**LAPORAN PRAKTIKUM**

**SISTEM OPERASI**



Modul 6

1. Asfal Fuad (240306014)
2. Lalu Sulthan Nabiel Wira Pratama (240306016)
3. Yaumilfirkah Harfinah (240306012)

Program Studi Teknologi Informasi

Fakultas Dakwah dan Ilmu Komunikasi

Universitas Islam Negeri Mataram

2025

|  |  |
| --- | --- |
| Pertemuan | 6 |
| Topik | Virtualisasi & Monitoring |
| Repository | - |
| Tanggal | - |

1. Tujuan

Tujuan praktikum Virtualisasi & Monitoring:

* 1. Memahami konsep dasar virtualisasi.
  2. Menggunakan Virtual Machine di Linux.
  3. Mengimplementasikan teknologi virtualisasi di Linux.
  4. Mengevaluasi manfaat virtualisasi dalam pengembangan dan pengelolaan system.
  5. Meningkatkan efisiensi dan keamanan dalam pengembangan system.

1. Requirement

1.      Sistem Operasi yang digunakan           : Ubuntu LTS 24.0.1

2.      Browser                                                : Google Version129.0.6668.10

3.      Tools yang digunakan                             : Virtualbox

1. Dasar Teori
2. **Virtualisasi**

Virtualisasi adalah suatu metode menggabungkan dan saling berbagi sumber daya teknologi informasi yang bertujuan untuk mempermudah pengelolaan dan meningkatkan penggunaan aset sehingga sumberdaya teknologi informasi lebih maksimal digunakan serta dapat memenuhi permintaan bisnis, dengan virtualisasi suatu perangkat komputer server dapat digunakan berbagai aplikasi dan sistem operasi sehingga seolah-olah beroperasi dalam beberapa aset. Hal ini dapat meningkatkan utilitas aset dan mengurangi biaya kebutuhan aset fisik. Inti dari virtualisasi adalah membuat suatu simulasi dari perangkat keras, sistem operasi, jaringan dan lainya. virtualisasi digunakan sebagai sarana untuk improvisasi skalibilitas dari perangkat keras yang ada .

“Menurut Alan Murphy dalam papernya Virtualization Defined Eight Different Ways, menyebutkan setidaknya terdapat delapan istilah dalam penerapan virtualisasi. Diantaranya adalah virtualisasi sistem operasi, virtualisasi aplikasi server, virtualisasi aplikasi, manajemen virtualisasi, virtualisasi jaringan, virtualisasi perangkat keras, virtualisasi penyimpanan dan layanan virtualisasi.

1. **Manfaat Virtualisasi**

Virtualisasi memberikan banyak manfaat dibandingkan infrastruktur biasa. yaitu diantaranya dengan virtualisasi dapat mengurangi biaya fiskal dan kemudahan dalam pengelolaan dan pemasangan tambahan. layanan virtualisasi dapat memungkinkan guest tertentu dapat diduplikasi dengan mudah sehingga dapat digunakan dalam melakukan testing aplikasi yang akan dipasang meski dalam lingkungan operasi sistem yang berbeda.

**Berikut adalah beberapa manfaat menggunakan virtualisasi :**

* **Meningkatkan Efektifitas Komputer server**

Virtualalisasi memberikan manfaat meningkatkan keefektifan penggunaan sumberdaya komputer seperti RAM dan Prosesor. serta penghematan terhadap energi listrik dengan sedikit pembelian terhadap fisikal server. penerapan virtualisasi tersebut dengan menggunakan beberapa virtual machine dalam satu fisikal server /Host.

* **Memudahkan dalam Proses Backup dan Recovery.**

Setiap server yang dijalankan di dalam sebuah mesin virtual dapat disimpan dalam sebuah image yang berisi seluruh konfigurasi system. Jika suatu saat terjadi masalah pada server yang berjalan, maka tidak perlu memasang dan konfigurasi ulang, cukup dengan mengambil salinan dari image yang telah disimpan.

* **Memudahkan dalam Proses Pemasangan.**

Dengan virtualisasi server virtual dapat dikloning dan dapat dijalankan pada komputer lain dengan mengatur sedikit konfigurasi sehingga mengurangi beban kerja para staf teknologi informasi dan dapat mempercepat proses implementasi suatu system.

* **Memudahkan dalam Pengelolaan dan Pemeliharaan.**

Jumlah perangkat komputer yang lebih sedikit secara otomatis akan mengurangi waktu dan biaya dalam pengelolaan, serta memberikan kemudahan dalam proses pemeliharaan perangkat yang lebih sedikit.

* **Melaukan Standardisasi Perangkay Keras.**

Virtualisasi akan melakukan dan enkapsulasi perangkat keras sehingga proses pengenalan dan pemindahan suatu spesifikasi perangkat keras tertentu tidak menjadi suatu masalah karena akan secara otomatis terbaca oleh sistem.

1. **Tipe Virtualiasi**
2. **Virtualiasi Perangkat Keras.**

Virtualisasi perangkat keras adalah upaya menciptakan mesin virtual yang bekerja layaknya sebuah komputer beserta dengan sistem operasi. dalam virtualisasi perangkat keras, mesin host adalah mesin tempat virtualisasi itu berada, dan mesin guest itu sendiri.

Ada beberapa jenis virtualisasi perangkat keras yaitu:

* **Virtualisasi penuh** yaitu virtualisasi yang hampir menyerupai mesin asli dan mampu menjalankan sistem operasi tanpa perlu diubah .
* **Virtualisasi sebagian** yaitu virtualisasi di mana tidak semua aspek lingkungan disimulasikan dan tidak semua perangkat lunak dapat langsung berjalan, beberapa perlu disesuaikan agar dapat berjalan dalam lingkungan virtual tersebut.
* **Para virtualisasi** yaitu virtualisasi perangkat tidak disimulasikan tetapi perangkat lunak guest berjalan dalam domain nya sendiri seolah-olah dalam sistem yang berbeda. Dalam hal ini perangkat lunak tamu perlu disesuaikan untuk dapat berjalan

1. **Virtualisasi Perangkat Lunak**

Virtualisasi perangkat lunak adalah virtualisasi yang memungkinkan satu komputer host untuk membuat dan menjalankan satu atau lebih lingkungan virtual. virtualisasi perangkat lunak banyak digunakan untuk mensimulasikan sebuah sistem komputer lengkap, dengan tujuan untuk memungkinkan sistem operasi guest berjalan.

1. **Virtualisasi Kontainer**

Di dalam konteks virtualisasi Kontainer dapat diartikan sebagai alat yang dapat dipergunakan untuk system yang terisolasi yang terletak pada level operasi sistem yang yang dijalankan pada satu induk kernel atau Host.

Terdapat 2 jenis Kontainer yang umum digunakan yaitu :

1. Kontainer berbasis sistem operasi adalah suatu kontainer yang memberikan isolasi pada level sistem operasi dan memanfaatkan kernel yang sama dari suatu induk. contohnya: LXC, openVZ dan lainya.
2. Kontainer berbasis aplikasi adalah suatu kontainer yang memberikan isolasi pada level aplikasi dengan memanfaatkan beberapa komponen yang ada pada sistem operasi induk yang ditambah dengan beberapa komponen pada kontainer lain yang menjadi dasar dari berjalannya suatu aplikasi. Contohya: Docker,Rocket.
3. **Virtual Machine dan Kontainer**

Virtual machine adalah perangkat lunak yang dapat mengisolasi sebuah mesin komputer serta dapat menjalankan semua program seperti komputer aslinya atau duplikat dari komputer asli, sedangkan kontainer adalah suatu teknologi virtualisasi yang dapat mengisolasi sebuah proses dari proses yang lain nya yang akan mengisolasi library dan aplikasi yang digunakan saja tanpa mengisolasi seluruh komponen seperti perangkat keras kernel, serta sistem operasi.

Dalam teknologi virtualisasi virtual machine dan docker sama-sama memiliki konsep skema virtualisasi, akan tetapi ada yang membedakan antara virtual machine dan Kontainer, berikut adalah beberapa perbedaan mendasar antara virtual machine dan kontainer:

* Virtual machine menggunakan kernel tersendiri sehingga membuat beban pada Host menjadi lebih lebih berat sedangkan Kontainer membagi kernel nya kedalam kedalam kontainer yang sudah ada sehingga lebih efektif digunakan.
* Virtual machine menggunakan keseluruhan sumberdaya perangkat keras yang ada pada host sehingga host tersebut menjalankan operasi sistem secara ganda bersamaan, sedangkan kontainer bersifat seperti aplikasi dan hanya sedikit menggunakan sumberdaya pada host
* Virtual machine tidak dapat mengaokasikan spesifikasi pada antar virtual machine sedangkan docker dapat mengalokasikan spesifikasi antar kontainer sehingga dapat melakukan efesiensi sumber daya dengan sebaik-

baiknya pada sistem.

1. **Virtualbox**

**VirtualBox adalah alat untuk memvirtualisasikan arsitektur komputasi x86 dan AMD64/Intel64, yang memungkinkan pengguna untuk menggunakan**[**desktop, server, dan sistem operasi**](https://www.spiceworks.com/tech/it-strategy/articles/what-is-it-infrastructure/)**sebagai mesin virtual. Anda dapat menggunakan solusi ini untuk menggunakan mesin virtual sebanyak mungkin sesuai dengan sumber daya yang dimiliki arsitektur host.**

Pengguna dapat memasang alat virtualisasi lintas platform ini di komputer mana pun yang menggunakan prosesor Intel atau AMD. Alat ini berjalan di semua sistem operasi utama, termasuk Windows, Linux, Oracle Solaris, dan macOS.

VirtualBox dapat memperluas kemampuan teknis komputer yang kompatibel, sehingga memungkinkannya menjalankan beberapa sistem operasi di beberapa mesin virtual sekaligus. Misalnya, pengguna dapat menjalankan Windows 11 dan Ubuntu di MacBook Air mereka tanpa mengorbankan konfigurasi sistem atau aplikasi yang ada.

VirtualBox memungkinkan sistem apa pun untuk memasang dan mengoperasikan mesin virtual sebanyak yang diizinkan oleh memori dan ruang disk. Bergantung pada konfigurasi sistem host, seseorang dapat menggunakan solusi ini untuk menyebarkan apa pun mulai dari mesin kelas desktop dan sistem tertanam kecil hingga lingkungan cloud dan penyebaran pusat data besar.

VirtualBox dimanfaatkan oleh administrator TI, pengembang perangkat lunak, sejumlah [profesional teknis lainnya](https://www.spiceworks.com/tech/cloud/articles/cloud-computing-careers-job-roles/" \o "profesional teknis lainnya) , dan pengguna akhir umum untuk berbagai keperluan. Misalnya, pengembang dapat membuat perangkat lunak untuk berbagai platform pada mesin yang sama, sementara penguji dapat menggunakannya untuk membangun jaringan virtual untuk pengujian perangkat lunak.

VirtualBox dapat diunduh dan digunakan secara **gratis oleh pengguna pribadi**di bawah Lisensi Publik Umum GNU. Namun, pengguna komersial harus membeli lisensi komersial **VirtualBox Extension Pack Enterprise**untuk penggunaan bisnis. Satu soket VirtualBox Enterprise permanen berharga $1.220, sementara 100 unit soket pengguna plus tak terbatas dan bernama harganya $6.100. Oracle juga menawarkan lisensi Enterprise tahunan seharga $50 per pengguna.

Lisensi Enterprise dilengkapi dengan manajemen Extension Pack yang lebih baik, paket fitur milik Oracle. Extension Pack gratis untuk penggunaan pribadi; namun, pengguna harus membeli lisensi untuk penggunaan komersial. Extension Pack membuka dukungan protokol desktop jarak jauh (RDP), dukungan perangkat USB 2.0 dan USB 3.0, enkripsi disk, serta boot NVMe dan PXE untuk sistem Intel.

Terakhir, pengguna dapat memanfaatkan Guest Additions untuk memperluas fungsionalitas VirtualBox lebih jauh. Modul ini menambahkan dukungan video yang ditingkatkan, integrasi penunjuk tetikus, folder bersama host/tamu dan saluran komunikasi generik, sinkronisasi waktu, jendela yang lancar, logon otomatis, dan clipboard bersama.

1. **Operasi**

VirtualBox bekerja dengan memperluas kemampuan sistem operasi yang ada, sehingga memungkinkannya menjalankan mesin virtual tanpa perubahan apa pun pada konfigurasi perangkat keras atau perangkat lunak. Sistem operasi tempat VirtualBox diinstal disebut host, sedangkan sistem operasi yang berjalan di dalam VirtualBox dikenal sebagai guest. Setelah sistem operasi guest diterapkan dan dikonfigurasi dengan benar, tidak ada batasan yang akan membedakannya dari mesin untuk pengguna akhir biasa.

Sederhananya, guest adalah sistem operasi yang berjalan sebagai mesin virtual. Guest adalah 'sistem dalam sistem' dan dapat terdiri dari sistem operasi yang terkenal atau khusus. Namun, pengguna memerlukan salinan berlisensi terpisah untuk menjalankan mesin virtual secara legal untuk sistem operasi yang memerlukan lisensi, seperti Windows dan macOS.

Host VirtualBox dapat menampung sebanyak mungkin sistem tamu sesuai dengan kemampuan perangkat kerasnya tanpa mengorbankan sumber daya yang dibutuhkan untuk menjalankan host tersebut. Instans tamu individual dapat diluncurkan, dihentikan, dan dijeda menggunakan GUI VirtualBox atau alat baris perintah. Opsi baris perintah memungkinkan VM untuk disebarkan di server tanpa dukungan GUI, menjadikan VirtualBox solusi yang sangat fleksibel. Mesin virtual ini dapat [terhubung ke jaringan](https://www.spiceworks.com/tech/networking/articles/what-is-wide-area-network/" \o "terhubung ke jaringan) jika dikonfigurasi dengan benar dan beroperasi seperti mesin biasa.

Penerapan VirtualBox pada umumnya mencakup komponen-komponen berikut:

* Sebuah hypervisor untuk sistem host.
* Kit pengembangan perangkat lunak (SDK) dan antarmuka pemrograman aplikasi (API) untuk mengelola operasi VM tamu.
* Alat baris perintah untuk manajemen tamu lokal.
* Layanan web yang memungkinkan pengguna untuk mengelola tamu dari jarak jauh.
* Alat bergaya wizard untuk manajemen tamu.
* Konsol yang menampilkan aplikasi tamu di host.
* Server protokol desktop jarak jauh (RDP) memungkinkan pengguna untuk mengakses tamu menggunakan klien jarak jauh.

Administrator dapat menerapkan sistem tamu menggunakan citra ISO atau citra VHD/VDI/VMDK pada platform host VirtualBox. Menerapkan sistem tamu menggunakan citra ISO memungkinkan sistem operasi tamu diinstal secara normal tetapi sebagai VM. Sebaliknya, citra VHD/VDI/VMDK akan memungkinkan administrator untuk menerapkan peralatan virtual dengan cepat tanpa menginstal sistem operasi pada sistem tamu. TurnKey Linux merupakan sumber daya yang berharga untuk menemukan peralatan virtual untuk VirtualBox.

VirtualBox bersifat modular, yang memungkinkannya menyediakan serangkaian fitur serupa tanpa memandang platform host. Selain itu, mesin virtual dan citra disk yang dihasilkan pada host mana pun akan kompatibel dengan host lain yang didukung. Selain itu, setiap pengguna yang belajar mengelola mesin tamu pada host tertentu akan dapat mengelola tamu pada sistem host yang berbeda juga.

Terakhir, beberapa sistem memungkinkan pengguna untuk mengaktifkan ekstensi virtualisasi perangkat keras AMD-V atau Intel VT-x di tingkat BIOS. Penerapan VirtualBox pada sistem tersebut dapat menghasilkan pengoperasian tamu yang lebih baik.

1. **KVM (Kernel-based Virtual Machine)**

Mesin Virtual Berbasis Kernel (KVM) adalah fitur perangkat lunak yang dapat Anda instal di mesin fisik Linux untuk membuat mesin virtual. Mesin virtual adalah aplikasi perangkat lunak yang bertindak sebagai komputer independen di dalam komputer fisik lainnya. Mesin virtual berbagi sumber daya, seperti siklus CPU, *bandwidth* jaringan, dan memori dengan mesin fisik. KVM adalah komponen sistem operasi Linux yang memberikan dukungan *native*untuk mesin virtual di Linux. KVM telah tersedia di distribusi Linux sejak 2007.

1. **Manfaat KVM**

Mesin Virtual Berbasis Kernel (KVM) dapat mengubah mesin Linux apa pun menjadi hypervisor bare-metal. Hal ini memungkinkan developer untuk menskalakan infrastruktur komputasi untuk berbagai sistem operasi tanpa berinvestasi dalam perangkat keras baru. KVM membebaskan administrator server dari penyediaan infrastruktur virtualisasi secara manual dan memungkinkan sejumlah besar mesin virtual untuk digunakan dengan mudah di lingkungan cloud.

Bisnis menggunakan KVM karena memiliki keuntungan sebagai berikut.

* **Performa tinggi**

KVM direkayasa untuk dengan lancar mengelola aplikasi dengan permintaan yang tinggi. Semua sistem operasi guest mewarisi performa tinggi dari sistem operasi host, yaitu Linux. Hypervisor KVM juga memungkinkan virtualisasi dilakukan sedekat mungkin dengan perangkat keras server, yang selanjutnya mengurangi latensi proses.

* **Keamanan**

Mesin virtual yang berjalan di KVM menikmati fitur keamanan asli pada sistem operasi Linux, termasuk Security-Enhanced Linux (SELinux). Hal ini memastikan bahwa semua lingkungan virtual secara ketat mematuhi batas keamanan masing-masing untuk memperkuat privasi dan tata kelola data.

* **Stabilitas**

KVM telah banyak digunakan dalam aplikasi bisnis selama lebih dari satu dekade. KVM menikmati dukungan yang sangat baik dari komunitas sumber terbuka yang berkembang. Kode sumber yang mendukung KVM sudah matang dan menyediakan fondasi yang stabil untuk aplikasi korporasi.

* **Efisiensi biaya**

KVM gratis dan sumber terbuka, yang berarti bisnis tidak perlu membayar biaya lisensi tambahan untuk meng-hosting mesin virtual.

* **Fleksibilitas**

KVM menyediakan banyak opsi bagi bisnis selama penginstalan karena KVM bekerja dengan berbagai penyiapan perangkat keras. Administrator server dapat mengalokasikan CPU, penyimpanan, atau memori tambahan secara efisien ke mesin virtual dengan KVM. KVM juga mendukung penyediaan yang sedikit jumlahnya, yang hanya menyediakan sumber daya ke mesin virtual bila diperlukan.

1. **Fungsi KVM**

### **Menyediakan Virtualisasi pada Sistem Linux**

KVM adalah modul di kernel Linux yang mengubah sistem operasi Linux menjadi hypervisor. Dengan KVM, kamu bisa menjalankan beberapa mesin virtual (VM) secara bersamaan di satu perangkat keras fisik.

### **Menjalankan Banyak Sistem Operasi Secara Bersamaan**

KVM memungkinkan kamu menjalankan berbagai sistem operasi (Windows, Linux, BSD, dll) sebagai mesin virtual di atas Linux tanpa perlu perangkat keras tambahan.

### **Memaksimalkan Pemanfaatan Hardware**

Dengan KVM, sumber daya komputer seperti CPU, memori, dan storage bisa dibagi dan digunakan secara efisien oleh beberapa VM yang berjalan secara paralel.

### **Mendukung Virtualisasi Berbasis Hardware**

KVM menggunakan fitur virtualisasi hardware dari prosesor modern (Intel VT-x dan AMD-V) untuk meningkatkan performa mesin virtual.

### **Mendukung Isolasi dan Keamanan**

Setiap mesin virtual berjalan terisolasi sehingga masalah pada satu VM tidak memengaruhi VM lainnya atau host Linux.

1. **Proxmox VE**

Proxmox Virtual Environment (VE) merupakan salah satu platform virtualisasi open-sourceberbasis Linux Debian, baik hardware maupun operating system, untuk mengelola dua teknologi virtualisasi. Platform virtualisasi ini kerap disebut-sebut sebagai alternatif selain VMware dan cocok ditujukan untuk development tmaupun production. Proxmox VE ini juga akan melakukan manajemen container, virtual machine, strorage, jaringan virtual,high availability clustermelalui antar-mukaweb, dan juga bisa menggunakan command line

1. **Fungsi Proxmox VE**

Proxmox VE memiliki fungsi khusus sebagai virtualisasi appliance maupun operating system. Berikut beberapa teknologi virtualisasi yang dapat digunakan pada Proxmox VE.

### **Kernel Virtual Machine (KVM)**

Merupakan Kernel-basedvirtual machineyang ditambahkan pada Linux untuk membuat full virtualisasi. KVM ini juga merupakan bagian integral dari Linux sejak 2007.

### **Linux Container (LXC)**

Kerap disebut sebagai virtualisas ioperating system(OS), LXC adalah virtualisasi yang menggunakan container. Virtualisasi ini memiliki tingkat efisiensi yang tinggi dan juga kecepatan aksesnya menjadikan LXC berkembang dengan cepat.

### **QEMU**

QEMU menyediakan emulasi dan virtualisasi interface.

1. **Wine (Wine Is Not Emulator)**

**WINE** adalah aplikasi perangkat lunak yang bertujuan untuk memungkinkan komputer program yang ditulis untuk Microsoft Windows untuk berjalan di Unix-seperti sistem operasi . Wine juga menyediakan perpustakaan perangkat lunak , yang dikenal sebagai **Winelib,** terhadap yang pengembang dapat mengkompilasi aplikasi Windows untuk membantu pelabuhan mereka untuk sistem Unix-like.

Wine bukanlah emulator , tetapi adalah lapisan kompatibilitas, menyediakan implementasi alternatif dari DLL bahwa Windows program memanggil [rujukan?], dan proses untuk menggantikan Windows NT kernel. Wine ini sebagian besar ditulis dengan menggunakan Black-kotak pengujian reverse-engineering, untuk menghindari hak cipta masalah.

Wine Nama awalnya singkatan untuk mulator **E WIN** dows. Maknanya kemudian bergeser ke rekursif backronym, **W** s **U** ine **aku** ot suatu mulator **E.**  Sementara nama kadang-kadang muncul dalam bentuk WINE dan anggur, pengembang proyek telah sepakat untuk standarisasi pada Wine bentuk.

Dalam sebuah survei tahun 2007 oleh desktoplinux.com dari 38.500 pengguna desktop Linux, 31,5% dari responden melaporkan menggunakan Wine untuk menjalankan aplikasi Windows. Pluralitas ini lebih besar daripada semua virtualisasi x86 program gabungan, serta lebih besar dari 27,8% yang melaporkan tidak menjalankan aplikasi Windows.

1. **Perbedaan Docker dan Virtual Machine(VM).**

| **Aspek** | **Docker (Container)** | **Virtual Machine (VM)** |
| --- | --- | --- |
| **Definisi** | Platform untuk membuat dan menjalankan kontainer ringan yang berbagi kernel OS host. | Emulasi lengkap dari sistem komputer, termasuk OS dan perangkat keras virtual. |
| **Arsitektur** | Berjalan di atas OS host dan menggunakan kernel bersama. | Berjalan di atas hypervisor dan menjalankan OS sendiri. |
| **Ukuran** | Ringan (beberapa MB) | Berat (bisa beberapa GB per VM) |
| **Boot Time** | Sangat cepat (dalam hitungan detik) | Lebih lama (menit) |
| **Isolasi** | Isolasi level proses dan filesystem | Isolasi penuh antar OS (sandboxing kuat) |
| **Performa** | Lebih cepat dan efisien karena tidak perlu OS lengkap | Lebih berat, tergantung spesifikasi perangkat keras |
| **Penggunaan Sumber Daya** | Lebih hemat CPU dan RAM | Konsumsi tinggi karena OS lengkap untuk tiap VM |
| **Portabilitas** | Sangat tinggi, mudah dipindahkan dan dijalankan di mana saja | Terbatas, memerlukan kompatibilitas hypervisor |
| **Sistem Operasi Guest** | Menggunakan image berbasis OS namun berbagi kernel host | Menggunakan OS lengkap (Windows, Linux, dll) |
| **Contoh Teknologi** | Docker, Podman, LXC | VMware, VirtualBox, KVM, Hyper-V |
| **Keamanan** | Cukup baik, namun masih berbagi kernel dengan host | Lebih aman karena isolasi total |
| **Cocok untuk** | Microservices, CI/CD, cloud-native apps | Aplikasi legacy, multi-OS, dan sistem yang butuh isolasi penuh |
| **Snapshot & Rollback** | Tersedia, tapi lebih kompleks dari VM | Sangat mudah menggunakan fitur snapshot |

1. **Docker**

Docker adalah sebuah platform terbuka bagi para pengembang dan sistem administrator untuk membangun, menyebarkan, dan menjalankan aplikasi terdistribusi baik pada perangkat pribadi seperti laptop ataupun pada virtual data center atau cloud data senter.

Docker kontainer berjalan pada sebuah mesin tunggal yang berbagi pakai sistem operasi kernel yang sama, dan dapat dimulai secara langsung serta menggunakan lebih sedikit RAM. Image (sekumpulan berkas dan folder) dibangun dari filesystem berlapis dan berbagi berkas umum, membuat penggunaan penyimpanan dan image download jauh lebih efisien.

# **komponen-komponen Docker**

* + - * **Docker image** adalah dasar template untuk docker kontainer, sebuah images yang berisi operasi sistem maupun aplikasi yang akan diinstall dan telah jadi. image ini digunakan untuk menjalankan kontainer, docker index terdapat banyak image yang dapat di pilih dan dapat digunakan sebagai base image.
      * **Docker Kontainer** adalah sebuah image yang dapat dikemas dan dibaca tulis, kontainer berjalan di atas nya. pada setiap perubahan yang dapat disimpan pada kontainer akan menyebabkan terbentuknya layer baru diatas base image.
      * **Docker registry** adalah sebuah repositori distribusi kumpulan docker image yang terpusat baik bersifat public dan private repositori. registry publik disebut dengan dockerhub, didocker hub dapat dilakukan push dan pull image sendiri.
      * **Docker file** adalah skrip yang berisi atau terdiri dari rangkaian perintah yang dieksekusi secara otomatis dan berurutan yang bertujuan membangun sebuah image. Dengan menggunakan perintah docker build dari terminal, maka kan terlihat docker membangun image secara bertahap berdasarkan eksekusi perintah dalam skrip.
      * **Docker Repository** yaitu tempat penyimpanan docker image yang telah dibuat oleh pengembang,

1. **Keunggulan Docker sebagai PlatformKontainer**

Sistem Kontainer merupakan solusi terhadap masalah bagaimana untuk menjalankan perangkat lunak secara handal ketika pindah dari suatu lingkungan komputasi ke lingkungan yang lain. sehingga dapat berpindah dari pengembang untuk lingkungan tes, dari lingkungan rilis kedalam produksi dan mungkin dari perangkat fisik di data senter untuk mesin virtual atau public cloud.

Berikut adalah beberapa keunggulan platform docker:

# **Dapat Melakukan Pengujian dan Distribusi Secara Terus Menerus.**

Docker mempunyai kemampuan dalam menghadirkan konsistensi diseluruh lingkungan sistem operasi. selalu ada perbedaan kecil antara lingkungan pengembangan dan aplikasi yang telah rilis. Kecuali jika anda memiliki lingkungan repositori sendiri sendiri dengan pemeriksaan yang lebih ketat. Perbedaan perbedaan karena versi paket yang berbeda atau dependensi tidak menjadi masalah, docker dapat mengatasi kesenjangan tersebut dengan memastikan lingkungan yang konsisten dari pembangunan untuk produksi. kontainer dikonfigurasi untuk memelihara semua konfigurasi dan dependensi internal, yang bertujuan agar anda dapat menggunakan kontainer yang sama pada tahap pengembangan hingga produksi perangkat lunak dan memastikan tidak ada perbedaan dan intervensi manual. dengan menggunakan docker kontainer anda juga dapat memastikan bahwa pengembang tidak perlu linkungan produksi yang mengatur secara identik. sebaliknya pengembang dapat menggunakan sistem mereka sendiri untuk menjalankan docker kontainer pada virtualbox.

* + - * **Docker Dapat Menjadi PlatfomMulti-Cloud.**

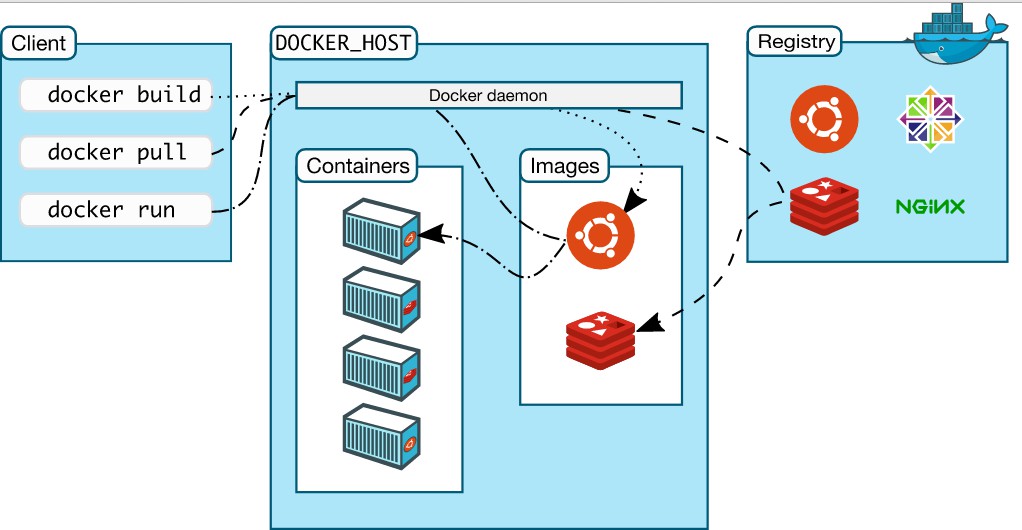
Manfaat docker kontainer lainya adalah portabilitas, Docker kontainer dapat dijalankan didalam Amazon EC2, Google compute engine, Server Rackspace atau virtualbox asalkan host mendukung platform docker. selain AWS dan GCP docker dapat bekerja sangat dengan berbegai penyedia IaaS lain seperti Microsoft Azure dan Openstack dan dapat digunakan berbagai manajemen konfigurasi seperti chef, bunnet, Ansible.

# **Standarisasi Lingkungan dan Kontrol Versi.**

Kontainer docker memastikan konsistensi di beberapa siklus pengembangan, rilis dan standardisasi lingkungan. docker kontainer bekerja seperti repositori GIT yang dapat melakukan perubahan Image docker dan dapat mengendalikan versi.

# **Docker Dapat Melakukan Isolasi.**

Docker memastikan pada setiap kontainer memiliki sumberdaya sendiri yang terisolasi dengan kontainer lainnya yang berada dalam satu lingkungan, sehingga dapat memiliki dapat memiliki berbagai kontainer untuk aplikasi yang terpisah yang berjalan sepeuhnya pada tumpukan yang sama.

1. **Arsitektur Docker**

Gambar Arsitektur Docker

Docker menggunakan arsitektur berbasis klien server. Dalam hal ini, docker klien mengirimkan permintaan berupa sebuah perintah kepada docker daemon untuk membangun, mendistribusikan, dan menjalankan Docker Kontainer. Docker klien dan server berkomunikasi via socket menggunakan RestfulSPI.

Terlihat pada gambar, Docker daemon berjalan pada host, sehingga pengguna tidak dapat berinteraksi langsung. Untuk mengaksesnya, menggunakan docker klien yang merupakan tampilan utama untuk pengguna, sehingga pengguna dapat berkomunikasi dengan docker daemon.

1. **Virtualenv (Virtual Environment)**

**Virtual Environment** (Lingkungan Virtual) dalam Python adalah lingkungan terisolasi yang berisi salinan interpreter Python dan pustaka (library) yang terpisah dari instalasi Python sistem. Dengan menggunakan virtual environment, setiap proyek Python dapat memiliki dependensi dan konfigurasi yang tidak saling memengaruhi.

1. **Jenis jenis alat Virtual Environment.**

* **venv**

venv adalah modul bawaan Python sejak versi 3.3. Ini adalah cara paling standar dan direkomendasikan untuk membuat virtual environment saat menggunakan Python 3.  
Kelebihannya adalah tidak perlu instalasi tambahan dan mudah digunakan.

* **virtualenv**

virtualenv adalah alat eksternal yang tersedia melalui pip. Ia digunakan sebelum venv tersedia, dan masih populer karena memiliki lebih banyak fitur, seperti kompatibilitas dengan Python 2, performa lebih baik, serta dukungan pembuatan lingkungan yang lebih fleksibel.

* **conda**

conda adalah manajer paket dan lingkungan virtual yang sangat populer di bidang data science. conda mendukung tidak hanya pustaka Python, tetapi juga pustaka dari bahasa lain seperti R dan C. conda cocok digunakan untuk proyek ilmiah atau analisis data yang memerlukan pustaka lintas bahasa.

* **pipenv**

pipenv merupakan alat yang menggabungkan manajemen virtual environment dengan manajemen dependensi menggunakan file Pipfile dan Pipfile.lock. Alat ini dirancang untuk menggantikan penggunaan requirements.txt, namun penggunaannya kini mulai menurun karena kurang aktif dikembangkan.

* **poetry**

poetry adalah manajer proyek Python modern yang mengelola virtual environment, dependensi, dan distribusi paket sekaligus. poetry semakin populer karena memudahkan pembuatan pustaka Python yang siap dipublikasikan ke PyPI, dengan manajemen versi dan struktur proyek yang rapi.

1. **Fungsi Virtual Environment**

* **Memisahkan Proyek**

Virtualenv membuat lingkungan terpisah untuk setiap proyek Python, sehingga setiap proyek bisa menggunakan versi library yang berbeda tanpa saling mengganggu.

* **Mencegah Konflik Versi**

Dengan virtualenv, kamu bisa menginstal versi pustaka tertentu hanya untuk satu proyek. Ini mencegah konflik antara proyek yang membutuhkan versi pustaka yang berbeda.

* **Menjaga Sistem Tetap Bersih**

Virtualenv menyimpan semua pustaka dalam folder proyek, sehingga tidak mengubah atau merusak sistem Python utama di komputer.

* **Memudahkan Kerja Tim**

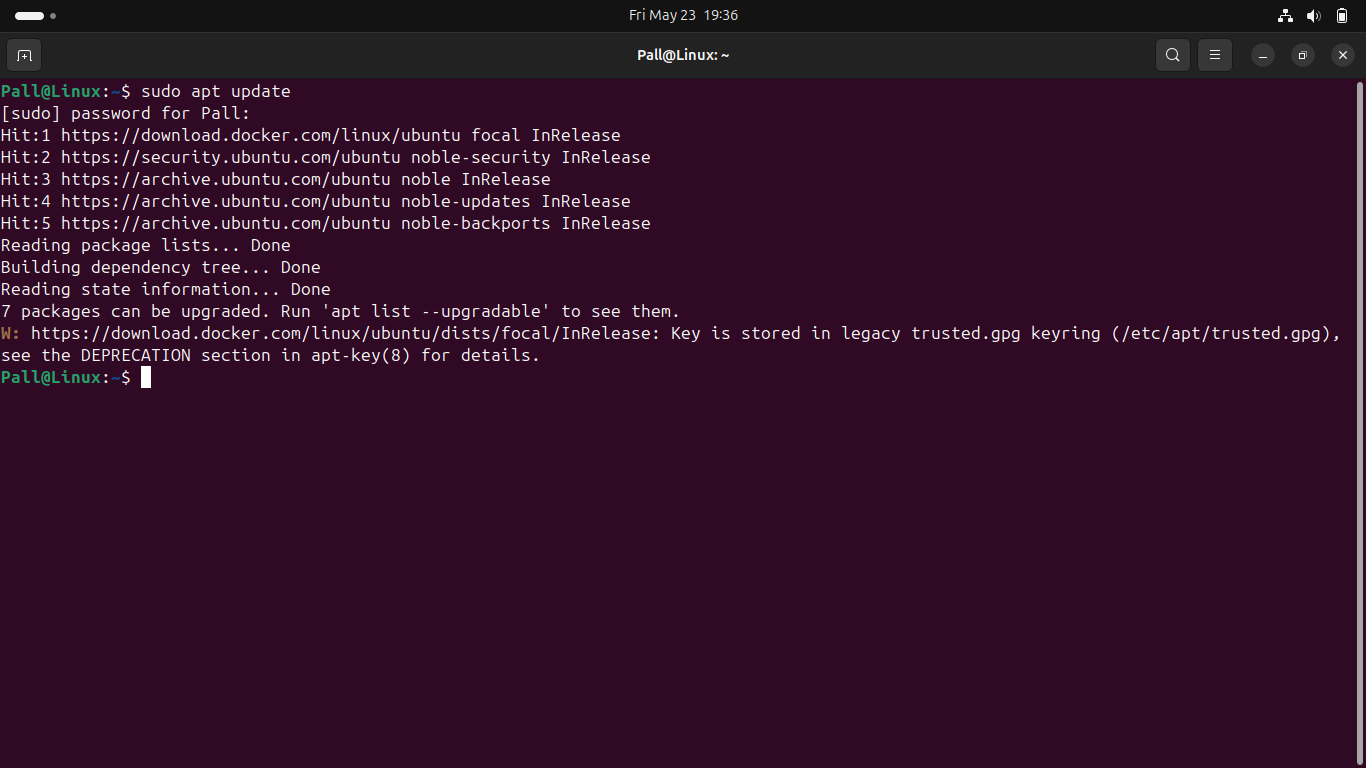
Virtualenv memungkinkan semua anggota tim menggunakan pustaka yang sama dengan cara membagikan file requirements.txt, sehingga hasil pengembangan lebih konsisten.

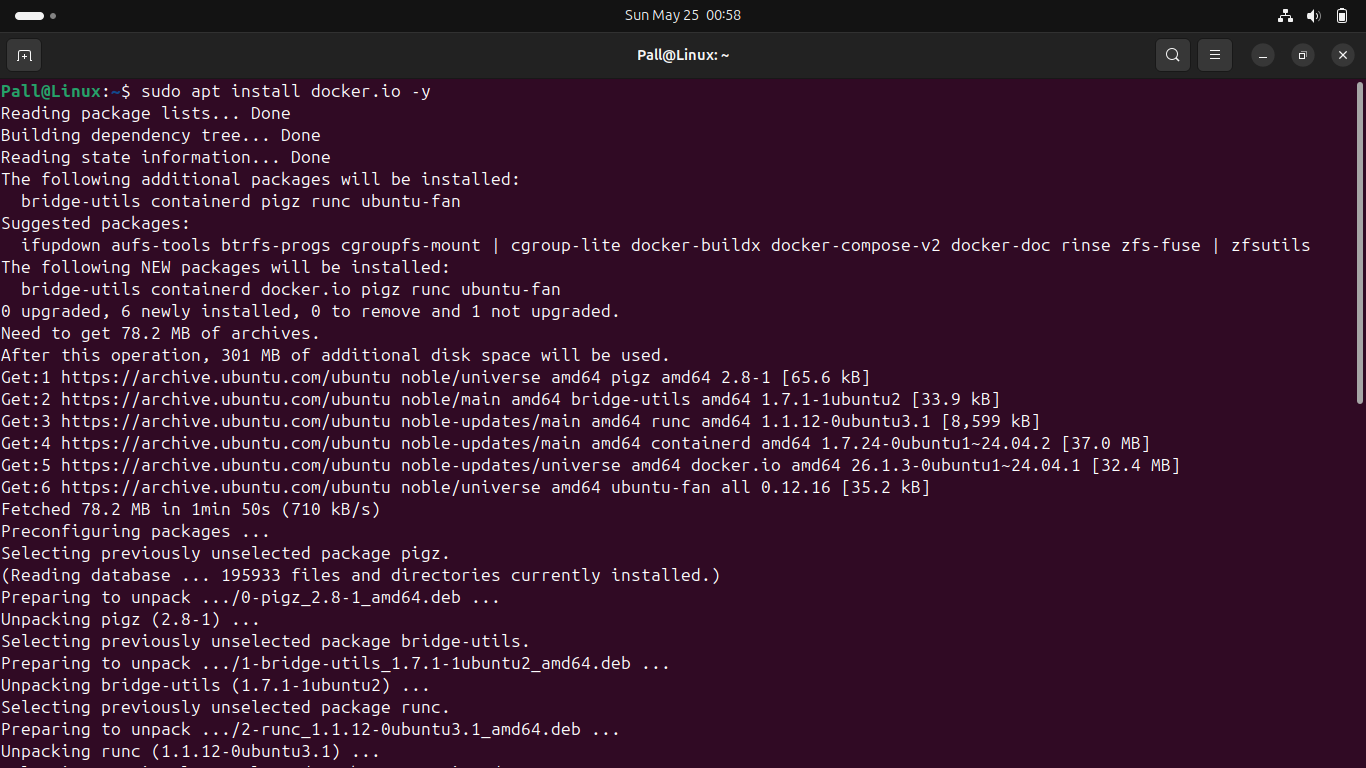
* **Mendukung Pengujian Percobaan**

Kamu bisa mencoba pustaka baru atau versi Python tertentu dalam virtualenv tanpa merusak proyek yang sudah ada atau sistem utama.

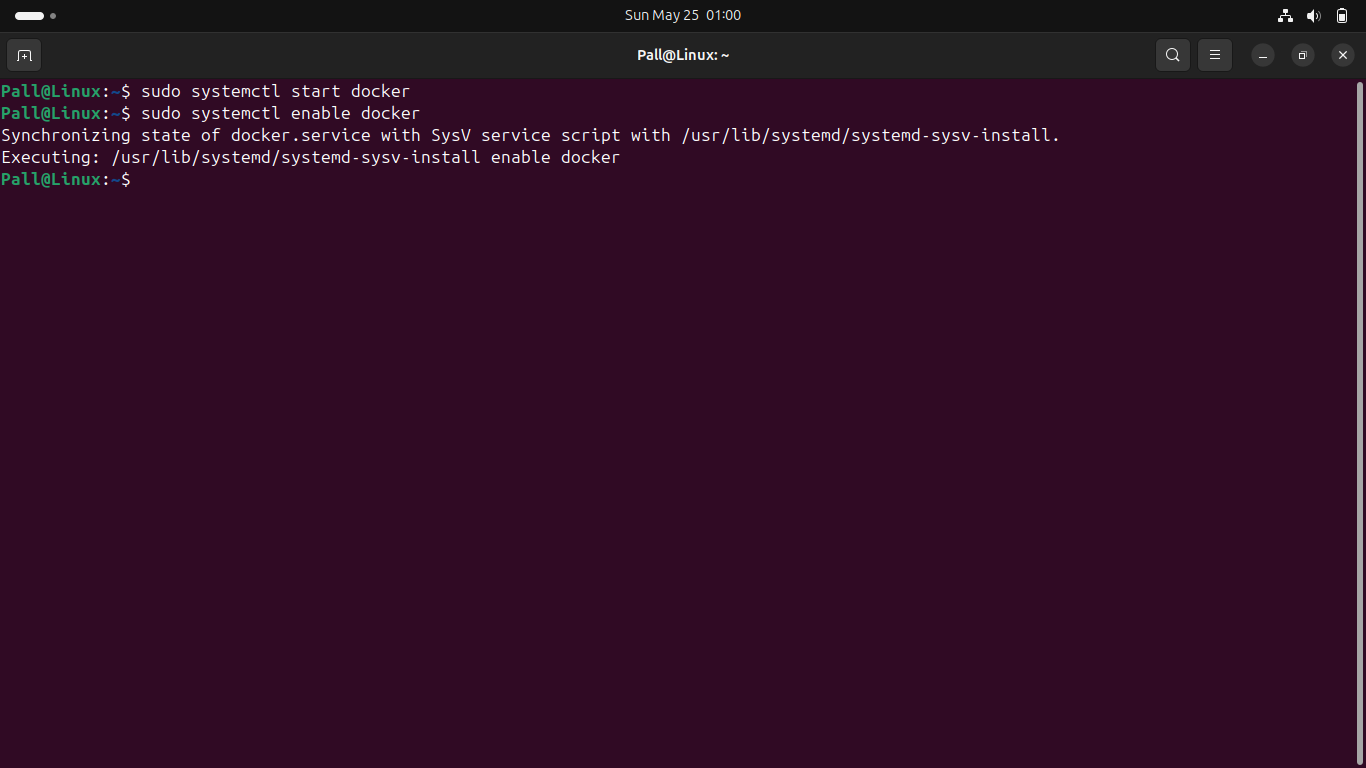
1. Implementasi
2. Instalasi Docker pada system operasi Linux
3. Perbarui daftar paket saat ini

* Masuk ke terminal ketik sudo apt update.



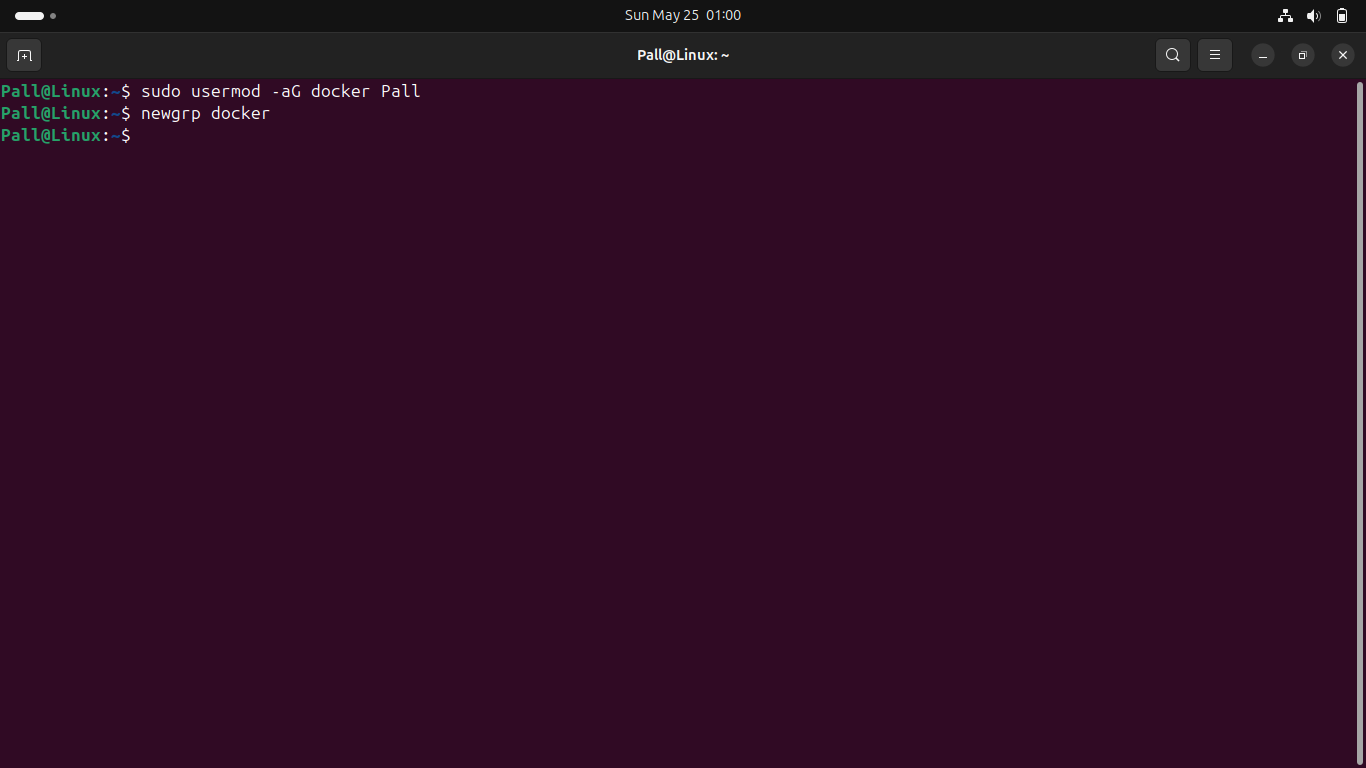
1. Instal docker versi komunitas dari repository default ubuntu dengan prompt sudo apt install docker.io – y.
2. Memulai layanan docker dan Mengatur agar layanan docker secara otomatis di jalankan saat booting dengan menggunakan prompt:

* sudo systemctl start docker
* sudo systemctl enable docker

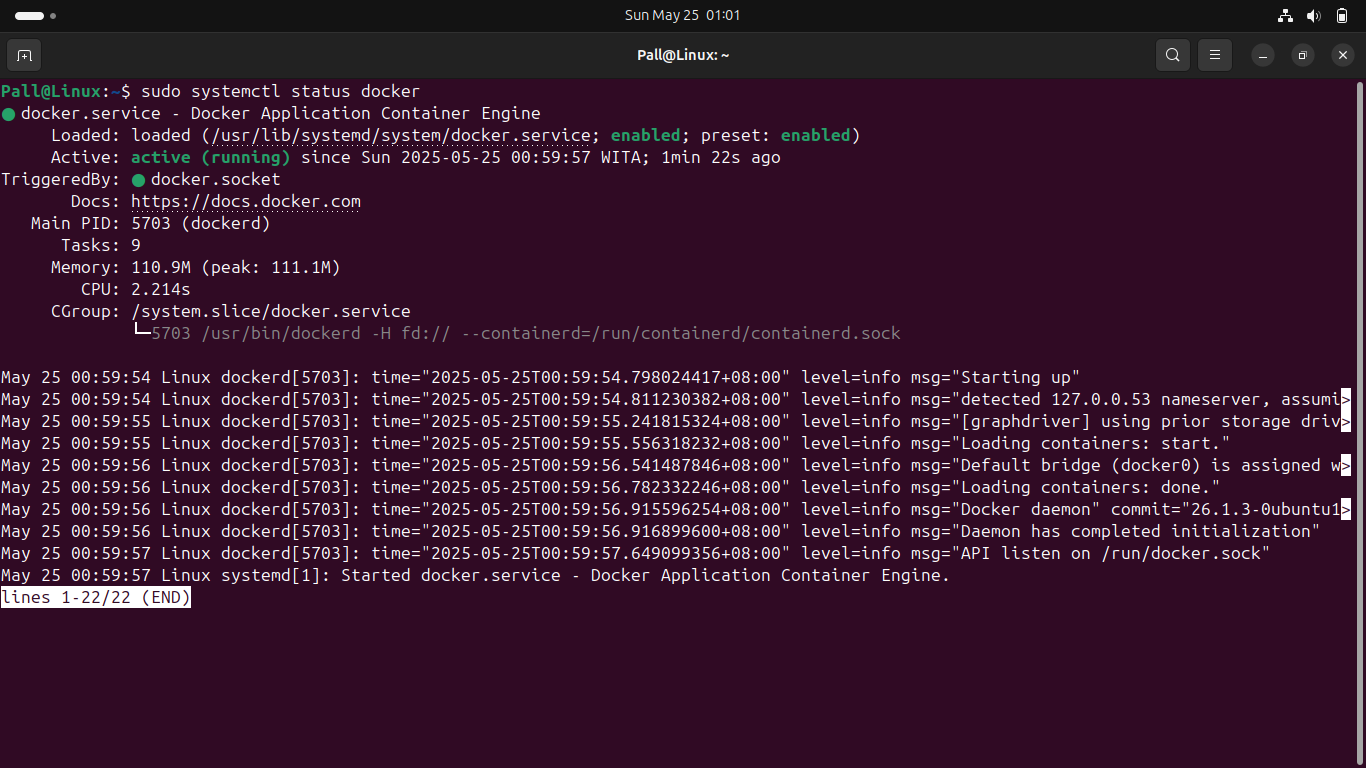


1. Menambahkan user ke grup docker agar bisa menjalankan docker tanpa prompt sudo dan membuat shell baru yang langsung menggunakan grup docker tanpa harus logout-login dengan menggunakan prompt berikut sebagai contoh :

* sudo usermod -aG docker Pall
* newgrp docker

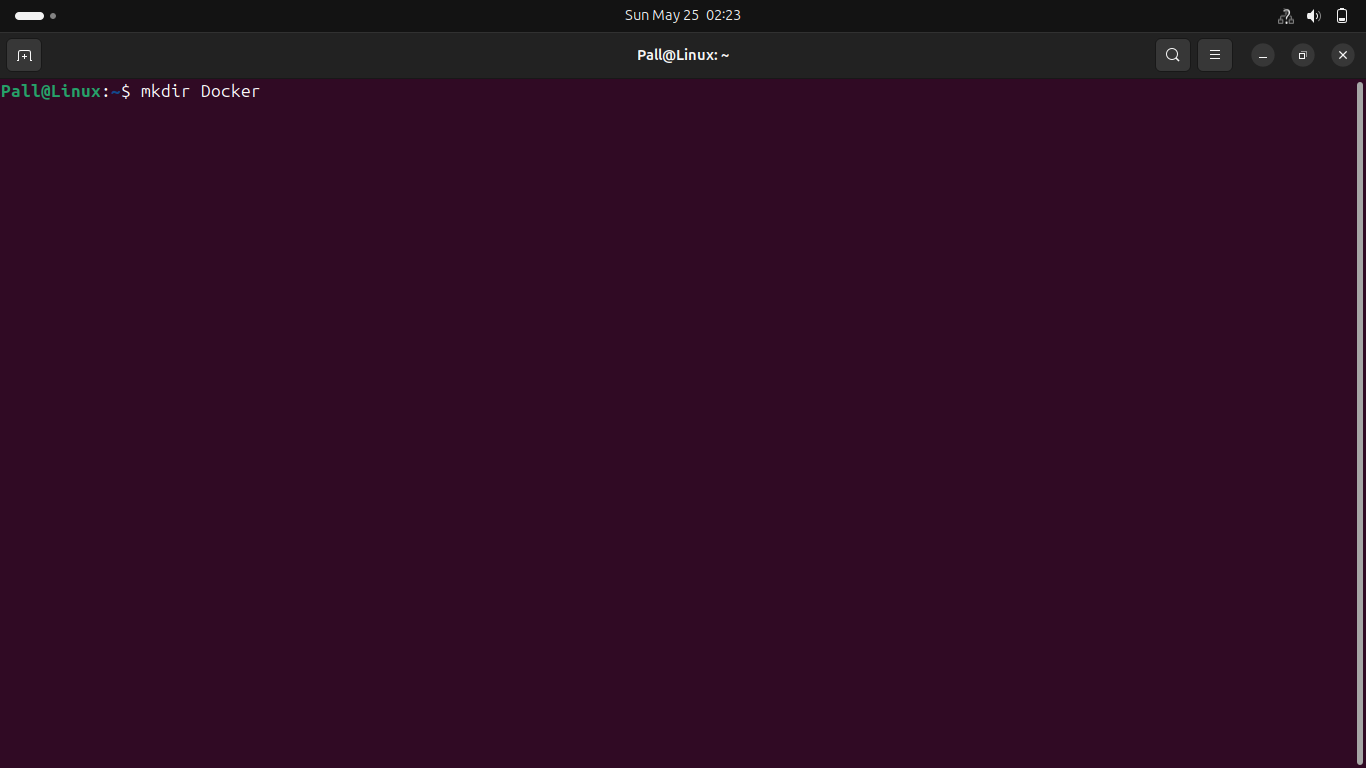


1. Melihat layanan docker apakah sudah aktif atau tidak dengan menggunakan prompt sudo systemctl status docker.



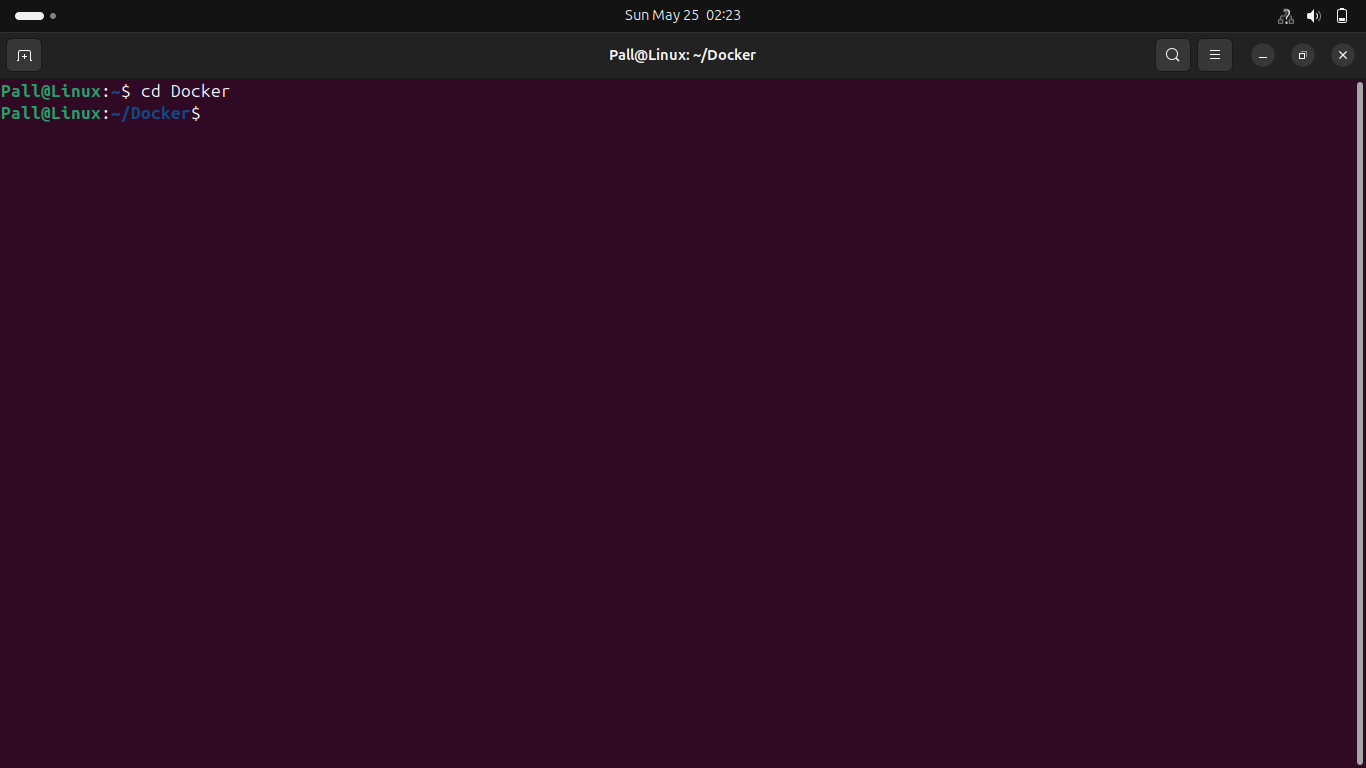
1. Membuat direktori baru yang akan dijadikan tempat menyimpan file konfigurasi dan sumber daya proyek docker dengan menggunakan prompt mkdir nama\_direktori.

Contoh : mkdir Docker



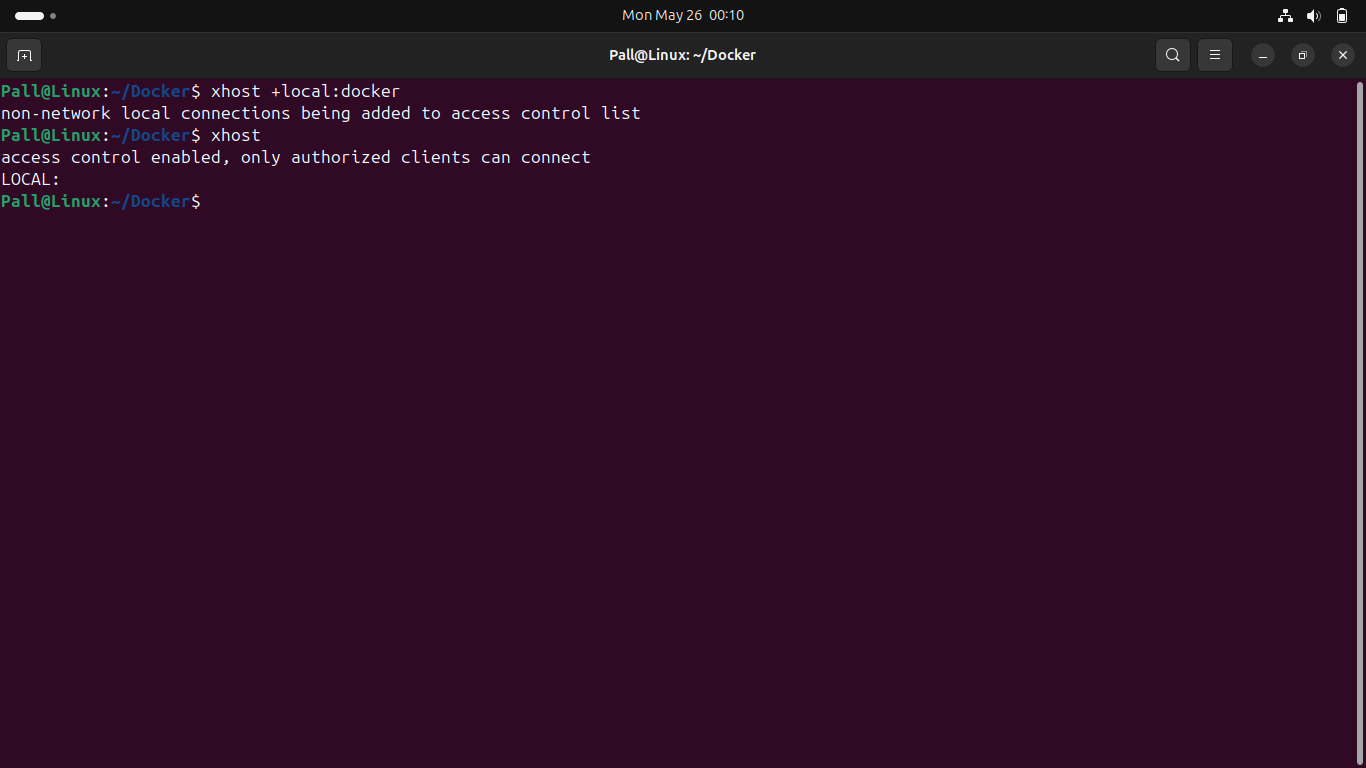
1. Masuk ke direktori yang baru di buat dengan menggunakan prompt cd nama\_direktori.

Contoh : cd Docker



1. Mengizinkan akses tampilan grafis (X11) bagi aplikasi yang akan di jalankan agar bisa menampilkan GUI di layar dengan menggunakan prompt xhost +local:nama\_user dan perintah xhost digunakan untuk memeriksa status akses X server. Dengan prompt :

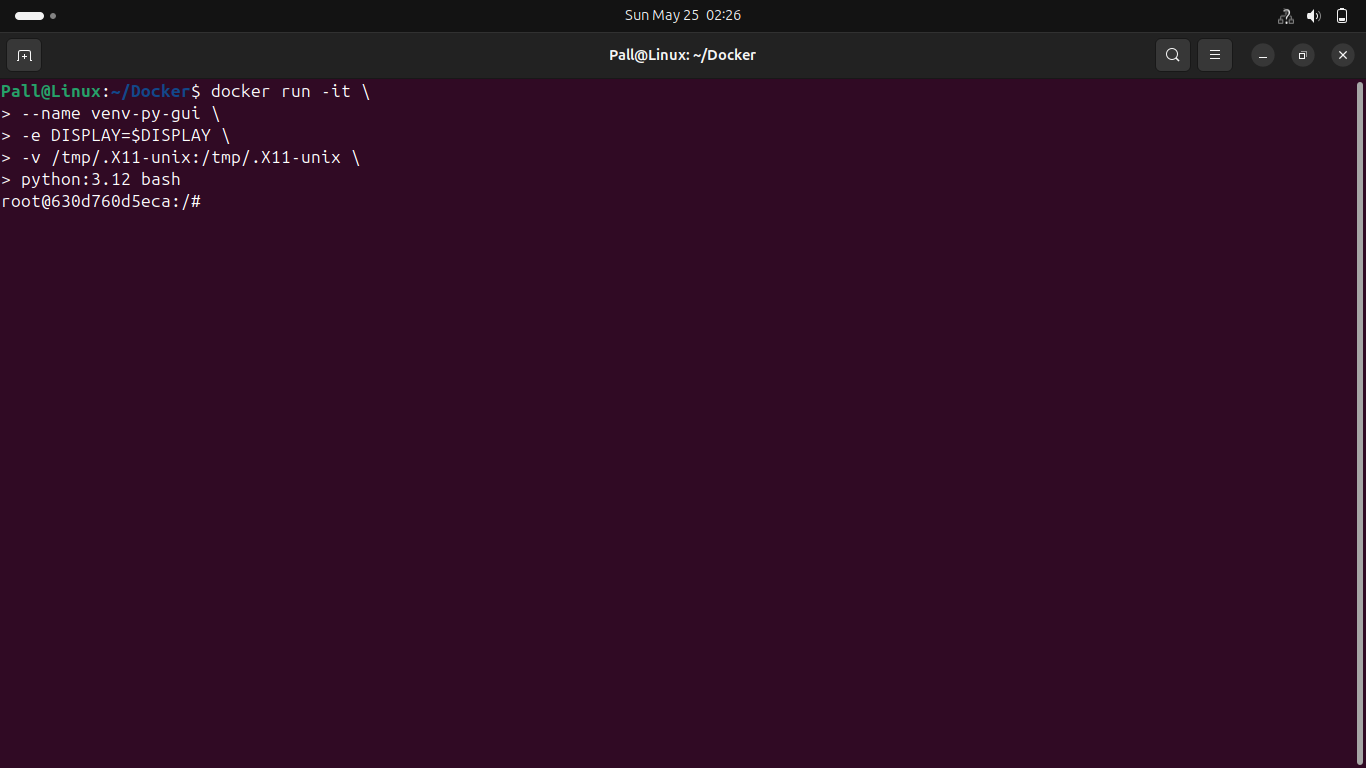
* xhost +local:docker
* xhost



1. Menjalankan container docker berbasis image python:3.12 dalam mode interaktif dengan dukungan GUI(X11) dari system host dengan prompt:

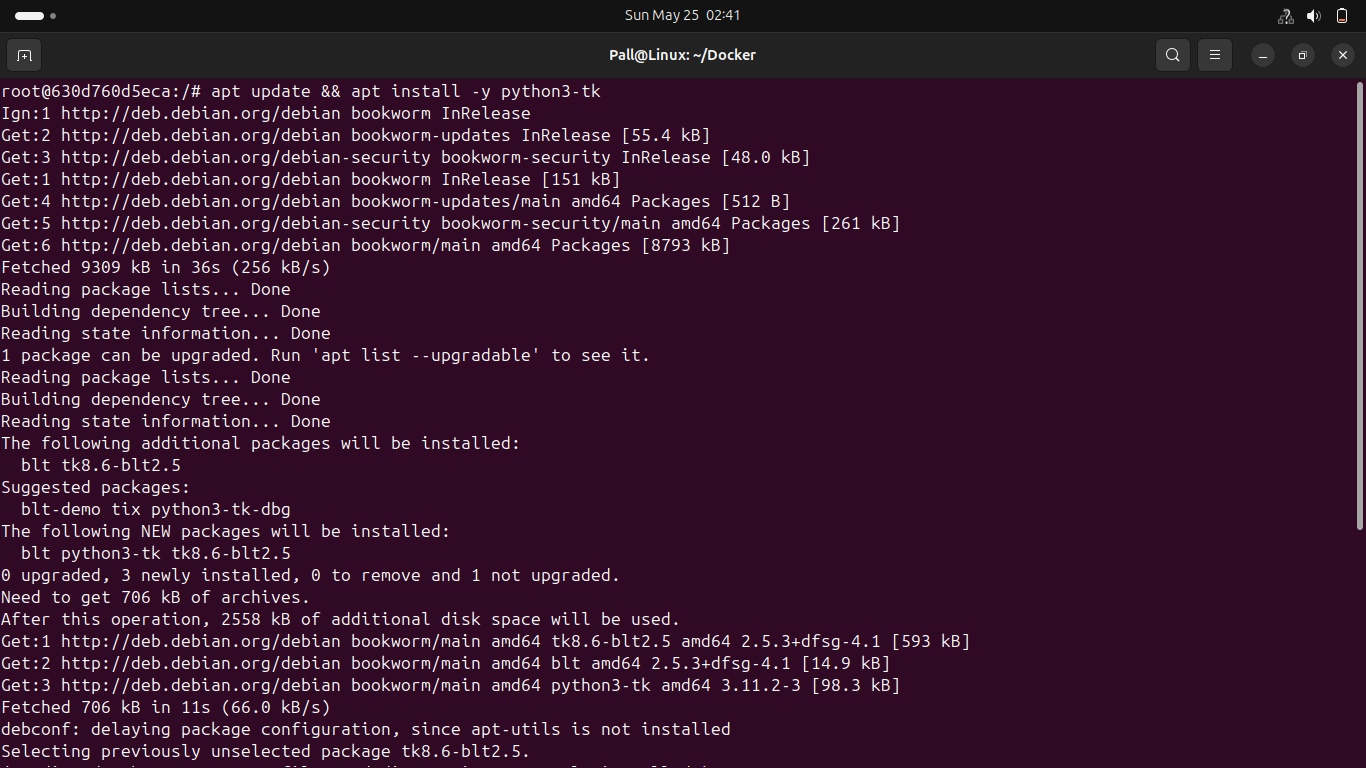
* docker run -it berfungsi untuk menjalankan container baru dengan mode interaksi dan bisa berinteraksi langsung dengan shell dalam container.
* --name venv-py-gui berfungsi untuk memberikan nama pada container agar mudah dikenali dan tidak memakai nama acak.
* -e DISPLAY=$DISPLAY berfungsi untuk mengekspor environment variable DISPLAY dari host ke dalam container agar bisa menjalankan aplikasi gui.
* -v /tmp/.X11-unix:/tmp/.X11-unix berfungsi untuk mount volume dari host untuk share X11 socket. Dengan ini container bisa mengirim tampilan GUI ke X server host.
* Python:3.12 berfungsi untuk menarik image Docker resmi berbasis pyhton 3.12 dari Docker Hub.
* bash berfungsi untuk masuk ke shell bash agar bisa langsung bekerja di dalam container.

Setelah masuk ke dalam container maka akan berada di dalam prompt shell dan user@host:~$ akan berubah menjadi root@container-id:#

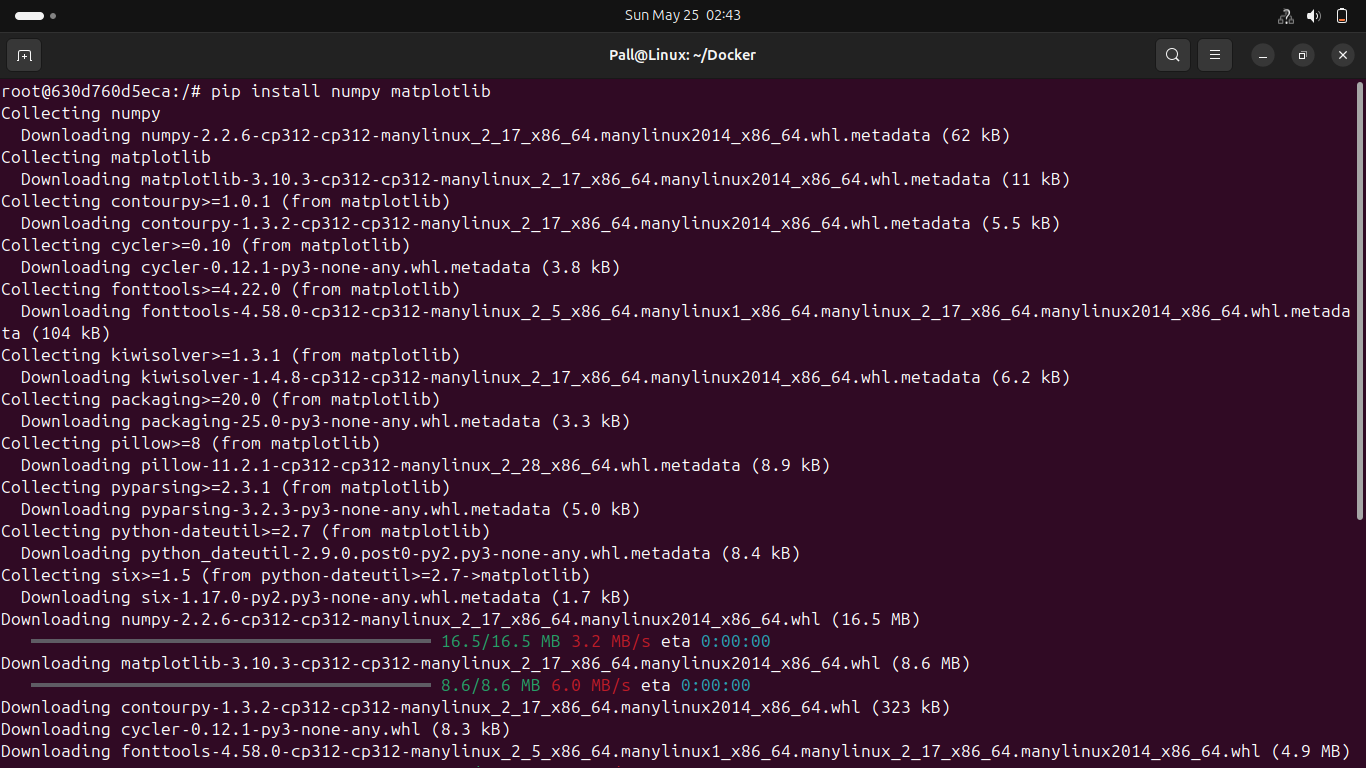


1. Menginstal modul GUI tkinter untuk python 3 agar bisa menjalankan GUI(Graphical User Interface) di dalam container, dengan menggunakan prompt:

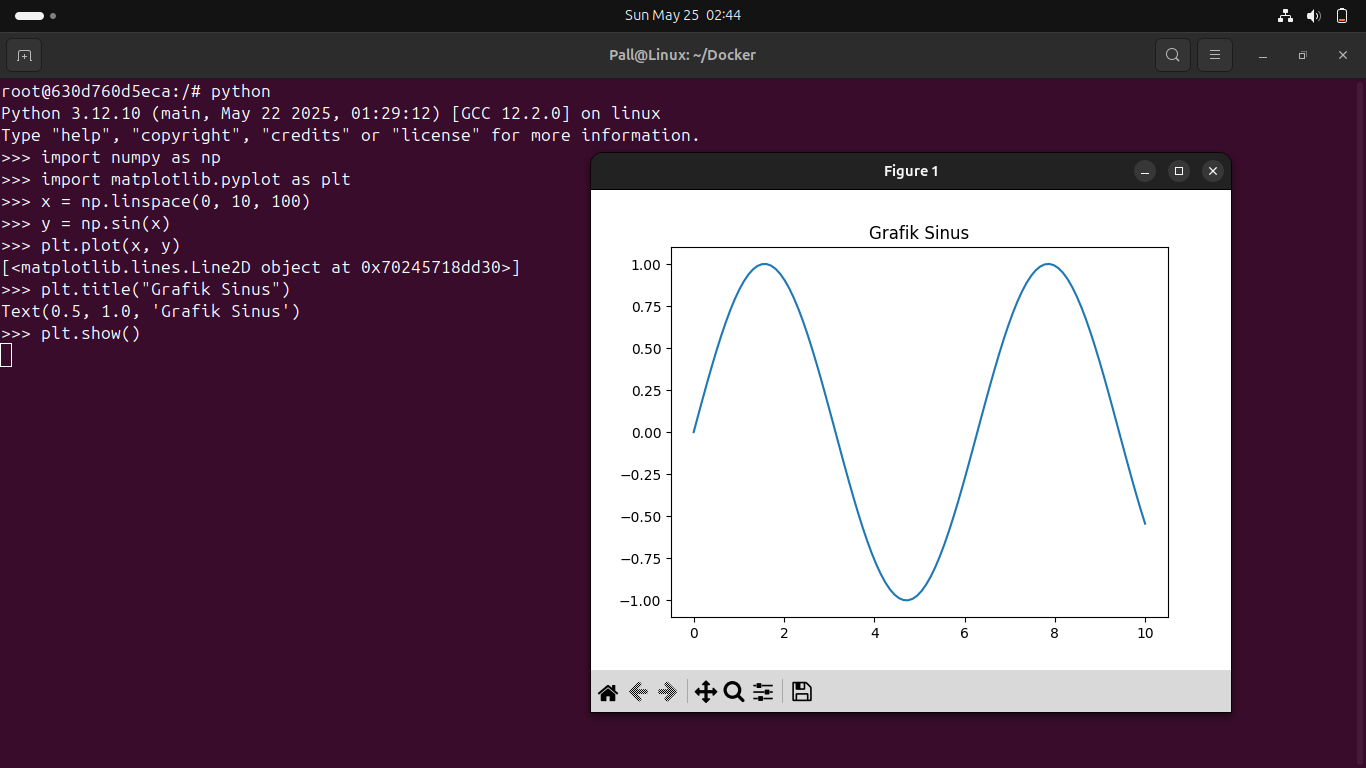
* apt update berfungsi untuk mengambil daftar paket terbaru dari repositori.
* apt install -y pyhton3-tk untuk menginstal modul GUI tkinter untuk python 3.



1. Instalasi numpy dan matplotlib di dalam container Docker.
2. Instal numpy dan matplotlib dengan menggunakan prompt pip install numpy matplotlib .



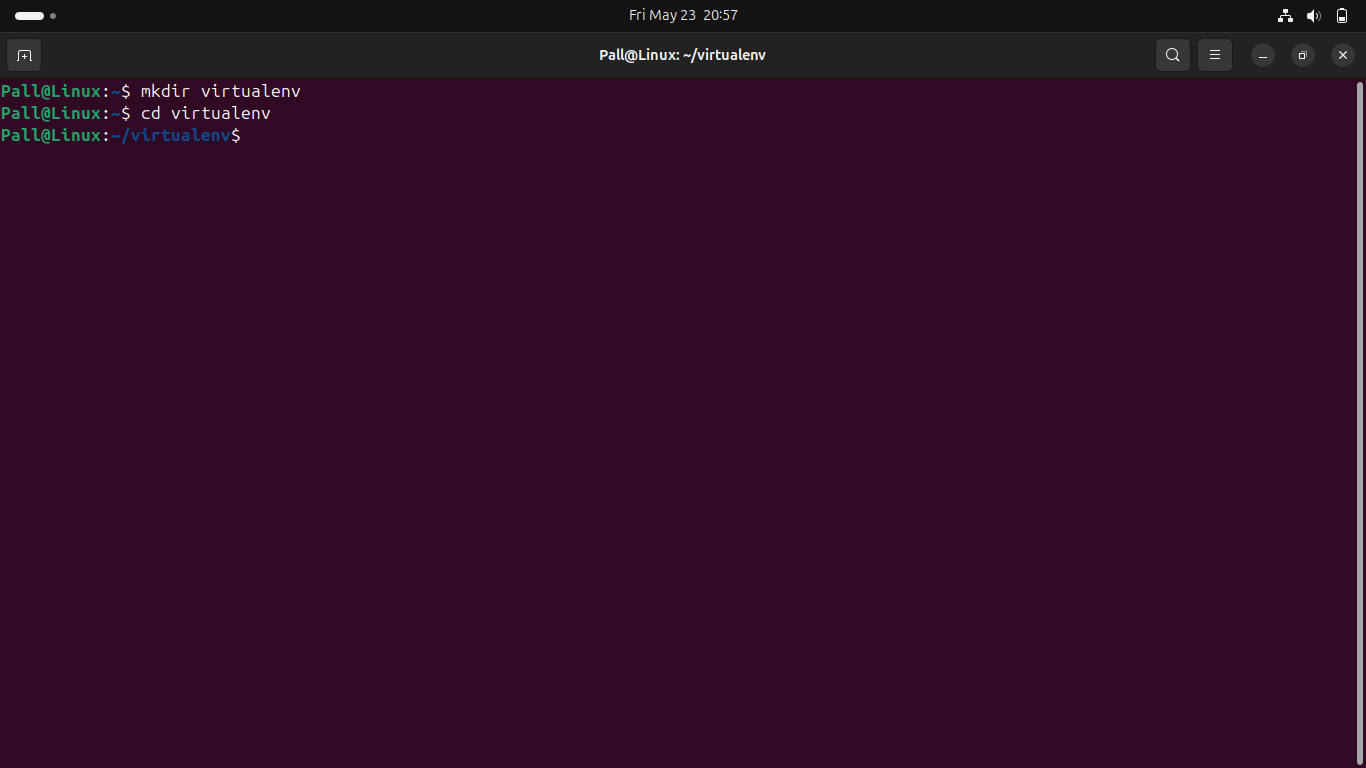
1. Menjalankan interpreter Python langsung dalam mode interaktif (REPL : Read-eval-print loop) dengan mengetik pyhton di dalam container dan bisa langsung mengeksekusi syntax python. Setelah selesai klik ctrl+d untuk keluar dari interpreter python dan container.



1. Instalasi Virtualenv pada system operasi Linux
2. Membuat Direktori baru yang akan di gunakan untuk instalasi virtualenv dengan prompt mkdir nama\_direktori lalu masuk ke dalam direktori yang di buat dengan prompt cd nama\_direktori.

Contoh :

* mkdir virtualenv
* cd virtualenv

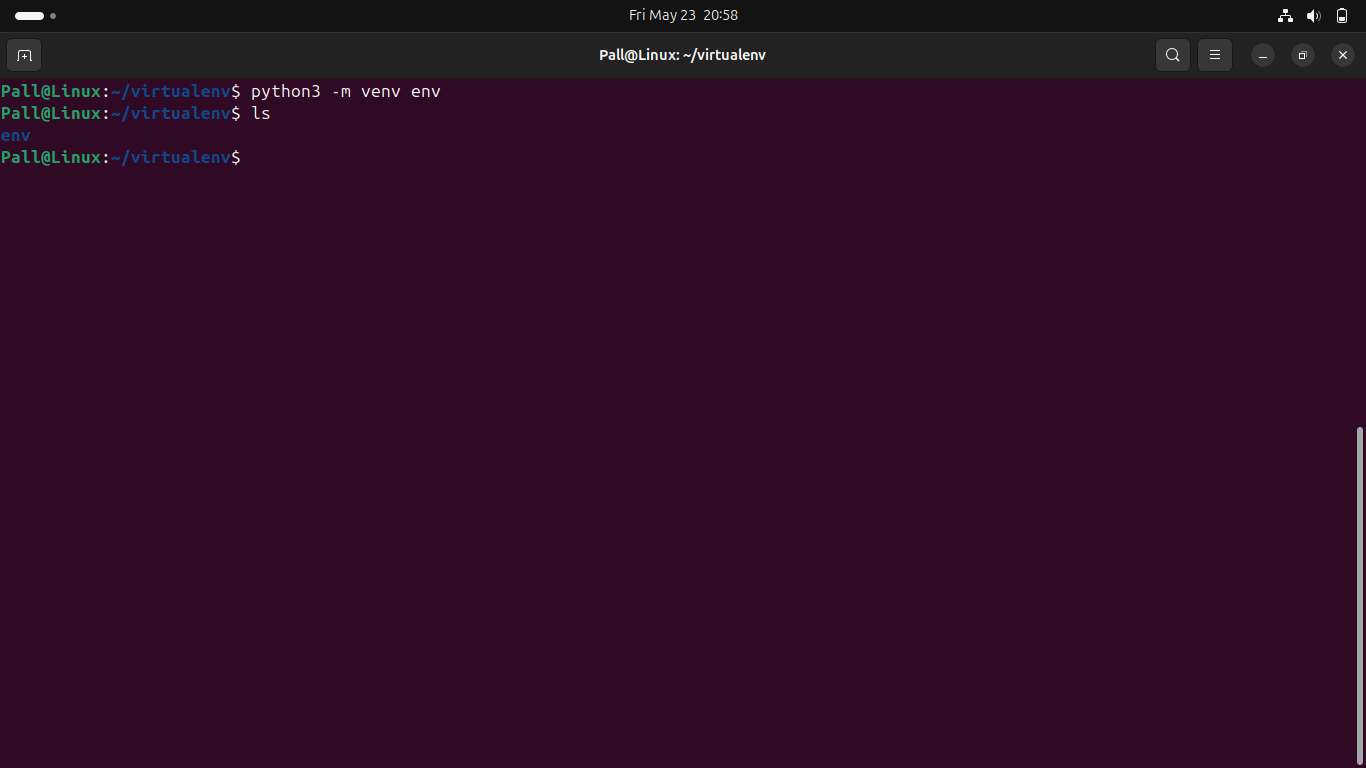


1. Membuat Virtual Environment di dalam direktori yang baru di buat dengan prompt python3 -m venv nama\_direktori , di sarankan menggunakan nama env atau venv.

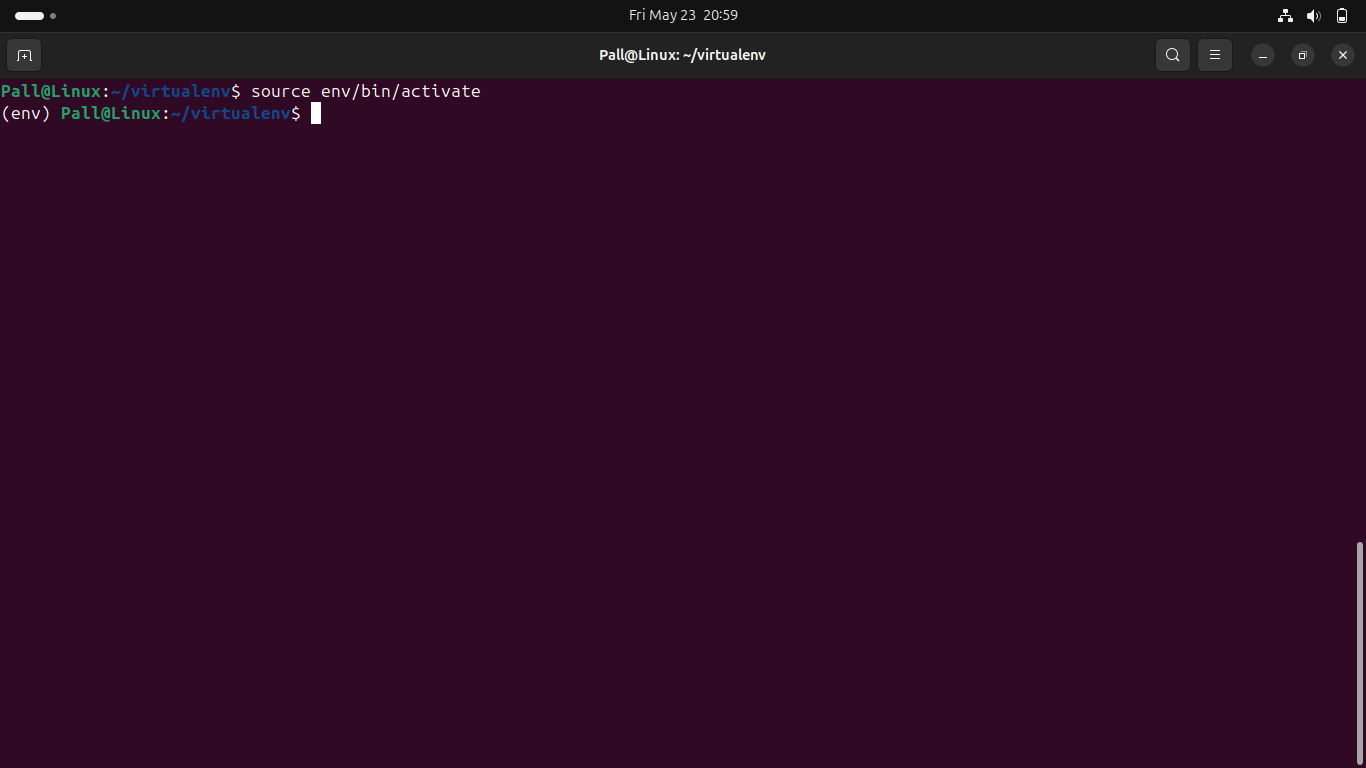
Contoh : python3 -m venv env

* Membuat direktori yang bernama env di dalam direktori virtualenv
* Semua library python dan lingkungan terisolasi ada di dalam direktori virtualenv.

Kemudian gunakan prompt ls untuk mengecek apakah Virtual Environment sudah di buat.

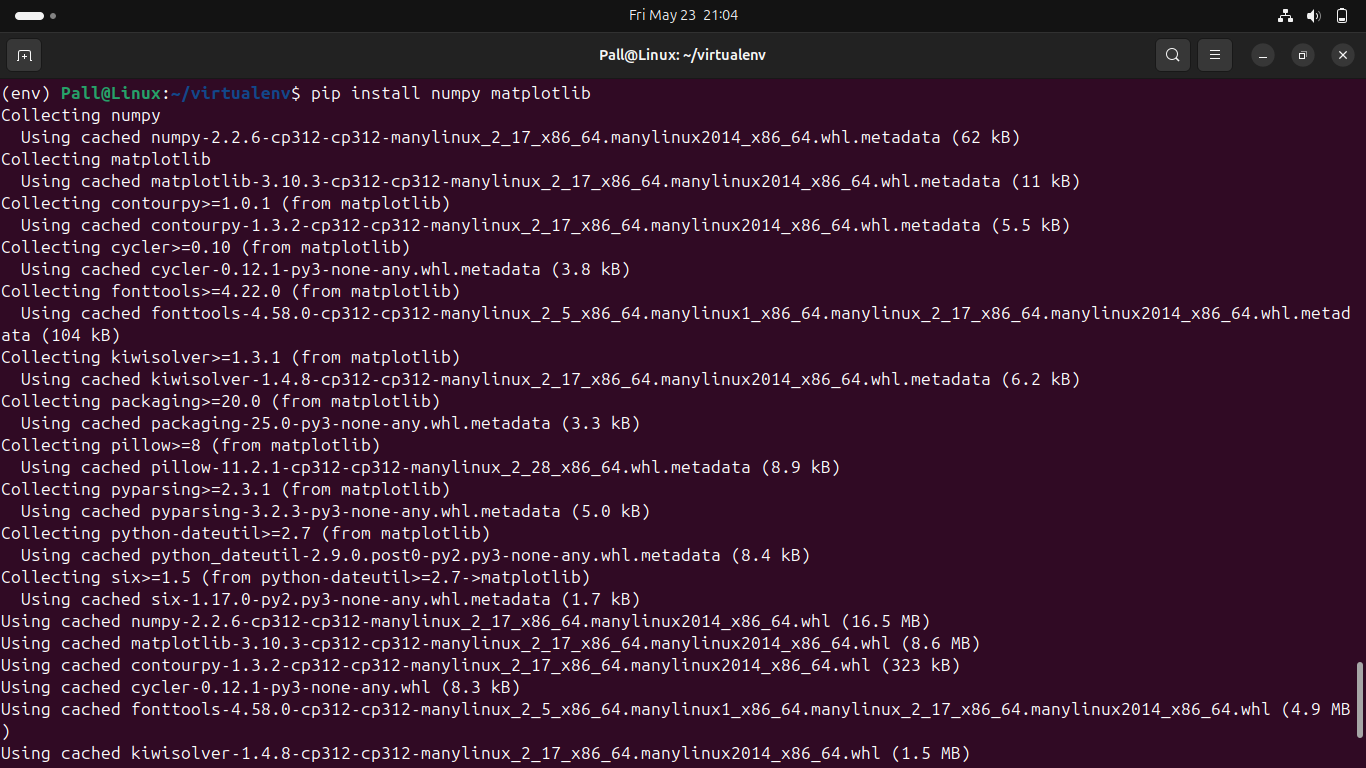


1. Mengaktifkan Virtual Environment dengan perintah source env/bin/activate. Setelah itu, prompt terminal akan berubah jadi seperti : (env) user@linux:~/virtualenv$ .



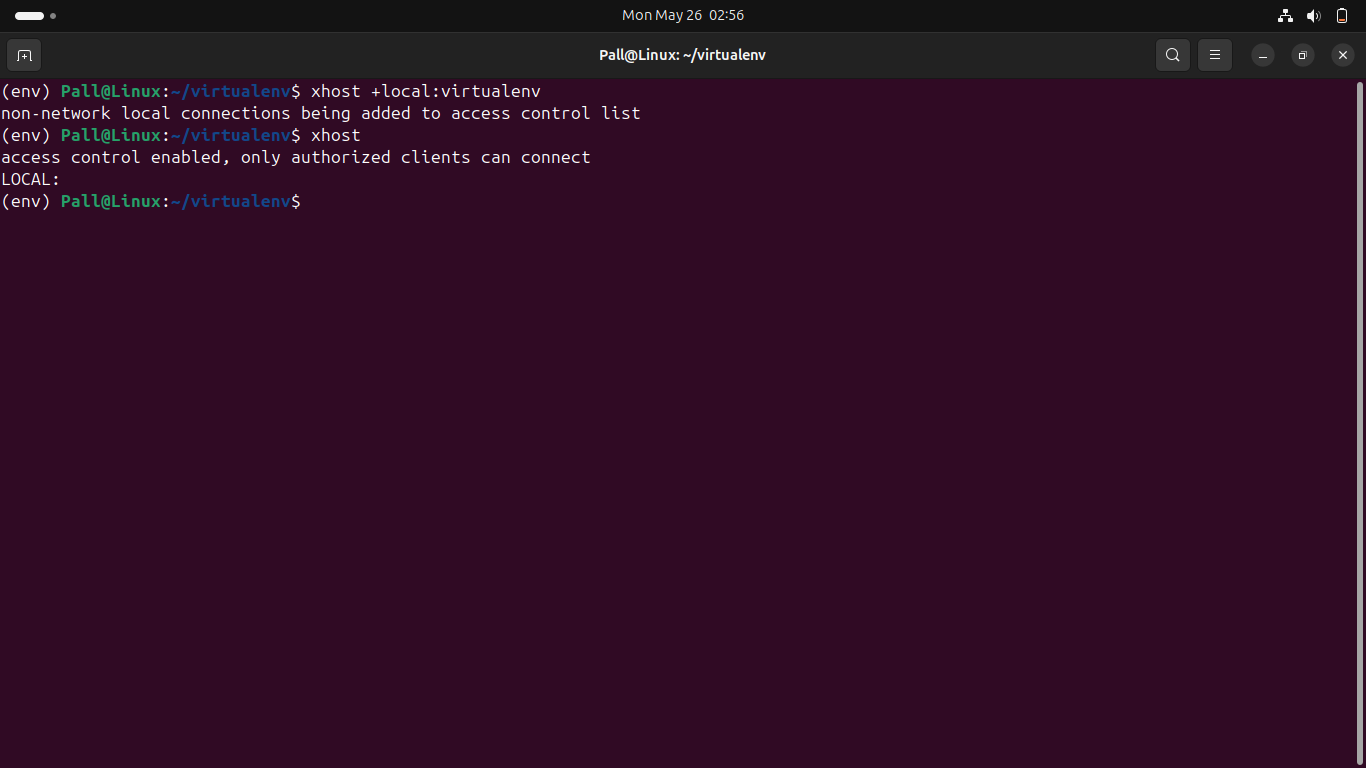
1. Instalasi numpy dan matplotlib di dalam virtualenv.

* Gunakan prompt pip install numpy matplotlib .

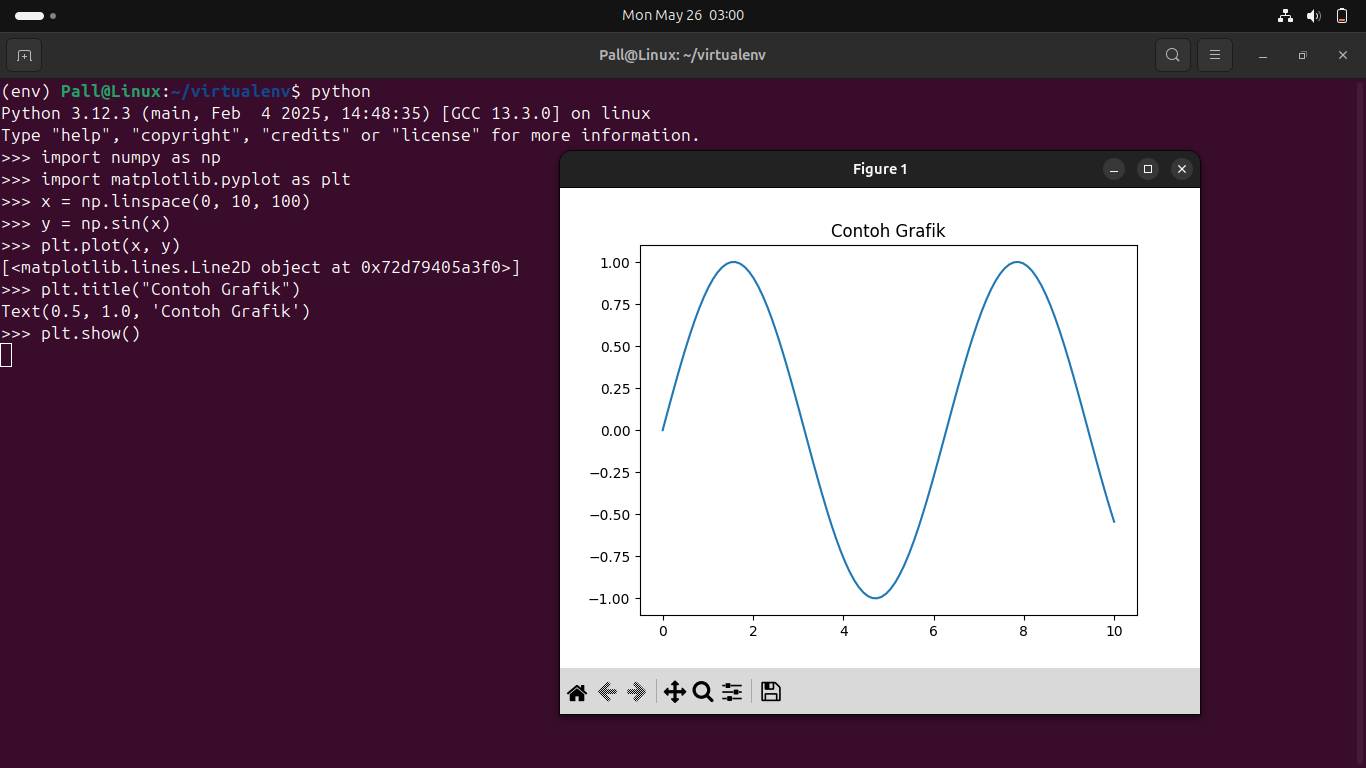


1. Mengizinkan akses tampilan grafis (X11) bagi aplikasi yang akan di jalankan agar bisa menampilkan GUI di layar dengan menggunakan prompt xhost +local:nama\_user dan perintah xhost digunakan untuk memeriksa status akses X server. Dengan prompt :

* xhost +local:virtualenv
* xhost



1. Menjalankan interpreter Python langsung dalam mode interaktif (REPL : Read-eval-print loop) dengan mengetik pyhton di dalam container dan bisa langsung mengeksekusi syntax python. Setelah selesai klik ctrl+d untuk keluar dari interpreter pyhton.



1. Nonaktifkan Virtual Environment dengan menggunakan prompt deactivate, dan terminal akan Kembali seperti di awal.

