yaitu "-i" untuk melakukan input dan "-t" untuk masuk kedalam Command line dari Container tersebut. seperti gambar berikut.



```

PowerShell
> docker run alpine /bin/sh
> docker run -it alpine /bin/sh

/ # ls
bin  dev  etc  home  lib  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var

/ # ps -a
PID  USER  TIME  COMMAND
1    root   0:00  /bin/sh
9    root   0:00  ps -a

/ # uname -a
Linux 2c8e2c38679d 5.10.102.1-microsoft-standard-WSL2 #1 SMP Wed Mar 2 00:30:59 UTC 2022 x86_64 Linux

/ # echo "hello world"
hello world

/ # exit

>

```

Apa itu images

Apa itu deamon