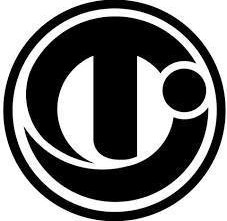
# Laporan Praktek Topik Khusus 9

Instalasi & Konfigurasi Docker



## SEMESTER VI

**Muhammad Abel Al-fahrezi 2211083034**

# PROGRAM STUDI D4 TEKNOLOGI REKAYASA PERANGKAT LUNAK

**JURUSAN TEKNOLOGI INFORMASI**

## Landasan Teori

* 1. **Windows Subsystem for Linux (WSL) :**

WSL adalah fitur dari sistem operasi Windows yang memungkinkan pengguna untuk menjalankan lingkungan GNU/Linux secara langsung di Windows, tanpa harus menggunakan mesin virtual (Virtual Machine) secara penuh. Versi terbaru, **WSL2**, menggunakan kernel Linux asli yang mendukung sistem file yang lebih cepat dan kompatibilitas lebih baik dengan tool berbasis Linux seperti Docker. Dengan WSL2, pengguna Windows dapat menjalankan container Docker **secara native**, tanpa harus bergantung pada teknologi virtualisasi berat seperti Hyper-V atau VirtualBox.

* 1. **Docker dan Container :**

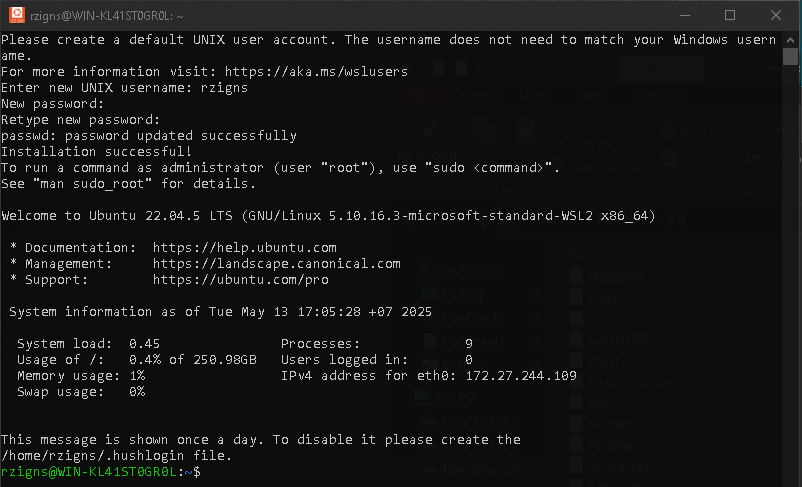
Nginx Docker adalah platform open-source yang dirancang untuk mengembangkan, mengirimkan, dan menjalankan aplikasi dalam unit terisolasi yang disebut container. Container berisi semua dependensi aplikasi, seperti kode, pustaka, dan konfigurasi, sehingga menjamin konsistensi aplikasi saat dipindahkan antar lingkungan. Docker bekerja sangat baik di lingkungan Linux karena container dibangun di atas fitur kernel Linux seperti cgroups dan namespaces. Oleh karena itu, menjalankan Docker melalui WSL2 menciptakan kondisi ideal untuk penggunaan container di sistem operasi Windows.

## Tools

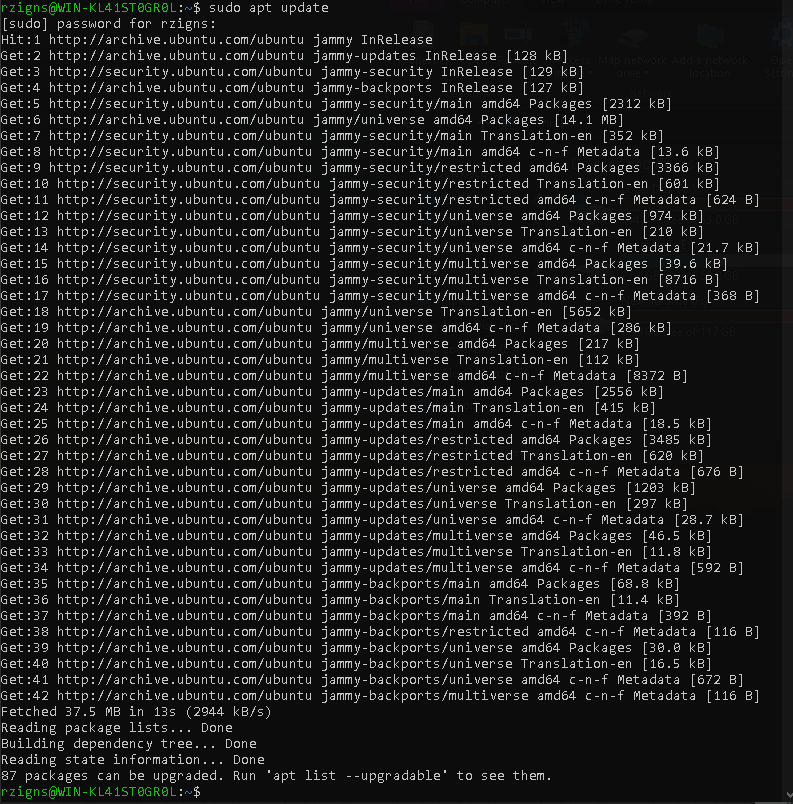
* Ubuntu (melalui WSL2)

## Langkah Kerja

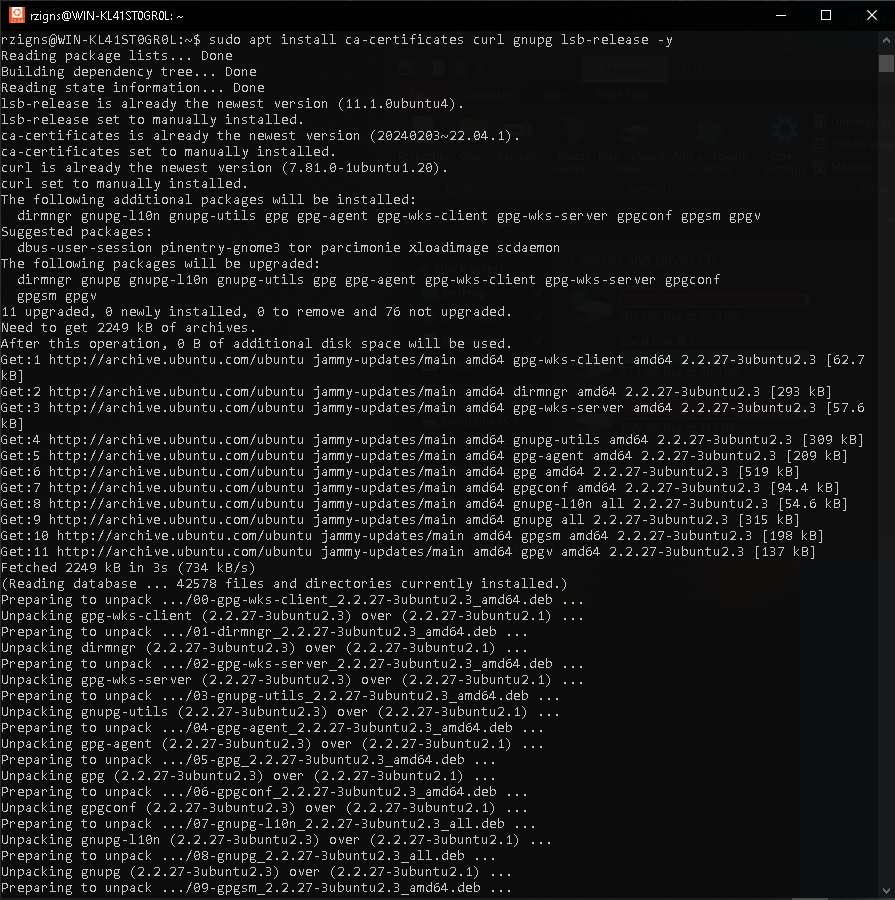
* 1. **Instalasi dan Konfigurasi**
* Create user ubuntu for wsl



* *sudo apt update* Memperbarui daftar paket dari repository agar sistem tahu versi terbaru dari software yang tersedia.

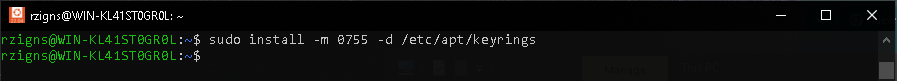


* *sudo apt install ca-certificates curl gnupg lsb-release -y* Menginstal dependensi penting yang dibutuhkan untuk mengunduh dan menambahkan repository pihak ketiga:
  + ca-certificates: validasi HTTPS
  + curl: unduh file dari internet
  + gnupg: verifikasi integritas file via GPG key
  + lsb-release: mendeteksi versi Ubuntu



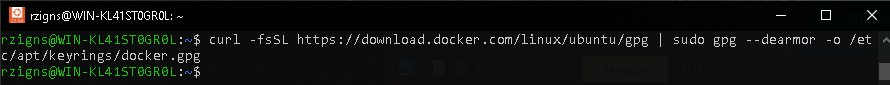
* *sudo install -m 0755 -d /etc/apt/keyrings*

Membuat direktori untuk menyimpan *GPG keyring* Docker, yang digunakan untuk memverifikasi bahwa file dari Docker benar dan aman.



* *curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg*

Download and store Docker’s official GPG key



* *echo \*

*"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] \ https://download.docker.com/linux/ubuntu \*

*$(lsb\_release -cs) stable" | \*

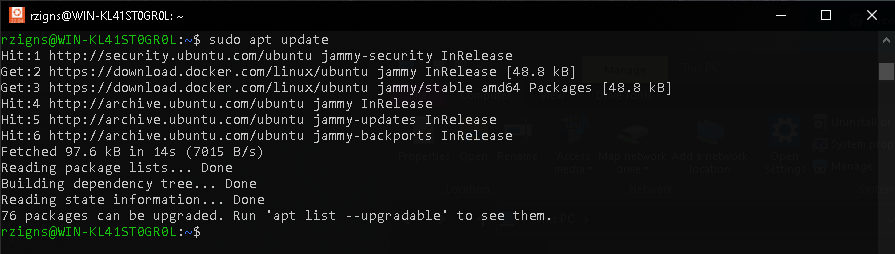
*sudo tee /etc/apt/sources.list.d/docker.list > /dev/null*

Set up Docker repository



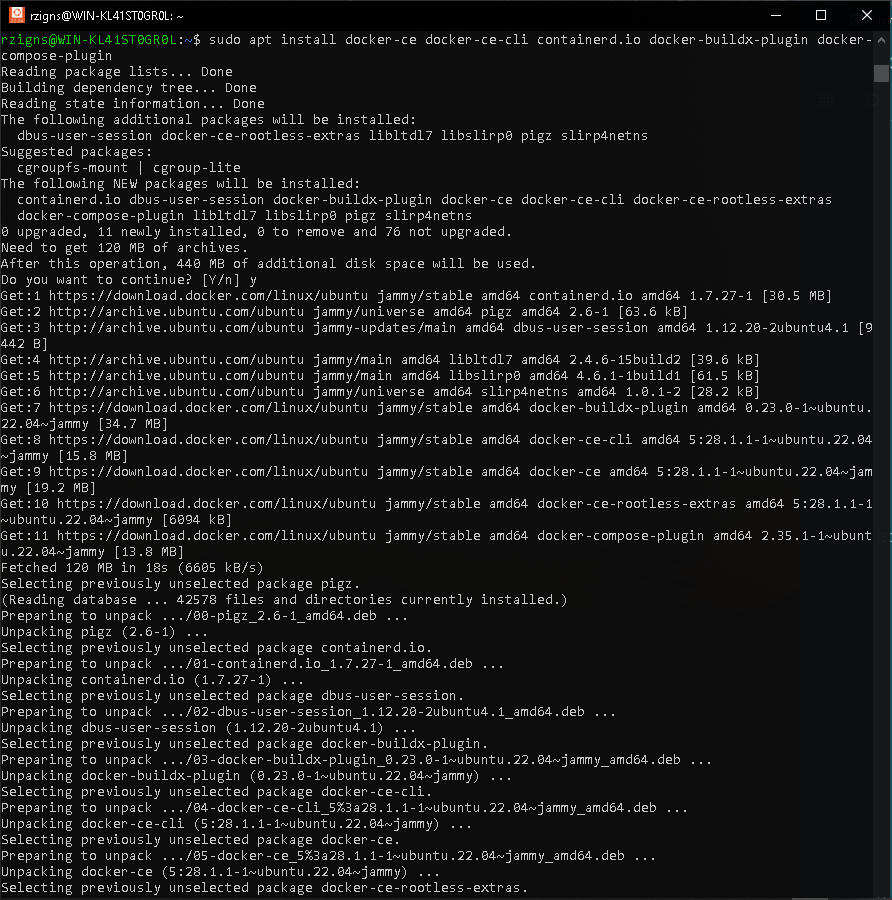
* *sudo apt update*

Update package index again (now includes Docker repo)



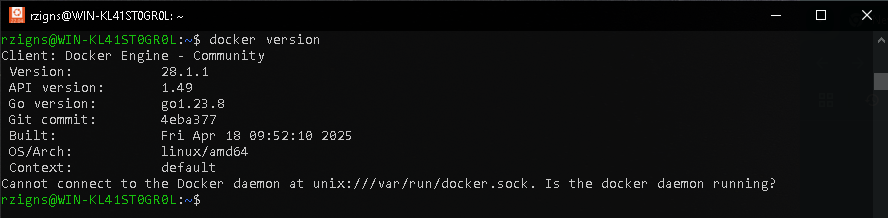
* *sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker- compose-plugin*

Install Docker Engine and related components

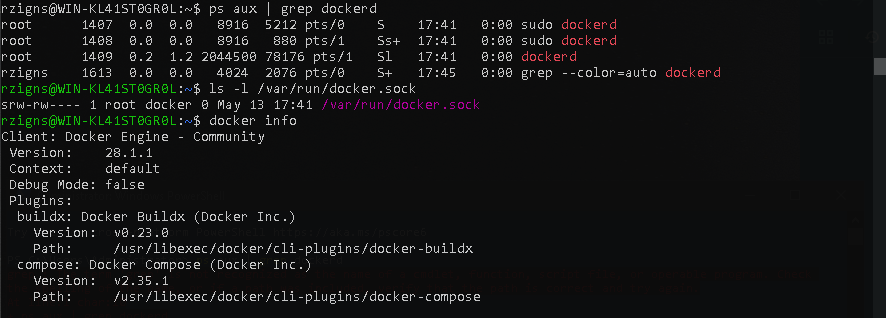


* *docker version*

Memeriksa apakah docker sudah terinstall dengan benar



* Perintah ini digunakan untuk memeriksa status layanan Docker jika tidak mengunakan systemcli



## Kesimpulan

Praktik instalasi dan konfigurasi Docker menggunakan WSL2 tanpa Docker Desktop menunjukkan bahwa Docker dapat dijalankan secara langsung di Ubuntu WSL2 dengan ringan dan efisien, tanpa ketergantungan pada GUI atau lisensi Docker Desktop. Meskipun WSL2 tidak mendukung systemd, sehingga perintah seperti systemctl tidak bisa digunakan, Docker daemon (dockerd) tetap dapat dijalankan secara manual dan digunakan oleh perintah docker. Setelah daemon aktif dan hak akses user dikonfigurasi (dengan menambahkan ke grup docker), Docker berfungsi dengan baik dan dapat menjalankan container seperti hello-world. Praktik ini cocok untuk lingkungan pengembangan ringan, tetapi memerlukan tambahan konfigurasi untuk otomatisasi daemon dan pengelolaan layanan karena keterbatasan sistem init di WSL2.