|  |
| --- |
| kata pengantar |

Modul Praktikum Cloud Computing ini diterbitkan sebagai petunjuk bagi peserta praktikum Cloud Computing pada Program Studi Informatika Fakultas Teknik Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta.

Peserta praktikum adalah mereka yang sedang mengambil mata kuliah Cloud Computing dan mereka yang telah menempuh mata kuliah Cloud Computing tetapi belum mengikuti Praktikum Cloud Computing. Modul ini diharapkan dapat membantu para praktikan dalam menguasai teknologi Cloud Computing khusunya yang menggunakan VMware vSphere Hypervisor.

Adapun rencana jadwal kegiatan Praktikum Cloud Computing adalah sebagai berikut:

|  |  |  |
| --- | --- | --- |
| Pertemuan | Materi | Waktu |
| I | Pengenalan Praktikum Cloud Computing | 1 x 30 menit |
| Instalasi dan Konfigurasi Awal FreeNAS | 3 x 30 menit |
| II | Konfigurasi Lanjutan dan Case Study FreeNAS | 2 x 60 menit |
| III | Instalasi dan Konfigurasi VSphere ESXi | 2 x 60 menit |
| IV | Instalasi dan Pengelolaan Virtual Machine pada VSphere ESXi | 2 x 60 menit |
| V | Kuis Praktikum Cloud Computing | 2 x 60 menit |
| VI | Instalasi dan Konfigurasi Layanan Hosting dengan LAMPP (SaaS) | 2 x 60 menit |
| VII | Implementasi Layanan API pada Cloud Computing | 2 x 60 menit |
| VIII | Instalasi dan Konfigurasi Layanan Docker dengan Dokku (PaaS) | 2 x 60 menit |
| IX | Responsi Praktikum | 2 x 60 menit |
| X | Presentasi Final Project Cloud Computing | 2 x 60 menit |

Tim penyusun modul praktikum merasa bersyukur dapat menyusun modul Cloud Computing. Tim penyusun mengucapkan terima kasih kepada pihak Program Studi Informatika yang telah memberikan kepercayaan kepada kami dan dosen pengampu mata kuliah Cloud Computing yang telah memberikan bantuan dalam penyusunan dan pembuatan modul Cloud Computing.

Akhir kata, tentunya kesempurnaan dalam penyusunan Modul Cloud Computing ini membutuhkan bantuan dari para pembaca. Sehingga tim penyusun berharap dapat menerima segala kritik dan saran dari siapa saja yang berkenan demi perbaikan modul praktikum. Semoga modul ini berguna dalam proses pendidikan di Program Studi Informatika Fakultas Teknik Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta.

Yogyakarta, Juli 2019

Tim Penyusun

**TATA TERTIB PRAKTIKUM**

1. PESERTA PRAKTIKUM
2. Mahasiswa memiliki KRP yang berlaku pada semester yang bersangkutan. Dan wajib mengumpulkan 2 lembar fotokopi KRP yang telah ditandatangani oleh dosen wali dan dicap pengesahan di jurusan.
3. Mahasiswa terdaftar mata praktikum yang diambil, sesuai dengan plug yang dipilih.
4. KEHADIRAN
5. Praktikan wajib mengikuti praktikum sesuai dengan jadwal yang telah ditentukan.
6. Hadir di laboratorium paling lambat 5 (lima) menit sebelum praktikum dimulai.
7. Praktikan tidak diperkenankan terlambat, keterlambatan mengurangi nilai atau membatalkan praktikum.
8. Presensi akan dilaksanakan pada 15 menit pertama masuk dan bagi yang terlambat tidak dapat melakukan presensi.
9. Jika Praktikan tidak hadir praktikum (tidak melakukan presensi) lebih dari 3 kali pertemuan tidak diperkenankan untuk mengikuti Responsi / Ujian Praktikum.
10. Waktu di luar jam jadwal praktikum praktikan, dilarang memasuki laboratorium tanpa seijin Dosen/Instruktur/Asisten yang bertugas pada jam tersebut.
11. ETIKA PELAKSANAAN PRAKTIKUM
12. Praktikan wajib memiliki Modul praktikum.
13. Praktikan wajib berpakaian yang sopan (rapi, berkerah bukan kaos oblong, dan tidak ketat).
14. Alas kaki dilepas di luar ruangan laboratorium (diletakkan pada tempatnya dengan rapi).
15. Tas diletakkan pada tempat yang telah ditentukan, bukan di atas meja komputer, membawa alat tulis, kertas/buku, dan modul praktikum.
16. Selama praktikum, Praktikan dilarang makan, minum, merokok, dan lain-lain yang dapat mengganggu jalannya praktikum.
17. Praktikan wajib menjaga kebersihan ruang laboratorium komputer dan peralatan yang digunakan.
18. Praktikan tidak memindahkan, merubah Setup, dan mengganti perangkat komputer atau peralatan lain yang ada di laboratorium komputer.
19. Praktikan tidak diperkenankan untuk menyalakan HP di dalam ruangan laboratorium karena akan mengganggu jaringan (wireless) yang ada di dalam laboratorium.
20. Praktikan wajib menaati asisten dan coass.
21. Asisten berhak menunda atau membatalkan praktikan untuk praktikum, jika praktikan tersebut tidak mau mengikuti peraturan (tata tertib di dalam laboratorium).
22. Praktikan wajib mengganti alat-alat yang rusak akibat keteledoran atau kurang hati-hati menggunakan alat-alat di dalam laboratorium.
23. Setelah selesai melaksanakan praktikum, praktikan wajib mengembalikan alat seperti pada saat praktikan mulai mempergunakannya.
24. Apabila Praktikan melakukan pengubahan jadwal praktikum harus seijin asisten yang bersangkutan dan memperhatikan jadwal waktu laboratorium serta kapasitas kelas.
25. LAPORAN
26. Praktikan wajib mengumpulkan tugas/laporran kegiatan praktikum pertemuan sebelumnya, sebelum mengikuti praktikum, tidak melaksanakan tugas, mengakibatkan pembatalan praktikum atau pengurangan nilai laporan.
27. Ketentuan penulisan laporan disesuaikan dengan tugas dan asisten masing-masing.
28. PENILAIAN
29. Penilaian praktikum terdiri dari : Tugas/kuis, ketrampilan/keaktifan/kinerja, dan laporan praktikum harian serta responsi praktikum, dengan persentase yang disesuaikan oleh Asisten Praktikum masing-masing.
30. Hal-hal yang tidak dicantumkan dalam tata tertub praktikum ini, akan ditentukan dalam buku pratikum.

|  |
| --- |
| pendahuluan |

## Apa itu Cloud Computing?

Cloud computing atau dikenal juga dengan nama komputasi awan merupakan teknologi komputasi dihubungkan kepada jaringan yang berbasis internet. Hal ini memungkinkan untuk menjalankan beberapa aplikasi dan program pada beberapa komputer dalam waktu bersamaan. Menurut NIST, cloud computing adalah model untuk memungkinkan untuk diakses dari manapun, nyaman, sesuai permintaan akses jaringan bersama dengan sumber daya komputasi yang telah dikonfigurasi (misalnya, jaringan, server, penyimpanan, aplikasi, dan layanan) yang dapat dengan cepat ditetapkan dan dirilis dengan usaha manajemen minimal dari penyedia layanan. Model cloud ini terdiri dari lima karakteristik penting, tiga model layanan, dan empat model pengerahan.

## Karakterisitik Cloud Computing

Berikut ini adalah lima karakteristik penting cloud computing menurut NIST.

### On-demand self service

On-demand self service atau layanan mandiri sesuai permintaan merupakan layanan yang dimiliki oleh sistem dimana konsumen dapat secara sepihak menyediakan kemampuan komputasi, seperti waktu server dan penyimpanan jaringan, sesuai kebutuhan secara otomatis tanpa memerlukan interaksi manusia dengan setiap penyedia layanan.

### Broad network access

Broad network akses merupakan kemampuan sistem yang dapat tersedia pada jaringan dan diakses melalui mekanisme standar pada platform kecil (misalnya, ponsel, tablet, laptop, dan workstation).

### Resource pooling

Sumber daya komputasi penyedia layanan dikumpulkan untuk melayani beberapa konsumen menggunakan model multi-penyewa, dengan sumber daya fisik dan virtual yang berbeda secara dinamis ditugaskan dan akan ditugaskan kembali sesuai dengan permintaan konsumen. Hal ini memberikan keringanan untuk pelanggan dimana umumnya pelanggan tidak memiliki kontrol atau pengetahuan atas tepat lokasi sumber daya yang disediakan tetapi mungkin dapat menentukan lokasi pada tingkat yang lebih tinggi abstraksi (misalnya, negara, provinsi, atau Datacenter). Contoh sumber daya termasuk penyimpanan, pemrosesan, memori, dan bandwidth jaringan.

### Rapid elasticity

Sistem bersifat elastis atau otomatis pada saat ditetapkan dan dirilis, sehingga dengan cepat dapat memproses permintaan masuk maupun keluar. Untuk konsumen, kemampuan yang disediakan sering tidak terbatas dan dapat disesuaikan jumlahnya setiap saat.

### Measured service

Sistem cloud secara otomatis mengontrol dan mengoptimalkan penggunaan sumber daya dengan memanfaatkan kemampuan metering pada beberapa tingkat abstraksi yang sesuai dengan jenis layanan (misalnya, Penyimpanan, pemrosesan, bandwidth, dan akun pengguna aktif). Penggunaan sumber daya dapat dipantau, dikontrol, dan dilaporkan, memberikan transparansi bagi penyedia dan konsumen dari layanan yang dimanfaatkan.

## Model Layanan

Adapun jenis layanan yang tersedia pada cloud computing menurut NIST adalah sebagai berikut.

### Software as a Service (SaaS)

Layanan yang diberikan kepada konsumen adalah layanan yang bisa diakses menggunakan aplikasi penyedia layanan pada infrastruktur cloud. Aplikasi ini dapat diakses dari berbagai perangkat klien baik melalui antarmuka klien tipis, seperti browser web (misalnya, email berbasis web), atau antarmuka program. Konsumen tidak mengelola atau mengontrol infrastruktur cloud seperti jaringan, server, sistem operasi, penyimpanan, atau bahkan kemampuan aplikasi secara spesifik.

Salah satu keuntungan dari SaaS adalah pengguna tidak perlu membeli lisensi baru setiap ada peningkatan software ke versi terbaru. Pengguna dimudahkan untuk bisa berlanggan dan membayar sesuai pemakaian kepada penyedia layanan SaaS. Contoh produk SaaS: Office365, Gmail, Adobe CC, dan Skype.

### Platform as a Service (PaaS)

Layanan yang diberikan kepada konsumen adalah pemberian akses ke infrastruktur cluod yang dibuat sehingga aplikasi yang diperoleh atau yang dibuat konsumen bisa berjalan. Konsumen tidak mengelola atau mengendalikan infrastruktur cloud seperti jaringan, server, sistem operasi, atau penyimpanan, tetapi memiliki kontrol atas aplikasi yang dijalankan dan mungkin pengaturan konfigurasi untuk hosting.

Keuntungan dari PaaS adalah konsumen tidak perlu memikirkan tentang kebutuhan maupun pemeliharaan lingkungan (OS, jaringan, database engine, framework, dll.). Hal tersebut merupakan tanggung jawab dari penyedia layanan. Contoh PaaS: Amazon Web Service (AWS), Windows Azure, dan GoogleApp Engine.

### Infrastructure as a Service (IaaS)

Layanan yang diberikan kepada konsumen adalah untuk penyediaan pengelolaan, penyimpanan, jaringan, dan sumber daya komputasi lainnya. Konsumen dapat menyebarkan dan menjalankan perangkat lunak secara bebas termasuk sistem operasi dan aplikasi. Konsumen tidak mengelola atau mengontrol infrastruktur cloud yang mendasar tetapi memiliki kontrol atas sistem operasi, penyimpanan, dan aplikasi yang dikerahkan.

Keuntungan dari IaaS adalah konsumen tidak perlu membeli komputer fisik, dan konfigurasi komputer virtual dapat diubah dengan mudah. Jika komputer sudah kelebihan beban, bisa dilakukan upgrade CPU, RAM, maupun storage. Contoh penyedia layanan IaaS: Amazon EC2 dan Windows Azure.

## Model Penyebaran

Selain memiliki 5 karakakteristik utama dan 3 model layanan, cloud computing juga memiliki 4 model penyebaran, yaitu *private cloud, community cloud, public cloud,* dan *hybrid cloud*.

### Private Cloud

Infrastruktur cloud ditetapkan untuk penggunaan eksklusif oleh satu organisasi yang terdiri dari beberapa konsumen (misalnya unit bisnis). Ini mungkin dimiliki, dikelola, dan dioperasikan oleh organisasi, dan/atau pihak ketiga.

### Community Cloud

Infrastruktur cloud disediakan untuk penggunaan eksklusif oleh komunitas konsumen tertentu dari organisasi yang memiliki kepentingan bersama (misalnya, misi, persyaratan keamanan, kebijakan, dan pertimbangan). Ini mungkin dimiliki, dikelola, dan dioperasikan oleh satu atau lebih organisasi dalam komunitas, dan/atau pihak ketiga.

### Public Cloud

Infrastruktur cloud disediakan untuk penggunaan terbuka oleh publik. Ini mungkin dimiliki, dikelola, dan dioperasikan oleh bisnis, akademik, dan/atau organisasi pemerintah.

### Hybrid Cloud

Infrastruktur cloud yang terdiri dari dua atau lebih infrastruktur cloud yang berbeda (pribadi, komunitas, atau publik) yang tetap menjadi entitas unik, namun terikat bersama oleh teknologi standar atau proprietary yang memungkinkan portabilitas data dan aplikasi.

## Resiko Cloud Computing

Selain memiliki keutungan dengan layanan dan model yang diberikan, komputasi cloud juga memiliki resiko seperti waktu respons, proteksi data, maupun kepemilkan data. Adapun resiko cloud computing dapat dibagi menjadi beberapa, antara lain.

### Service Level

Provider mungkin tidak konsisten dengan performa layanan yang mengakibatkan jeleknya transaksi. Hal ini mengharuskan pelanggan untuk memahami secara baik tentang layanan apa saja yang diberikan oleh provider.

### Privacy

Hosting yang digunakan bersifat sementara dan dimiliki secara fisik oleh penyedia layanan sehingga terdapat kemungkinan adanya kebocoran data atau adanya aliran datan masuk maupun keluar tanpa sepengetahuan pelanggan.

### Compliance

Pelanggan sebaiknya mengetahui level layanan yang disediakan oleh provider. Hal ini berguna untuk pemenuhan layanan yang seharusnya diterima oleh pelanggan dan untuk menghindari kerugian yang disebabkan oleh kecurangan yang dilakukan penyedia layanan.

### Kepemilikan Data

Pelanggan harus paham dengan aturan main atau legal kontrak terkait kepemilikan data ketika menggunakan layanan cloud.

### Mobilitas Data

Mobilitas data pelanggan merupakan hal yang cukup krusial. Misalnya kemungkinan pengambilan maupun pemindahan data ketika terjadi pemutusan kontrak atau selesainya masa kontrak.

|  |
| --- |
| modul 1 instalasi dan konfigurasi freenas |

## Tujuan Praktikum

* Memahami cara menginstall FreeNas
* Melakukan konfigurasi pada FreeNAS

## Alokasi Waktu

1 x pertemuan = 120 menit

## Dasar Teori

### NAS

Network Attached Storage (NAS) merupakan suatu perangkat penyimpanan (storage) yang tersambung ke jaringan sehingga memungkinkan proses pengambilan dan penyimpanan data dapat dilakukan oleh lebih dari satu klien pada suatu lokasi yang terpusat (centralized). Perangkat NAS biasanya tidak memiliki keyboard maupun tampilan/layar karena dikonfigurasi dan dikelola menggunakan alat bantu yang berbasis web.

### Keuntungan Menggunakan NAS

#### Faster data transfer

Salah satu keuntungan terbesar NAS adalah penggabungan tempat penyimpanan dalam suatu organisasi, sehingga memungkinkan pengguna melakukan kerjasama dan transfer data yang lebih cepat.

#### Reliable

NAS memiliki daya tahan lebih baik daripada storage biasa, sehingga administrator tidak perlu khawatir soal kerusakan perangkat.

#### Easier to manage

NAS memiliki keberadaan yang tersentralisasi dan mudah diakses sehingga penyimpanan dan pemulihan data dapat dilakukan dengan mudah dan dari manapun.

#### Cloud services

Kebanyakan penyedia NAS dikombinasikan dengan layanan cloud sehingga memungkinkan pelanggan untuk dapat melakukan control jarak jauh (remote) dan sinkronisasi. Selain itu, layanan cloud juga memungkinkan sharing data secara fleksible sehingga pelanggan tidak perlu khawatir masalah hardware yang akan digunakan.

#### Smaller Size

Ukuran NAS yang lebih kecil dari server sangat memudahkan karena dapat bersifat portable dan juga membutuhkan energi serta ruangan yang lebih kecil disbanding server lainnya.

### FreeNAS

FreeNAS merupakan suatu platform penyimpanan open-source berbasis FreeBSD yang bias dijalankan pada Windows, Apple, system berbasis Unix, bahkan bisa diinstal di dalam mesin virtual. FreeNAS memungkinkan pengguna menyimpan data pada lokasi yang tersentralisasi dan dapat diakses dari beberapa tempat secara bersamaan.

FreeNAS menggunakan Zettabyte File System (ZFS) yang merupakan suatu file system open-source yang dikombinasikan dengan logical volume manager. ZFS dikembangkan oleh Sun Microsystems (sekarang menjadi Oracle Corporation) dengan kemampuan seperti berikut:

#### Copy-on-write

Memungkinkan pengeditan file tanpa perubahan pada blok disk.

#### Massive Scalability

Mampu menampung hingga 256 zebibytes.

#### Compression

Kompresi data pada ZFS menggunakan sistem kompresi pada level blok disk sehingga memungkinkan semua data dapat dikompres.

#### Cache Management

ZFS menggunakan layer disk berbeda dengan penyimpanan untuk mengelola cache sehingga memungkinkan percepatan proses pengolahan cache.

#### Snapshots and clones

Penggunaan metode write-on-copy memungkinkan ZFS melakukan snapshot data lebih cepat ketika proses menulis (write) data baru.

#### Storage Pool

Berbeda dengan file sistem lainnya yang membutuhkan ukuran geometri yang tetap (fixed-size), NFS memungkinkan pengguna untuk membuat file sistem dengan geometri dan konfigurasi sesuai dengan yang diinginkan.

#### Continuous Integrity Checking

ZFS melakukan pengecekan (checksum) secara kontinu dengan cara membuat dan melakukan verifikasi checksum sebagai data yang ditulis ke disk.

#### Deduplication

ZFS mampu melakukan penghapusan duplikasi data dari level file, blok, hingga byte. Hal ini memberi ruangan yang lebih banyak untuk penyimpanan.

#### Thin Provisioning

Thin Provisioning (TP) merupakan suatu metode untuk efisiensi ruangan untuk penyimpanan dengan cara mengalokasi ruang penyimpanan secara fleksibel untuk para pengguna berdasarkan penggunaan minimum.

#### RAID Support

Mendukung penggunaan RAID, seperti Stripe, Mirror, dan RAIDZ.

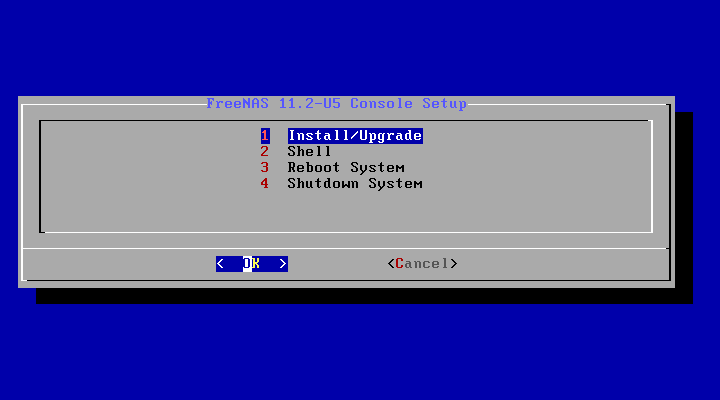
## Praktikum

### Instalasi Free NAS

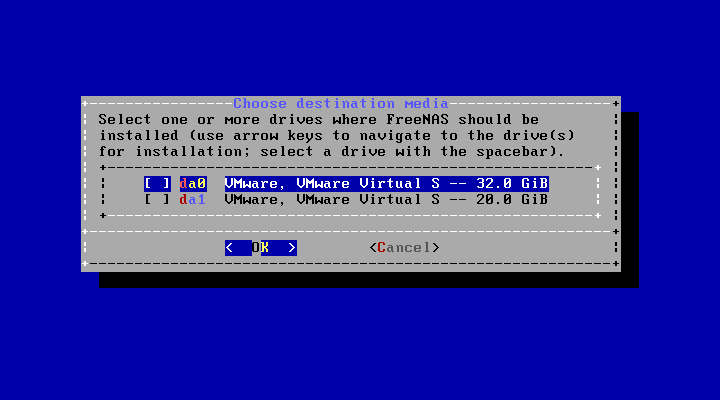
Setup pertama instalasi FreeNAS klik enter.



Pilih Install/Upgrade



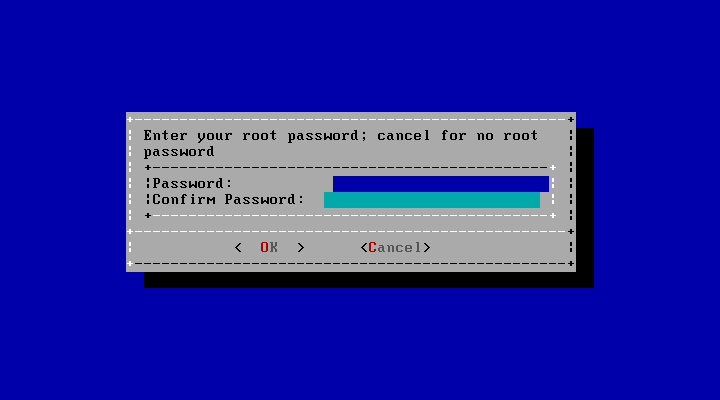
Pilih drive untuk install sistem operasi FreeNAS dengan **spasi** lalu **enter**



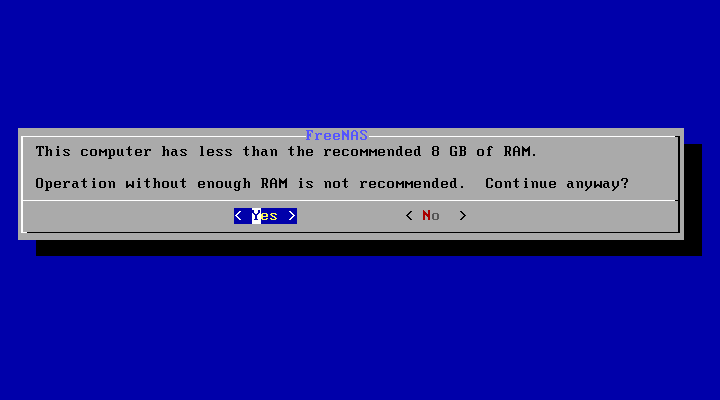
Pilih Yes



Buat Password untuk user Root



Pilih Yes



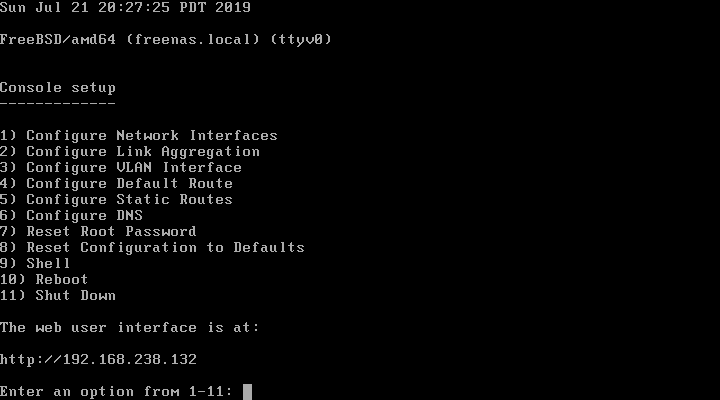
Pilih BIOS dan tunggu smpai instalasi selesai



Instalasi FreeNAS Selesai

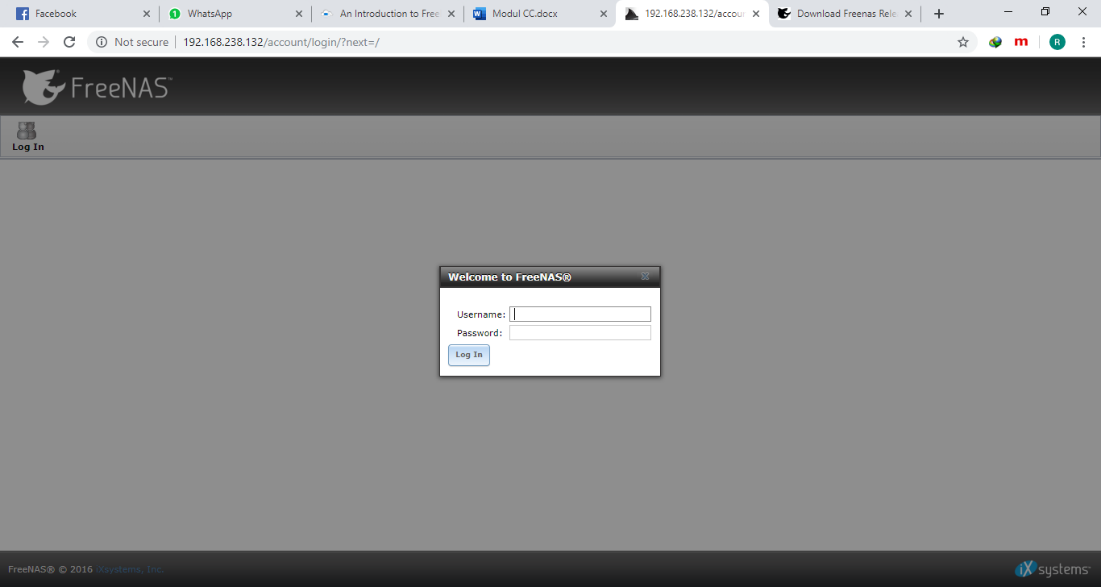


Tampilan Console FreeNAS

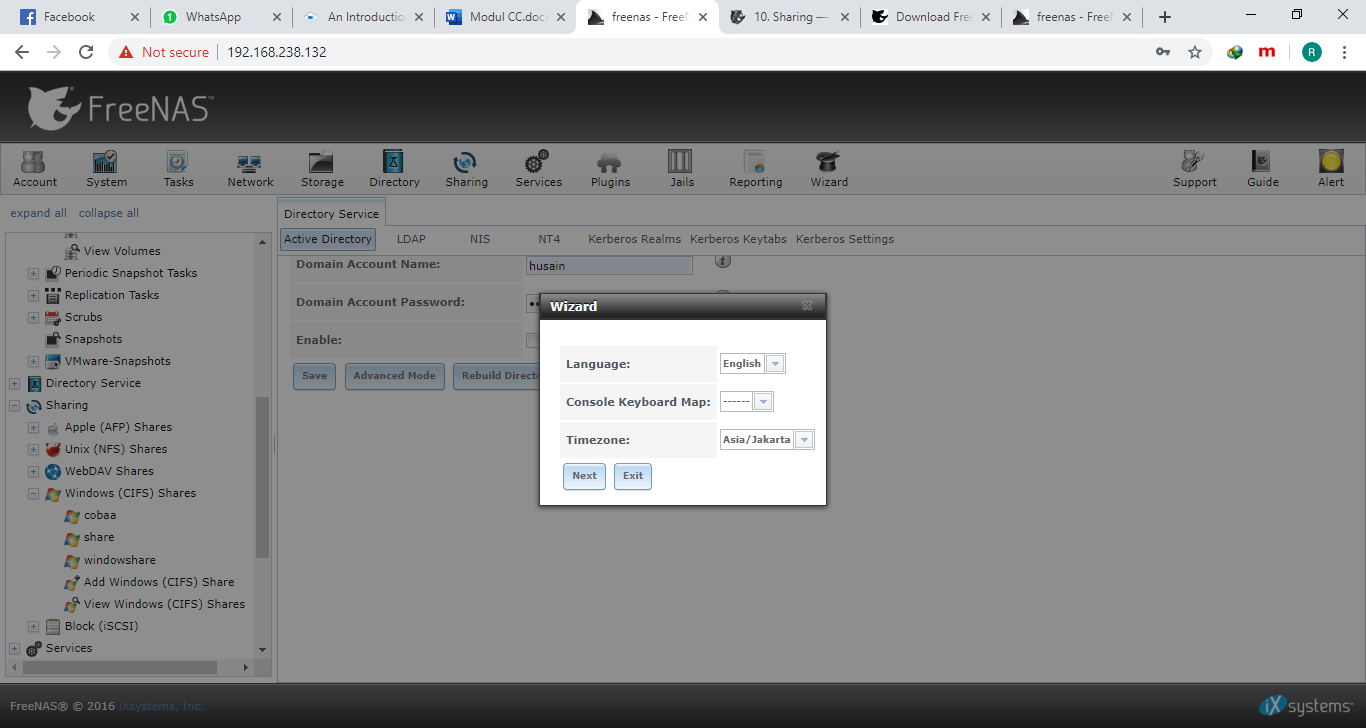


### Konfigurasi FreeNAS

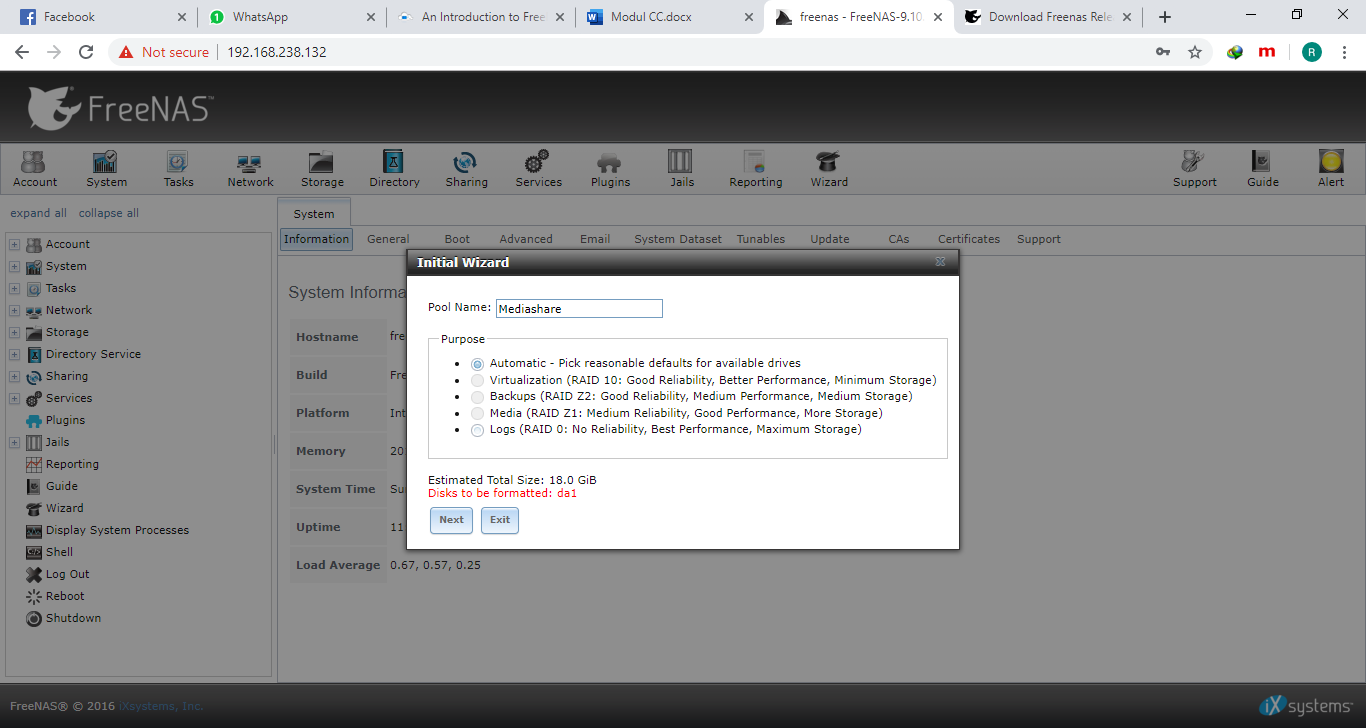
Input alamat IP pada URL bar browser



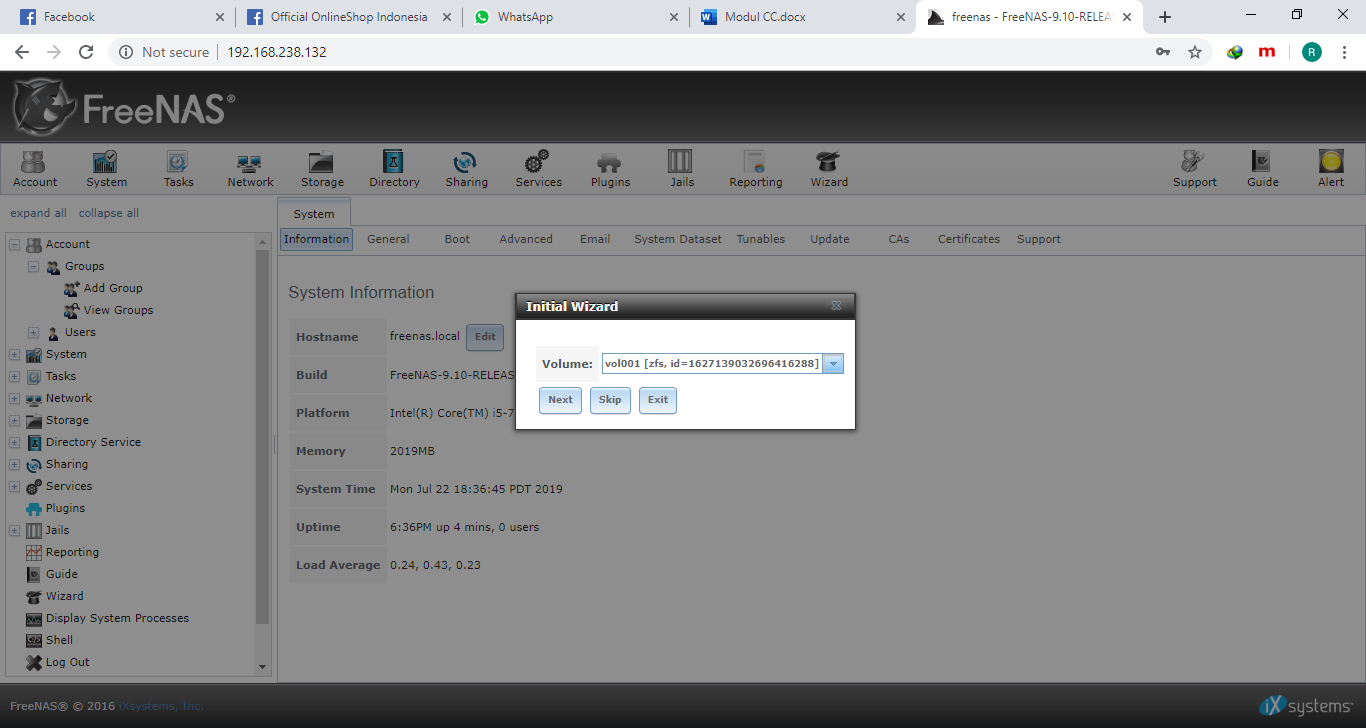
Pada tampilan awal akan muncul Setup Wizard



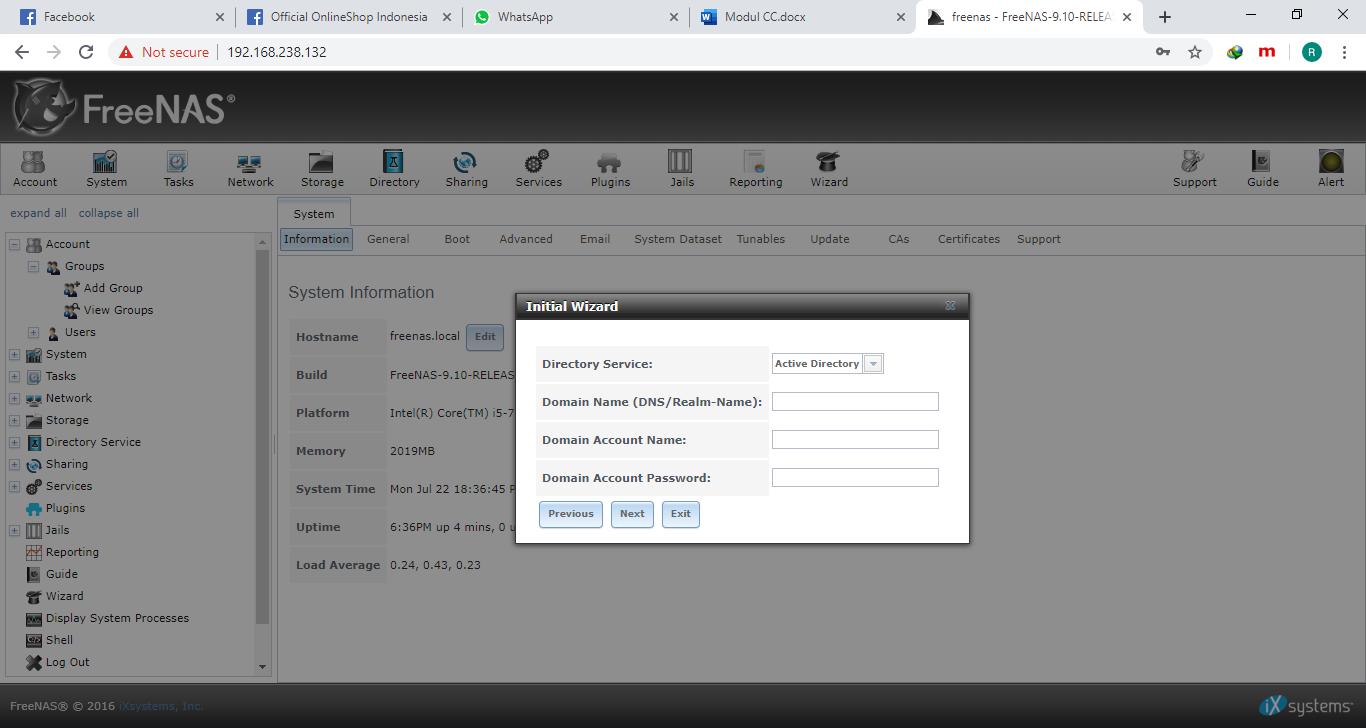
Membuat Pool name



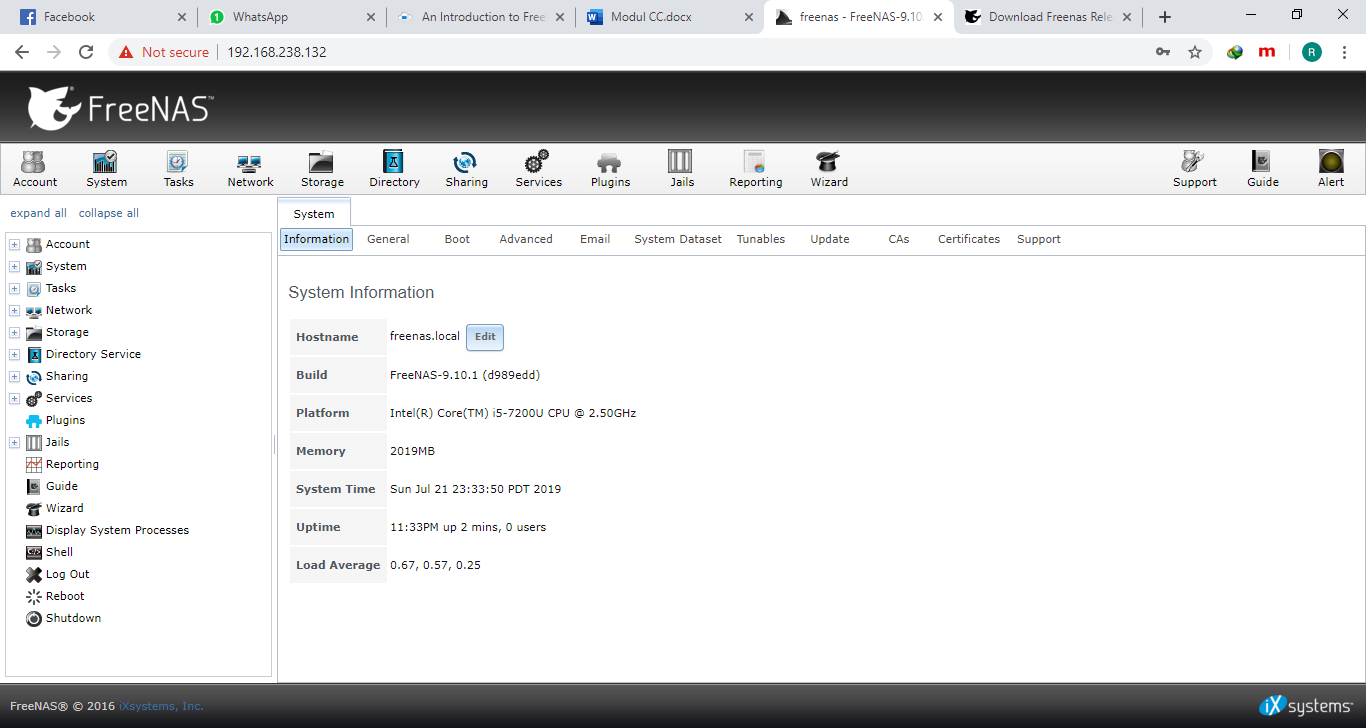
Initial wizard > volumes



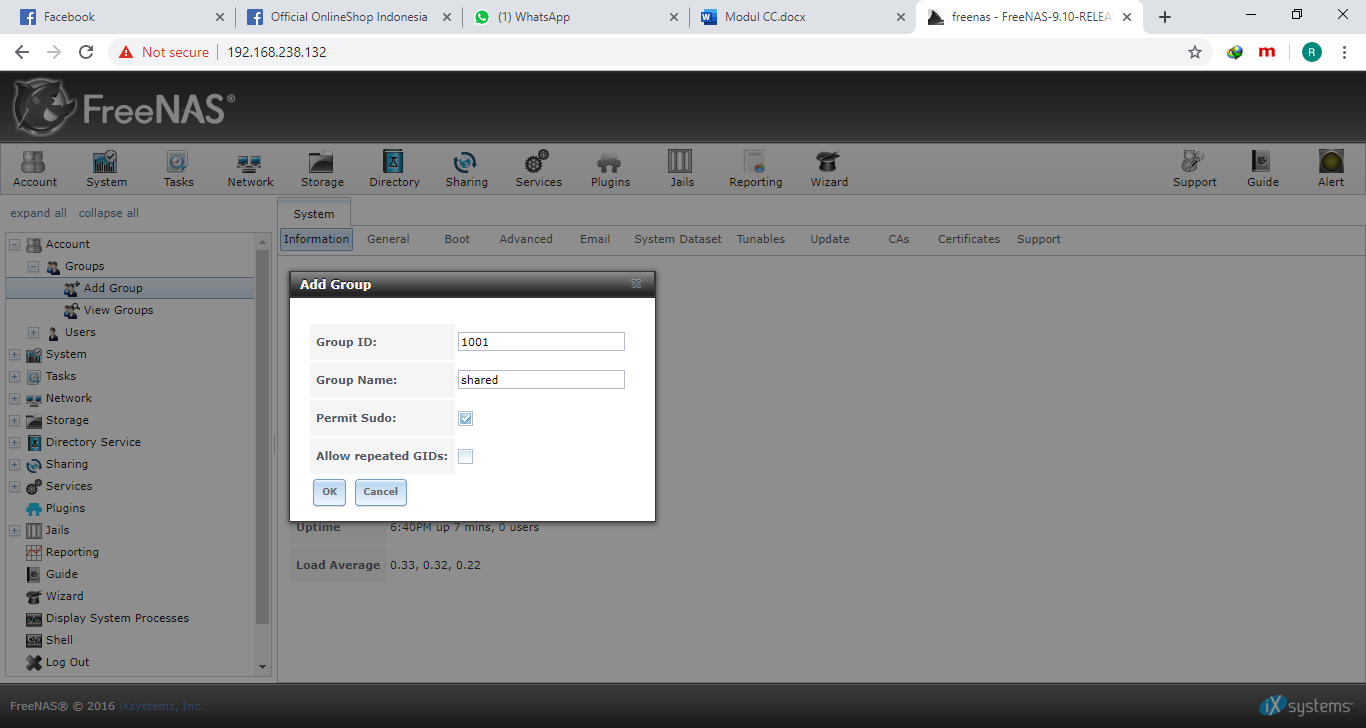
Initial wizard > Directory service



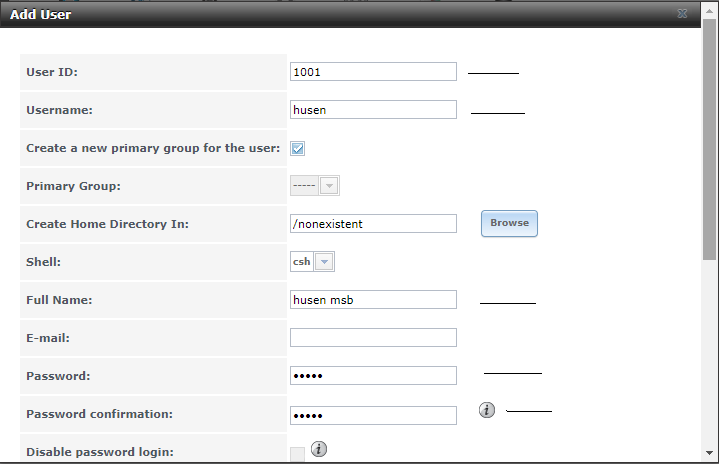
Tampilan awal FreeNAS 9.10 Release



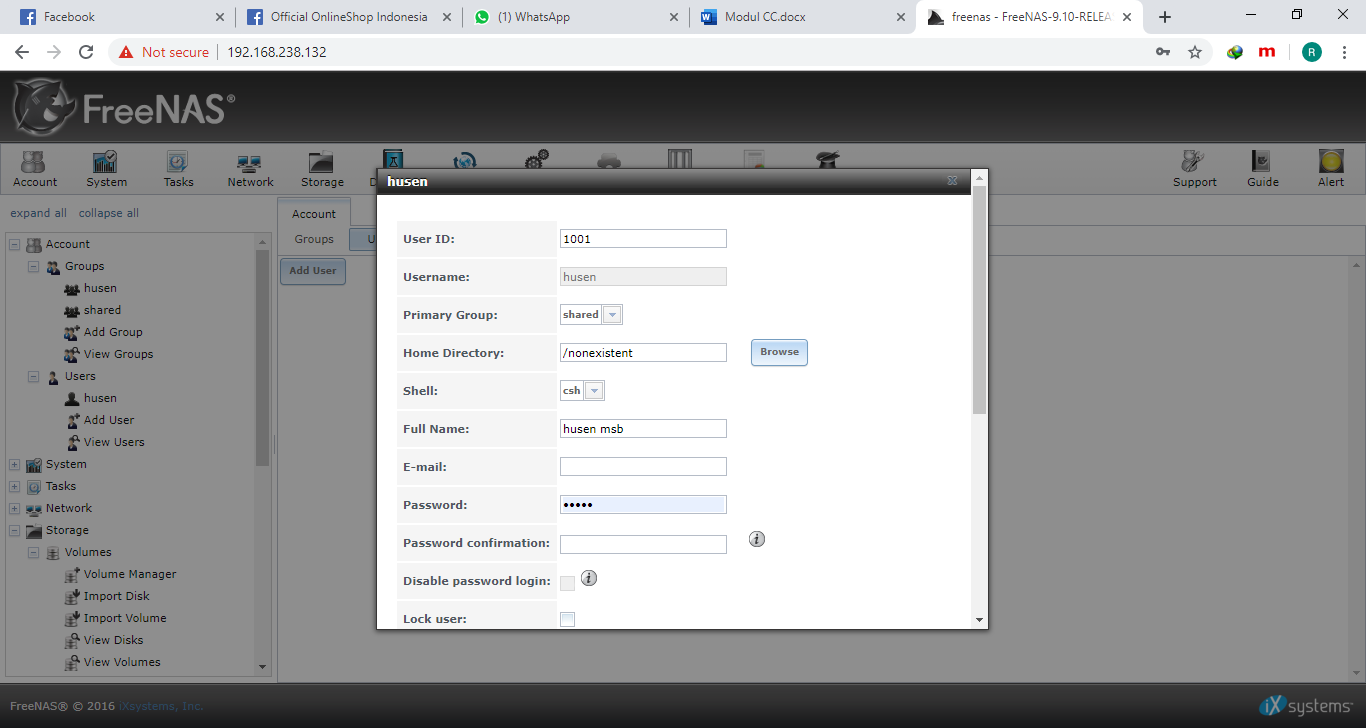
Tambah group pada menu **Account** > **Add Group**



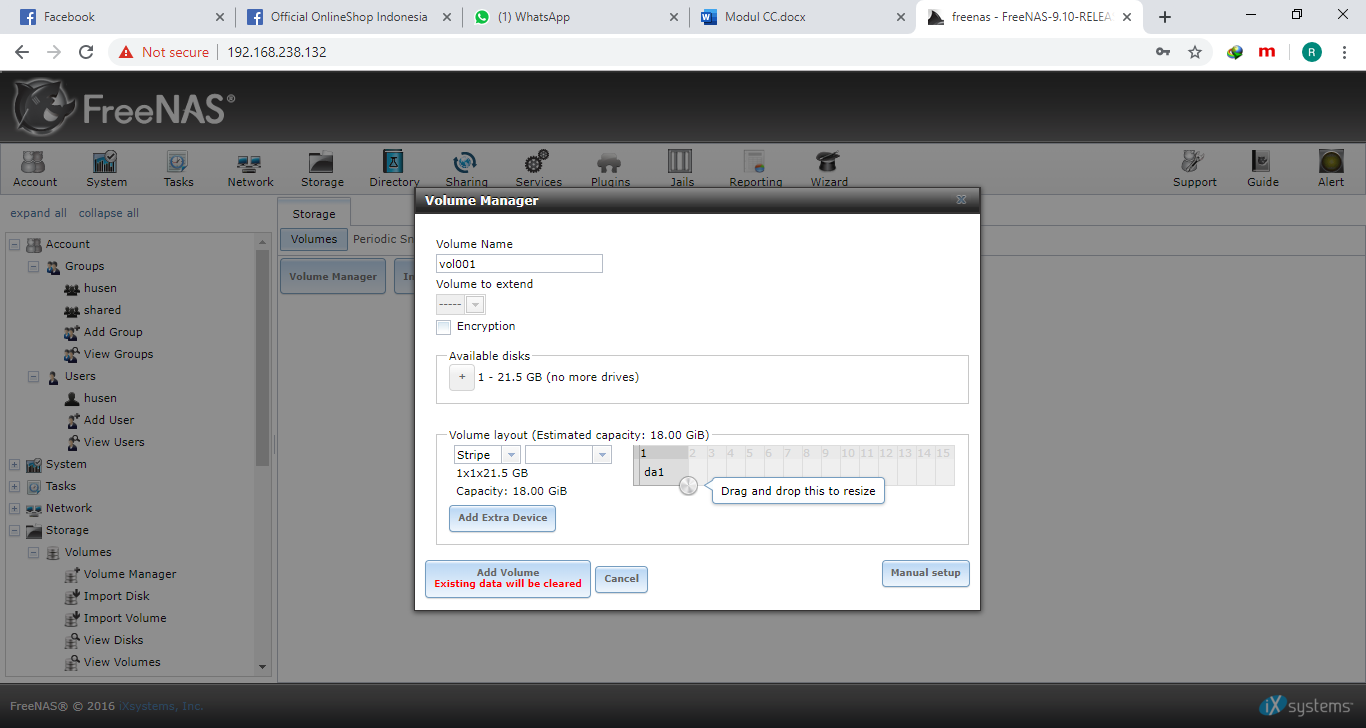
Tambah user pada menu **Account** > **Add User**



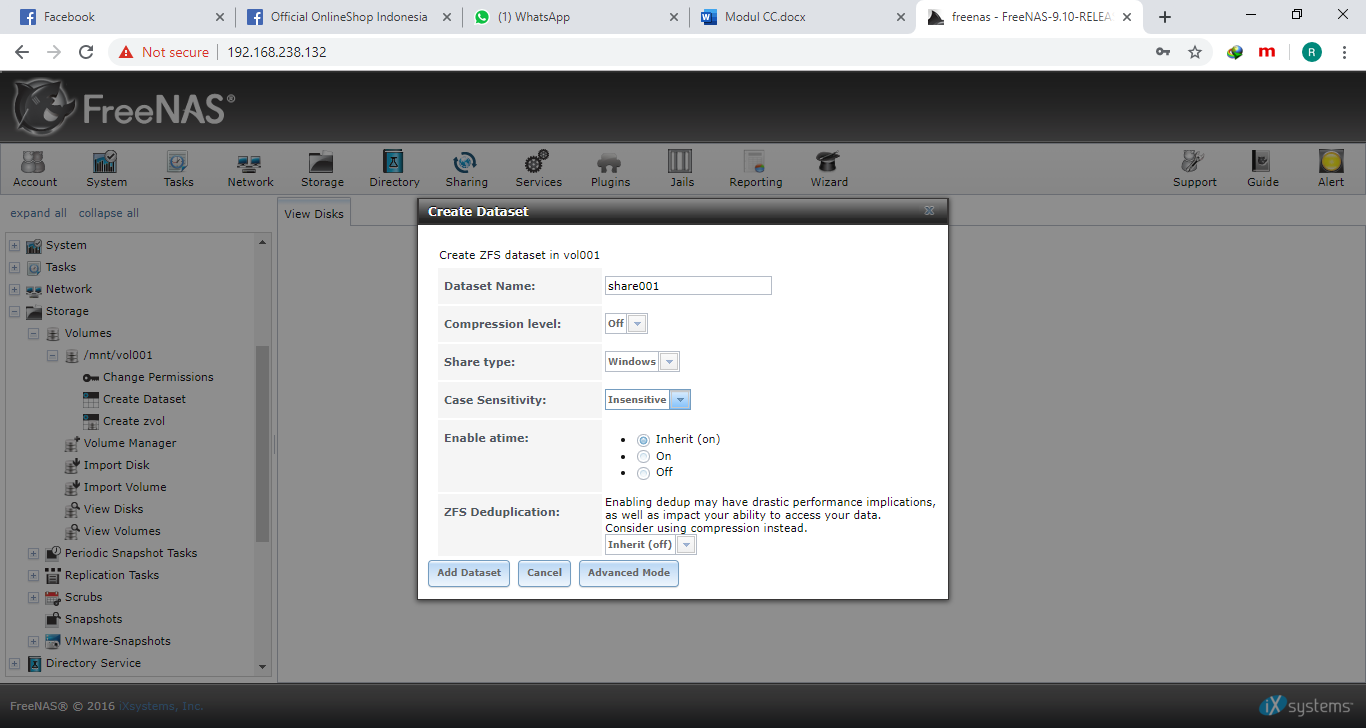
Edit User > Primary Group & Home Directory Mode



Volume manager



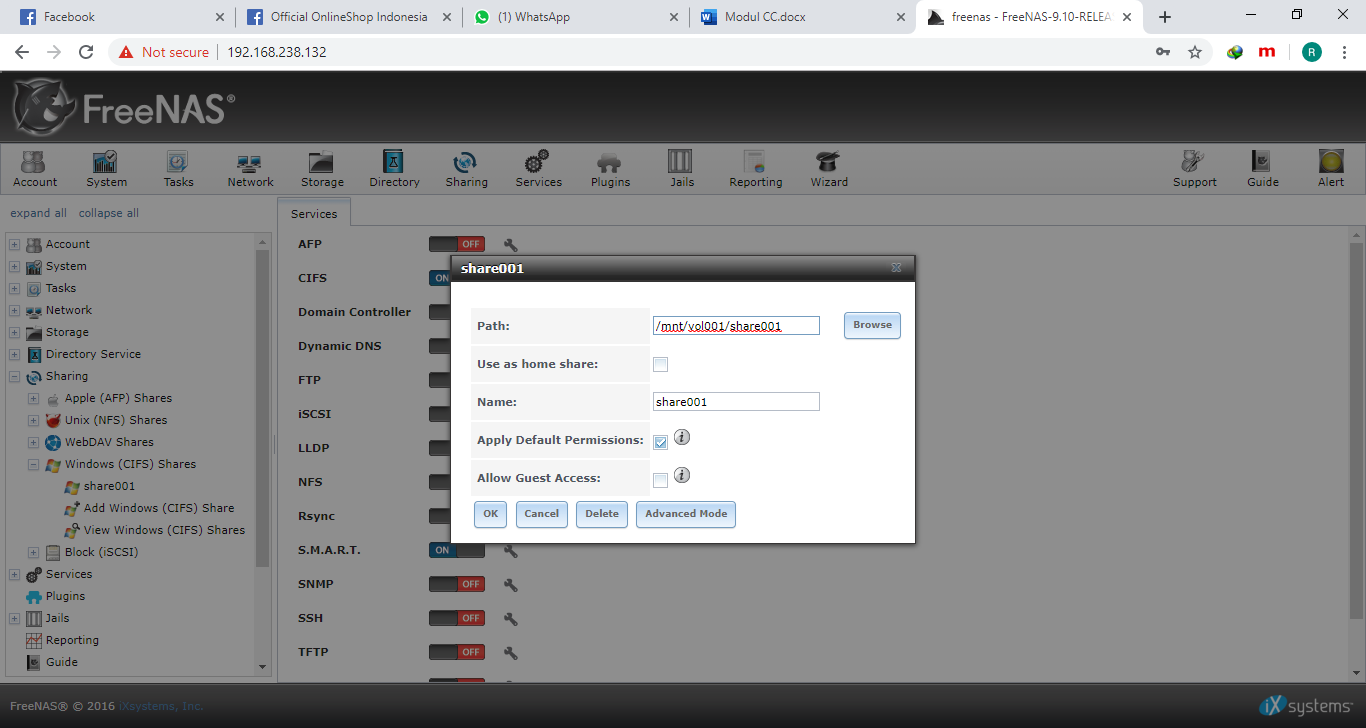
Create Dataset pada Volumes



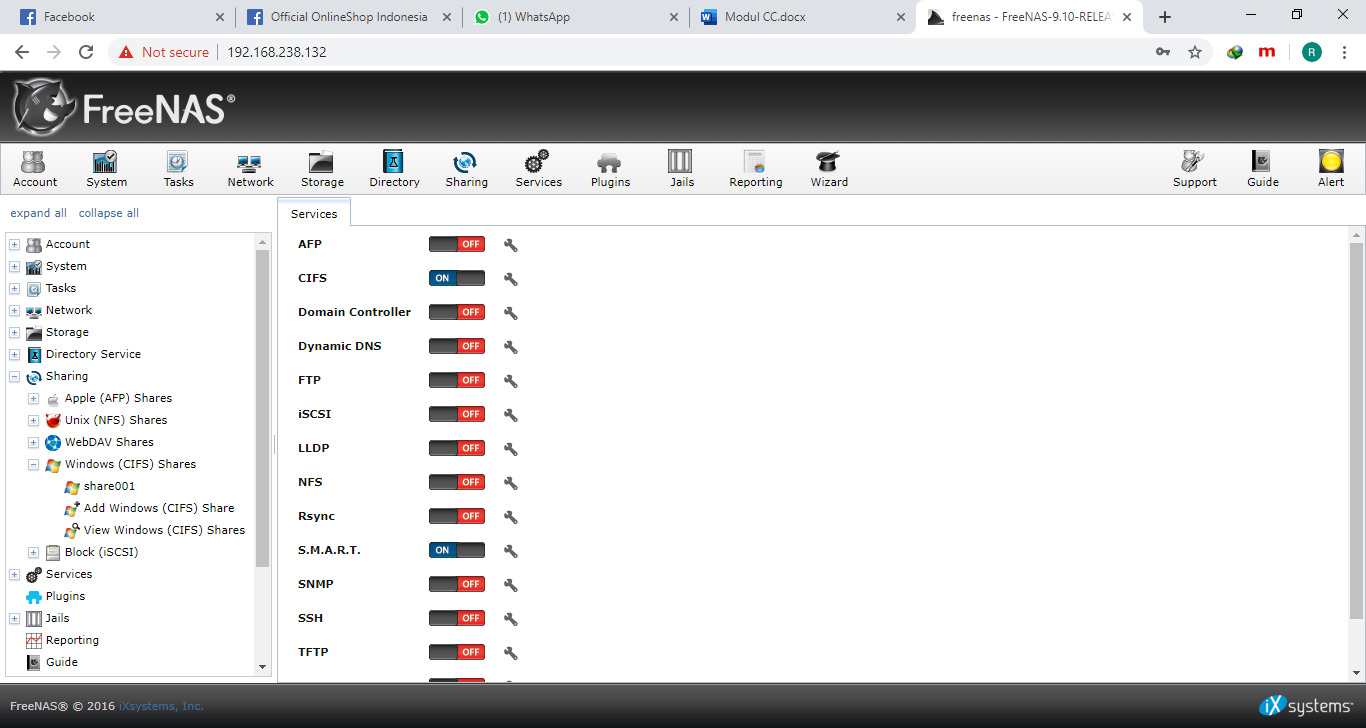
Change Permissions pada volumes



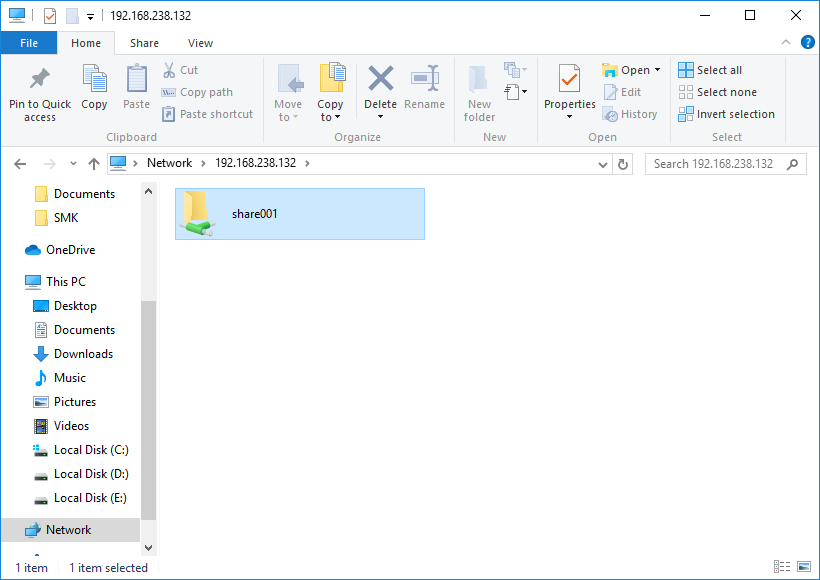
Add windows (CIFS) share



Pastikan services pada **CIFS** sudah **ON**



Cek pada windows explorer > Ip Address FreeNAS



|  |
| --- |
| Modul 2 instalasi dan konfigurasi hypervisor |

## Tujuan Praktikum

* Memahami cara menginstall VMware vSphere Hypervisor 6.7
* Melakukan konfigurasi awal pada VMware vSphere Hypervisor 6.7

## Alokasi Waktu

* 1 x pertemuan = 120 menit

## Dasar Teori

### VMware vSphere Hypervisor

VMware vSphere Hypervisor adalah sebuah teknik virtualisasi yang memungkinkan untuk mengelola sumberdaya komputasi dalam pengelolaan infrastruktur TI. Disebut teknik virtualisasi karena sistem operasi (OS) yang digunakan bukan merupakan sebuah sistem OS yang sebenarnya melainkan sebuah *virtual machine* (VM). Hypervisor mampu menjalankan dan mengatur beberapa OS secara bersamaan pada satu host. Sehingga Hypervisor juga disebut sebagai *Virtual Machine Management* (VMM).

### Virtualisasi

Secara umum virtualisasi oleh Hypervisor dibedakan menjadi 3 (tiga) jenis, yaitu *partial virtualization, full virtualization,* dan *paravirtualization.*

#### Partial Virtualization

Merupakan suatu teknik virtualisasi atau simulasi perangkat keras yang mem-virtual-kan sebagian dari perangkat keras, sehingga diperlukan modifikasi sebagian pada OS atu perangkat lunak *guest*.

Contoh: Compatible Time-Sharing System (CTTS) pada IBM M44/44X.

#### Full Virtualization

Merupakan teknik virtualisasi yang mem-virtual-kan semua bagian perangkat keras untuk digunakan oleh *guest*. Sering juga disebut dengan simulasi lengkap sehingga OS pada *guest* tidak perlu dimodifikasi. Selain itu, semua *guest* juga mendapatkan sebuah *virtual hardware* dan semua *virtual hardware* dikelola oleh Hypervisor.

Contoh: Vmware, virtualbox, Parallels Desktop, dll.

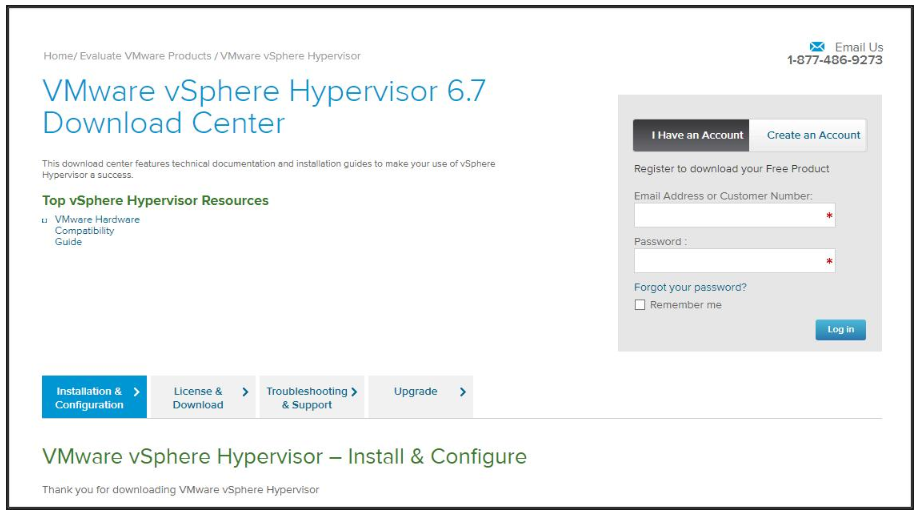
#### Paravirtualization

Berbeda dengan kedua jenis virtualisasi sebelumnya, *paravirtualization* ini tidak melakukan simulasi lengkap melainkan memanfaatkan API (para-API) untuk melayani *guest*. OS pada *guest* harus dimodifikasi untuk dapat berjalan pada lingkungan *paravirtualization*.

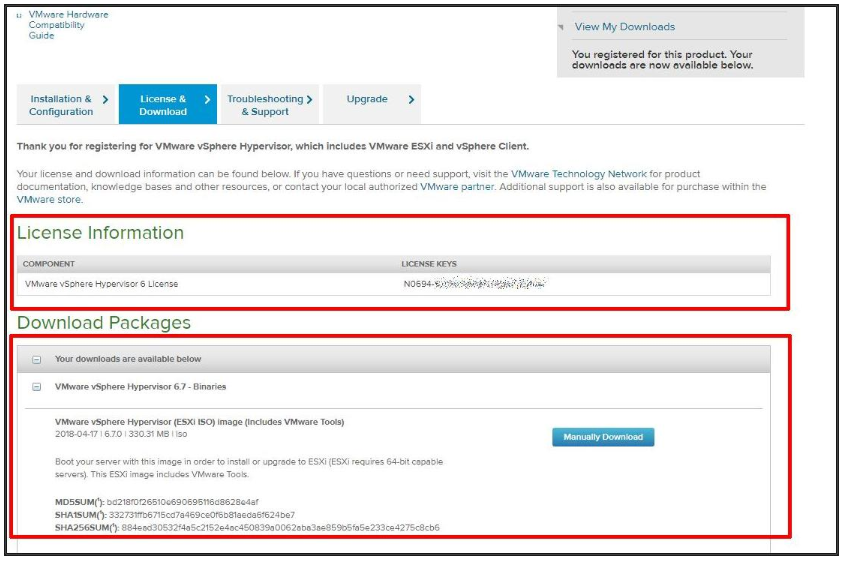
## Praktikum

### Download VMware vSphere Hypervisor

* Registrasi di VMware.com dan download VMware vSphere Hypervisor 6.7.



* Click **License & Download** untuk mendapatkan lisensi dan file download.

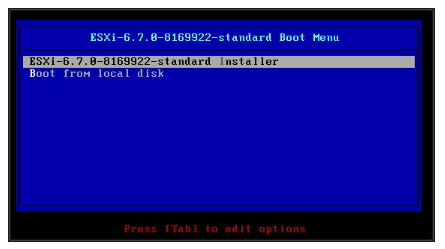


Perhatikan pada lisensi yang diberikan, **simpan di tempat yang aman** karena akan digunakan pada saat instalasi.

* Klik link download untuk melakukan proses download dan tunggu hingga selesai.
* Buatlah file image (\*.ISO) yang telah di download kedalam sebuah CD/DVD atau bisa menggunakan virtual CD/DVD.

### Instalasi VMware vSphere Hypervisor

* Setting BIOS BOOT sehingga CD/DVD drive yang berisikan image Hypervisor menjadi prioritas.



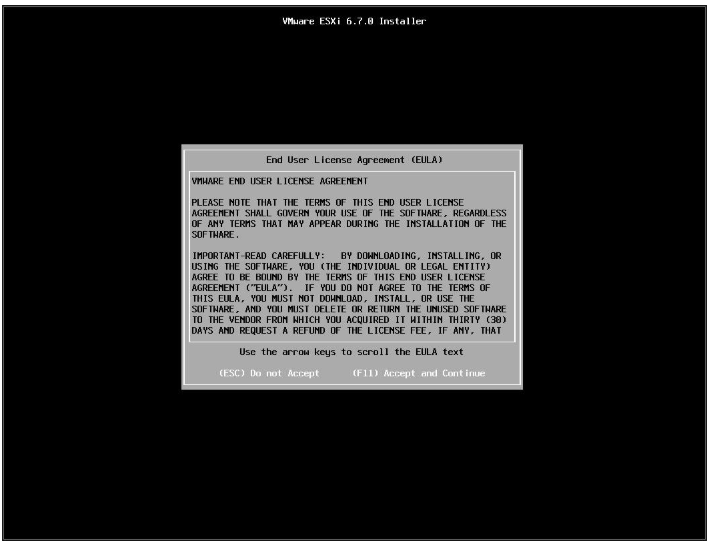
* Tampilan awal jika berhasil melakukan booting.



* Tekan **Enter** untuk melanjutkan proses instalasi



* Tekan **F11** untuk menyetujui *End User License Agreement* (EULA)

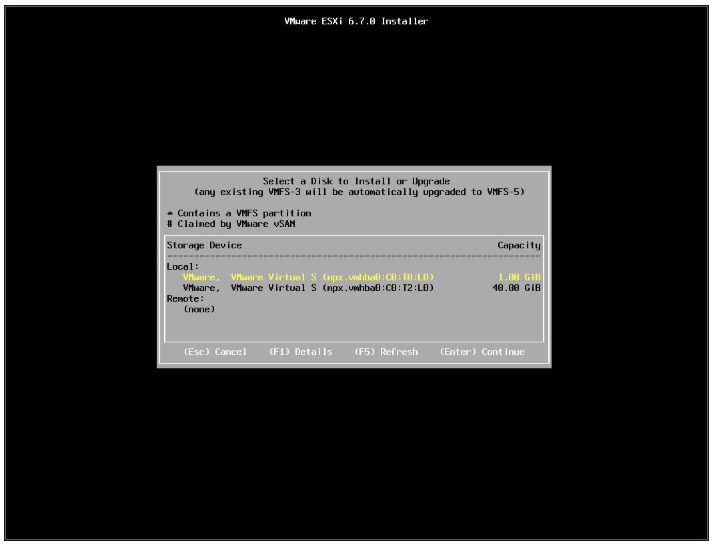


* ESXI installer akan memindai komputer untuk mencari tempat penyimpanan untuk melakukan proses install.



ESXI dapat diinstal pada storage manapun, baik itu berupa USB Flash Drive maupun Hard disk drive. Hal ini disebabkan karena ESXI hanya membutuhkan ruang sebesar 1GB untuk instalasi OS.

* Pilih lokasi penyimpanan atau lokasi instalasi ESXI.



Untuk tempat penyimpanan yang digunakan pada praktikum ini, gunakan tempat penyimpanan lokal (***local storage***). Kemudian tekan **Enter** untuk melanjutkan proses instalasi.

* Pilih jenis keyboard yang akan digunakan.

Ada banyak pilihan keyboard yang tersedia pada menu pilih keyboard. Sebaiknya memilih keyboard yang sesuai dengan yang anda gunakan, supaya ketika mengoperasikan sistem operasi akan menjadi lebih mudah.

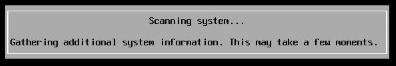
Untuk jenis keyboard yang digunakan pada pratikum ini adalah **United Kingdom**. Setelah memilih keyboard, tekan **Enter** untuk melanjutkan proses instalasi.



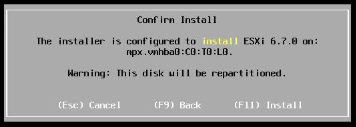
* Langkah Selanjutnya adalah membuat **root password.** Password yang dibuat sebaiknya disimpan baik-baik karena **tidak ada mode password recovery**. Sehingga jika password root hilang, maka satu-satunya cara adalah instal ulang ESXI. Setelah memasukkan dan mengkonfirmasi password, maka tekan **Enter** untuk melanjutkan proses instalasi.



* Setelah mengkonfirmasi password, ESXI akan melakukan pemindaian sistem dengan tampilan berikut.

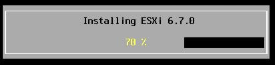


* Gunakan tombol **F11** untuk konfirmasi instalasi.



Setelah ESXI mengkonfirmasi bahwa instalasi akan dilakukan pada disk yang telah dipilih, semua data akan dihapus dan disk tersebut akan diformat dan dipartisi.

* Berikut adalah tampilan proses instalasi.



* Jika proses instalasi telah selesai, maka tampilan berikut akan muncul.

Untuk melanjutkan ke proses selanjutnya pilihnya salah satu dari menu yang disediakan, kemudian tekan **Enter.**



* Lepaskan tempat penyimpanan yang digunakan untuk membaca image instalasi ESXI sebelum sistem melakukan *reboot*. Berikut adalah tampilan sebelum sistem melakukan reboot.



### Konfigurasi VMware vSphere Hypervisor 6.7 (ESXi 6.7)

Setelah melakukan instalasi, langkah selanjutnya adalah melakukan konfigurasi supaya bisa digunakan dalam praktikum ini.

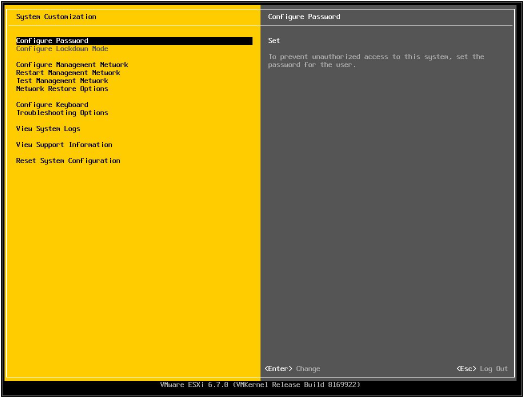
* Berikut adalah tampilan awal ketika pertama kali dinyalakan setelah diinstal.



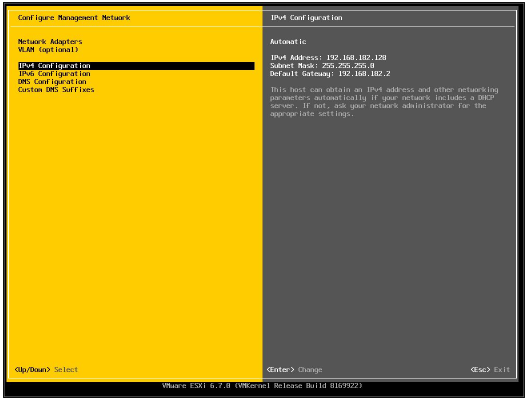
* Ketika dihidupkan, akan ada proses booting yang harus dilalui oleh server. Jika server sudah berhasil *boot,* maka akan menampilkan tampilan berikut. Dengan menampilkan spesifikasi sistem dan IPv4 untuk mengakses *host*.



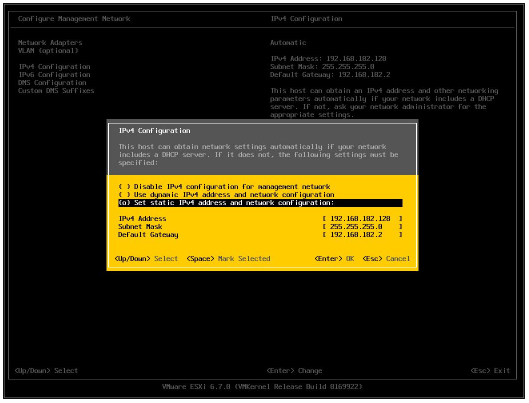
* Tekan **F2** untuk melakukan konfigurasi. Akan muncul tampilan seperti berikut.



* Pilih menu **Configure Management Network** untuk mengganti parameter jaringan yang disesuaikan dengan IP address yang dimiliki oleh komputer yang digunakan. Akan muncul tampilan berikut.



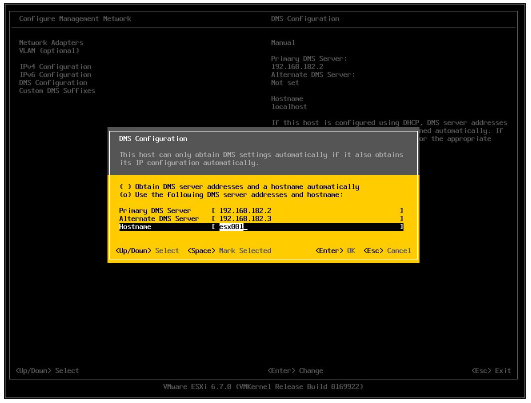
* Pilih menu **IPv4 Configuration** kemudian tekan **Enter** untuk mengkonfigurasi IP address pada komputer yang digunakan. Akan muncul tampilan seperti berikut.



* Pilih menu Set static IPv4 address and network configuration.
* Cocokkan IPv4 address pada server dengan IPv4 VMware adapter pada komputer yang digunakan. **IPv4 pada VMware** dijadikan sebagai **Default Gateway** pada server ESXI, sedangkan untuk **IPv4 pada konfigurasi** gunakan pengaturan berikut:
* 3 byte pertama sama dengan Default Gateway.
* 1 byte terakhir boleh diisi menggunakan angka 1-255.

Tekan **Enter** setelah mengisi IP address.

* Setelah lakukan konfigurasi IP adress, lakukan konfigurasi DNS dengan memilih menu **DNS Configuration** kemudian tekan **Enter**. Akan muncul tampilan seperti berikut.



Masukkan DNS sesuai dengan DNS komputer anda. Untuk **hostname**, boleh menggunakan format berikut: **cc\_ifupnvy\_<nomor>**.

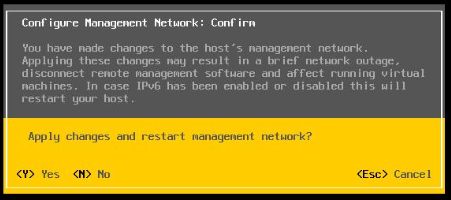
* Untuk mengkonfigurasi DNS suffix, pilih menu **Custom DNS Suffixes** kemudian tekan **Enter**. Akan muncul tampilan sebagai berikut.



Suffix dapat diubah sesuai keinginan. Pada praktikum ini gunakan format: **vmlab\_<nomor>.local**.

Tekan **Enter** untuk menyimpan pengaturan yang dilakukan.

* Setelah selesai mengkonfigurasi jaringan, tekan **Escape** untuk keluar. Akan muncul tampilan berikut. Tekan **Y** untuk menyimpan semua konfigurasi jaringan dan restart ulang pengaturan jaringan.



* Jika tahap diatas sudah selesai dilakukan, maka akan muncul tampilan sebagai berikut.



Dapat dilihat bahwa konfigurasi yang dibuat sebelumnya sudah tampil pada *console*.

|  |
| --- |
| modul 3 INSTALASI DAN PENGELOLAAN VIRTUAL MACHINE PADA VSPHERE ESXI |

Setelah selesai menginstall dan melakukan konfigurasi terhadap Hypervisor, berikutnya pada praktikum ini ada menunjukkan bagaimana cara membuat datastore dengan cara mengakses Hypervisor menggunakan web.

## Tujuan Praktikum

* Memahami cara menginstall Virtual Machine pada Vsphere ESXi
* Melakukan konfigurasi pada Virtual Machine pada Vsphere ESXi

## Alokasi Waktu

## Dasar Teori

### Pengertian Hypervisor

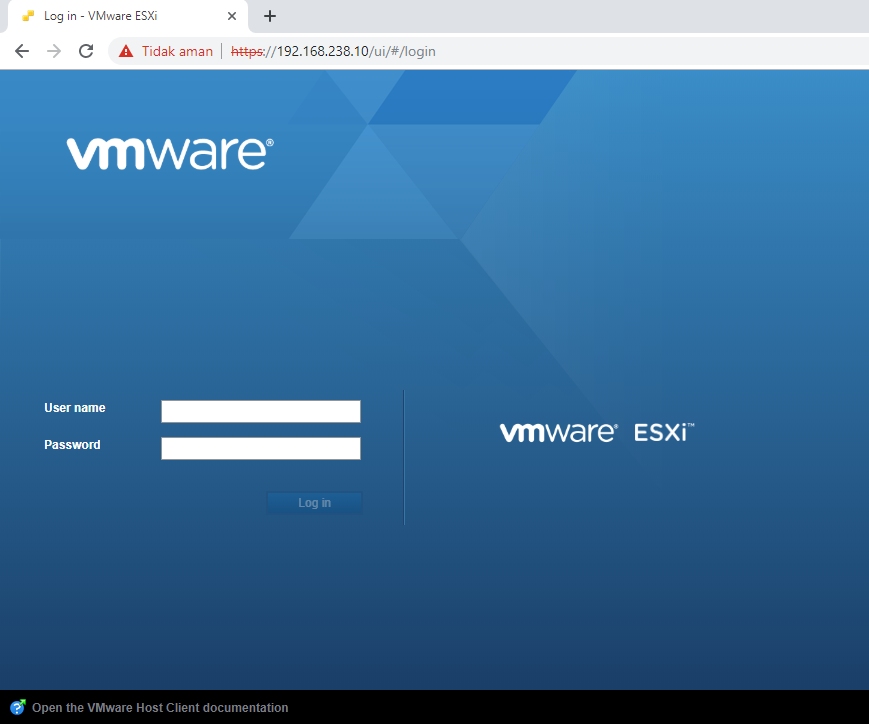
Hypervisor adalah sebuah teknik virtualisasi yang memungkinkan beberapa sistem operasi untuk berjalan bersamaan pada sebuah host. Dikatakan teknik virtualisasi karena sistem operasi yang ada bukanlah sebuah sistem operasi yang sesungguhnya, hanya sebuah virtual machine saja.

### Kelebihan menggunakan Hypervisor

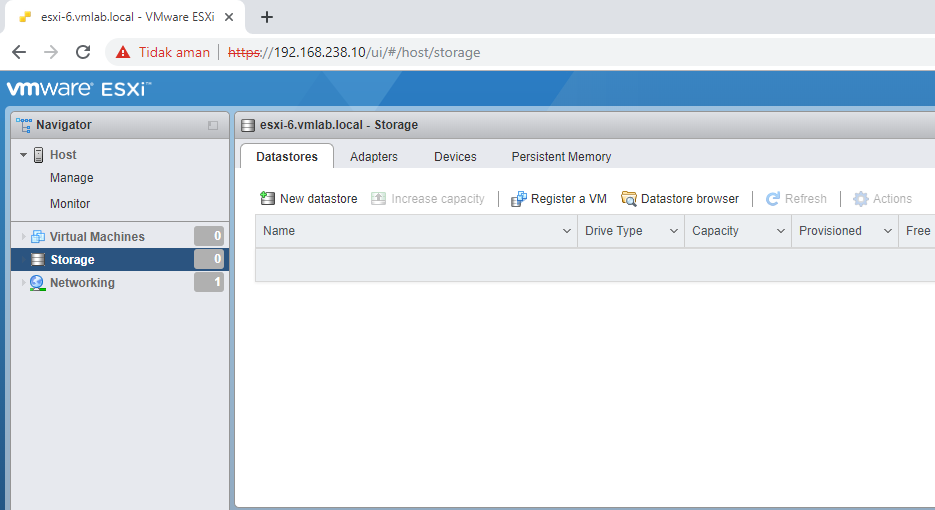
Kelebihan dari penggunaan Hypervisor adalah dapat mengatur penggunaan setiap sistem operasi sesuai dengan fungsinya masing-masing dan giliran penggunaannya sehingga Cloud Computing dapat berjalan dengan baik.

## Praktikum

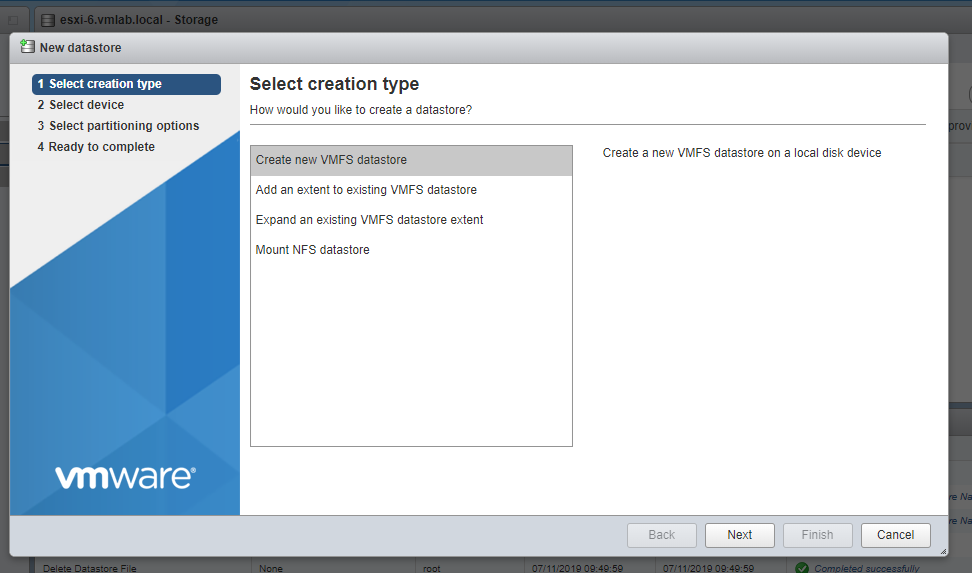
* Bukalah web browser dan ketikan IP Address yang sudah diatur sebelumnya.



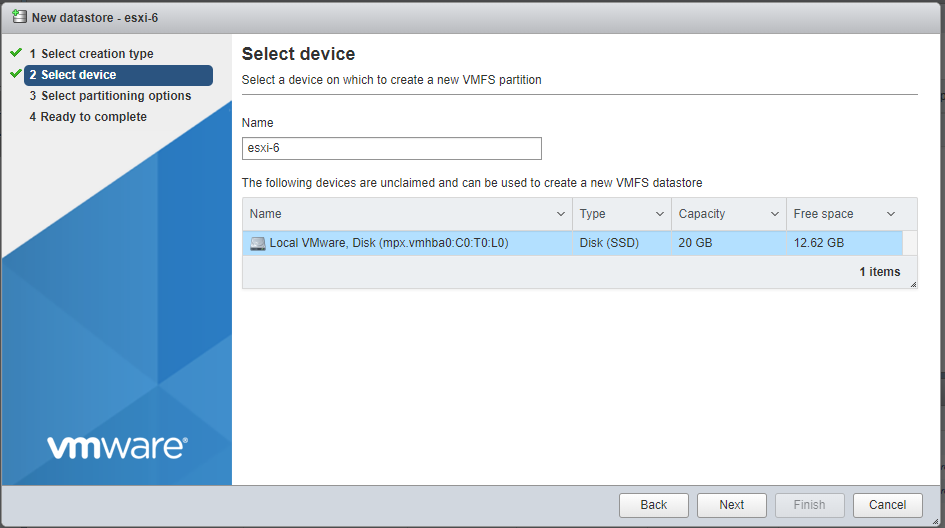
* Pilih menu ***Storage*** lalu klik ***New Datastore***.



* Pada window New Datastore pilih ***Create new VMFS datastore***



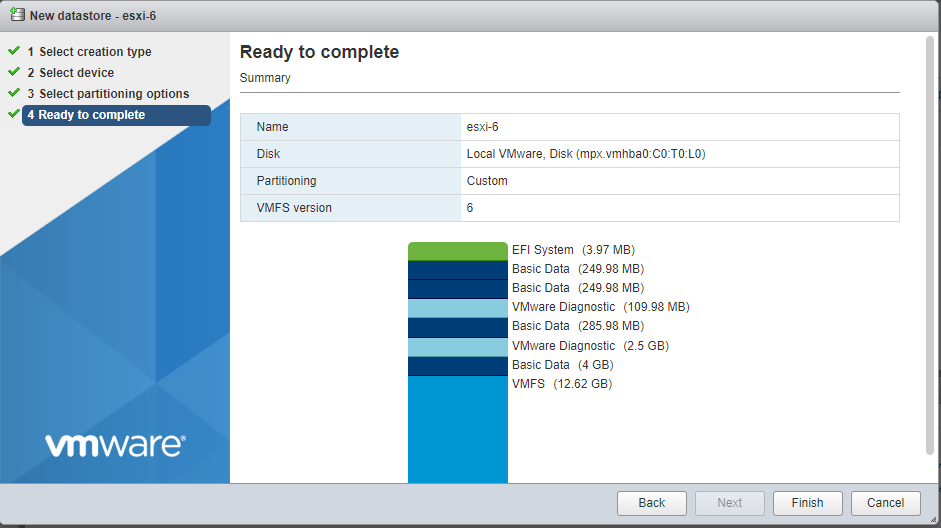
* Berikan Nama Datastore yang akan dibuat (hindari penggunaan spasi pada nama datastore)



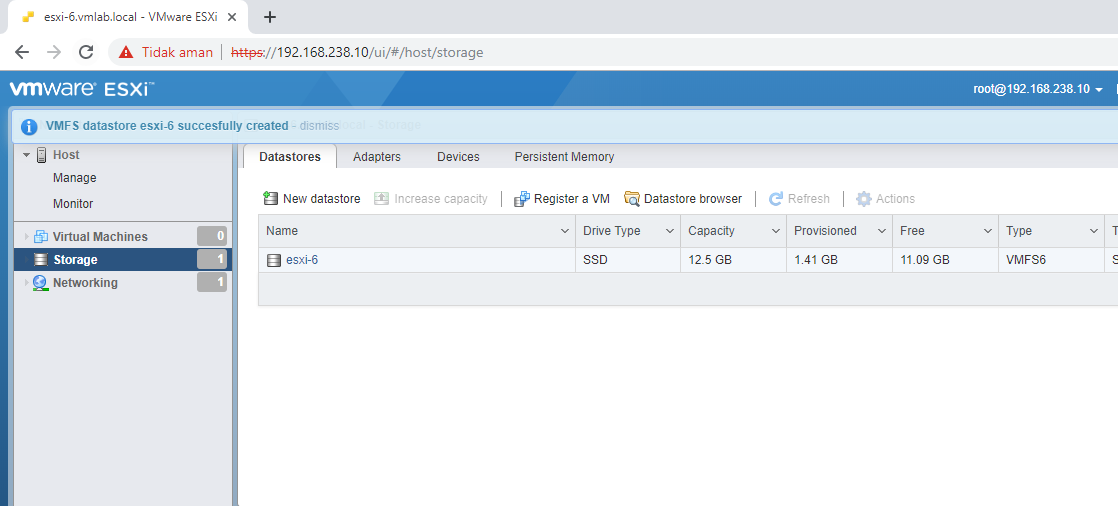
* Pada select partitioning options pilih custom lalu klik next



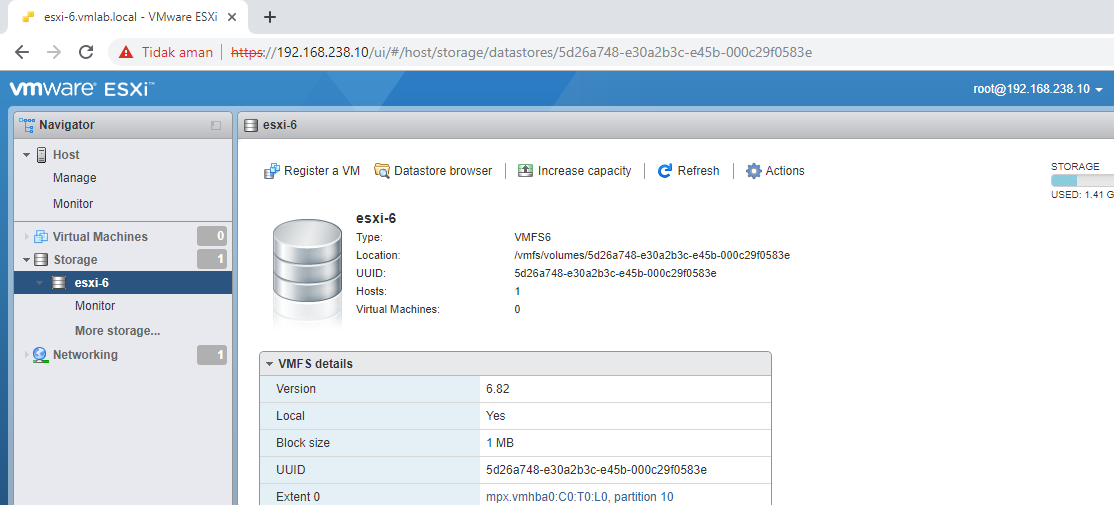
* Pada window terakhir pilih ***finish***



* Pada menu ***storage*** pilih nama datastore yang sudah dibuat



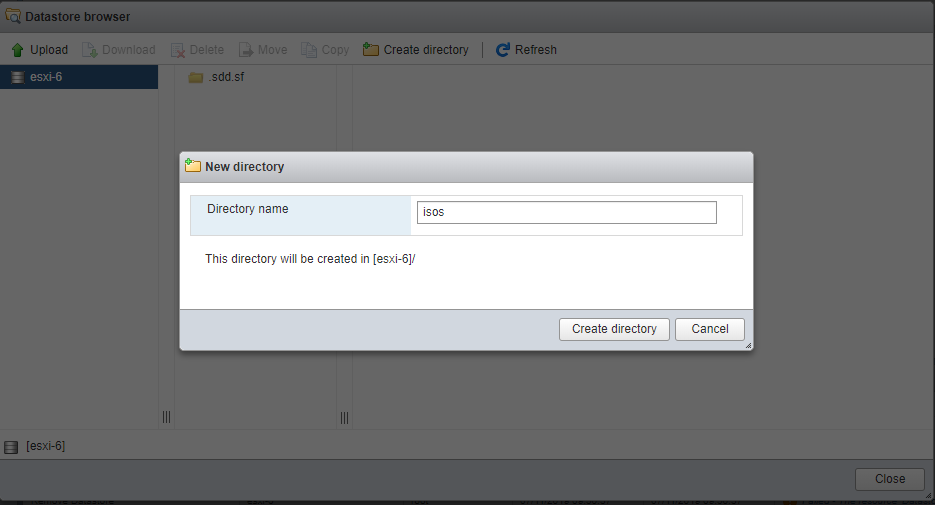
* Pilih menu ***datastore browser***



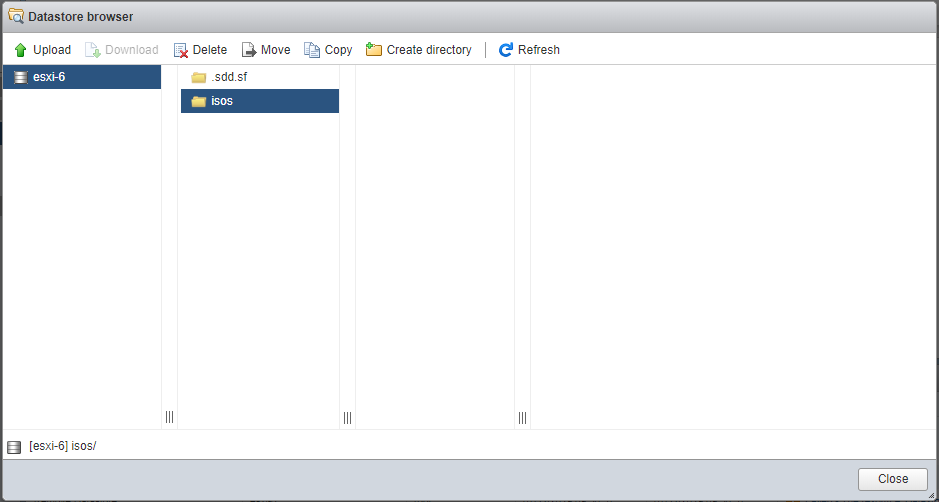
* Pilih ***create directory*** untuk membuat folder baru untuk menyimpan file ISO yang akan digunakan untuk meng-*install* virtual machine



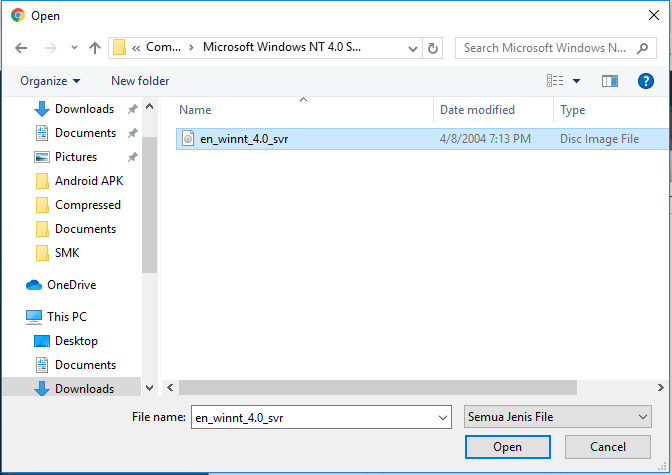
* Isikan nama directory lali pilik ***Create directory***



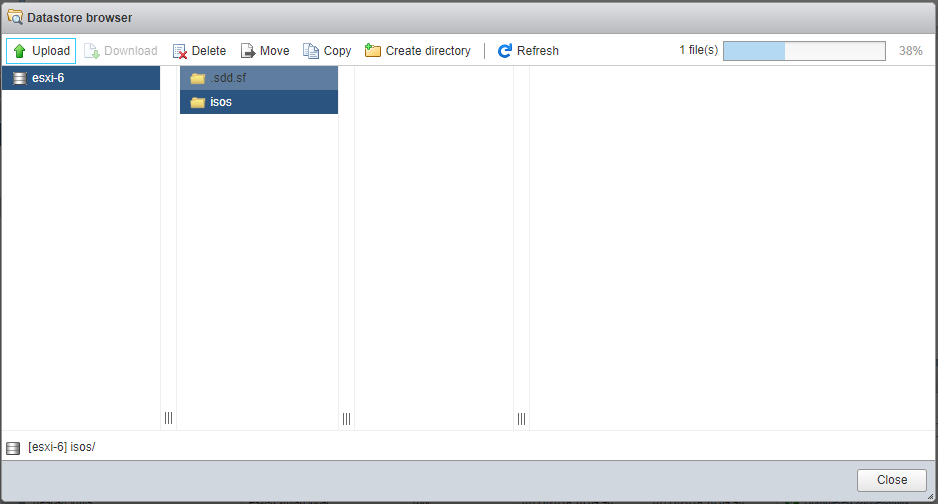
* Pilih folder yang sudah dibuat lalu klik menu ***Upload***



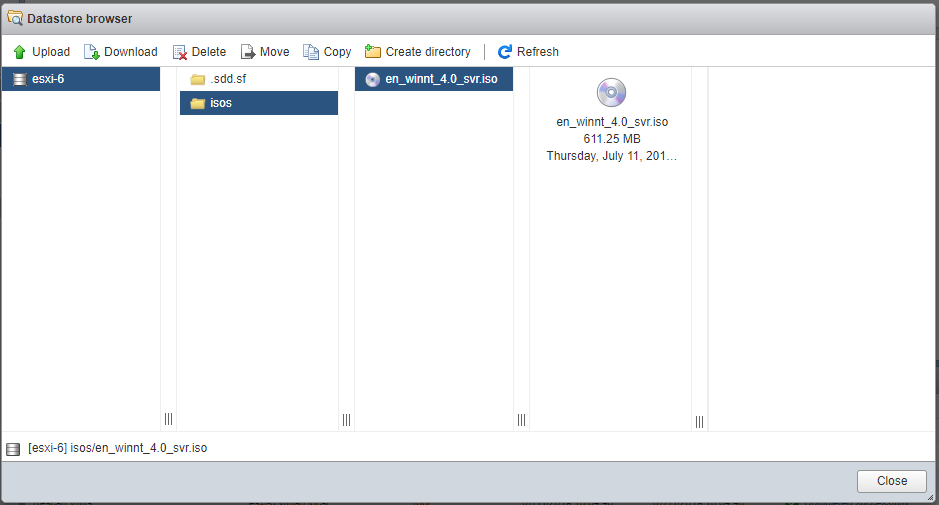
* Pada tampilan windows explorer pilih file ISO yang digunakan *install* virtual machine



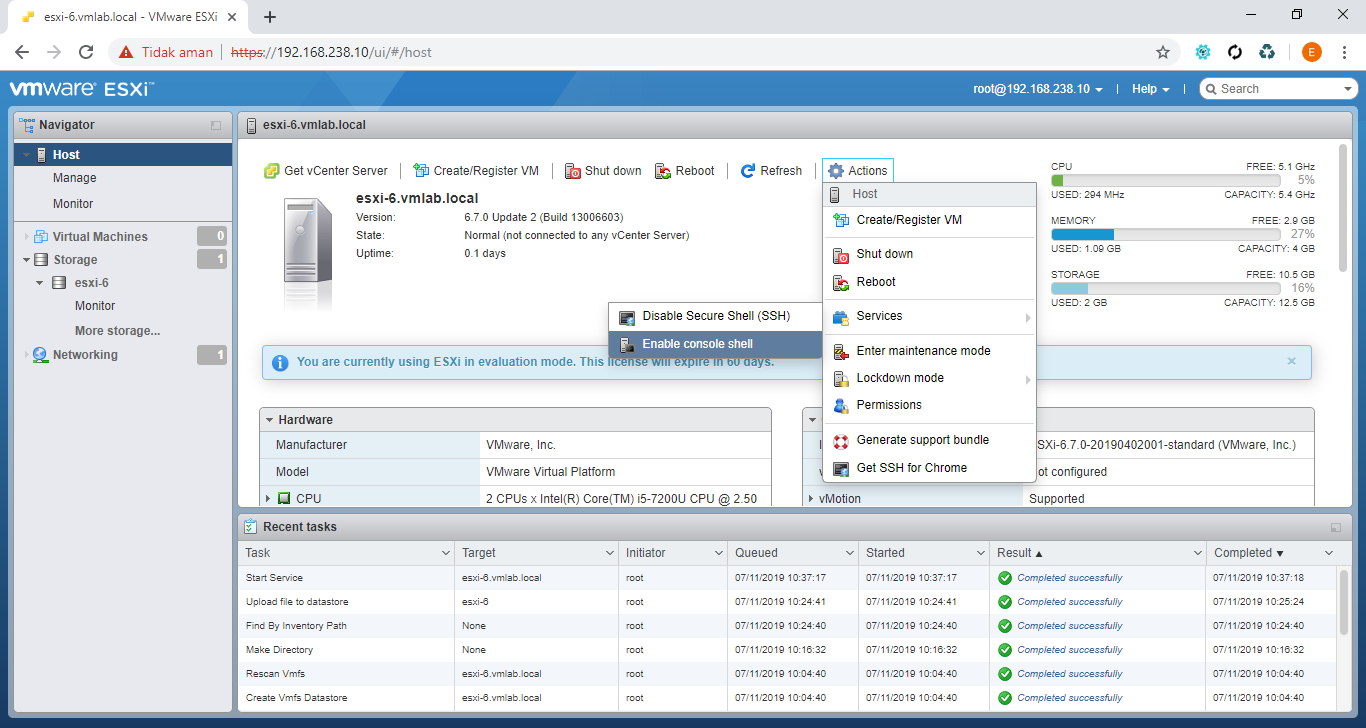
* Tunggu proses Upload sampai 100%



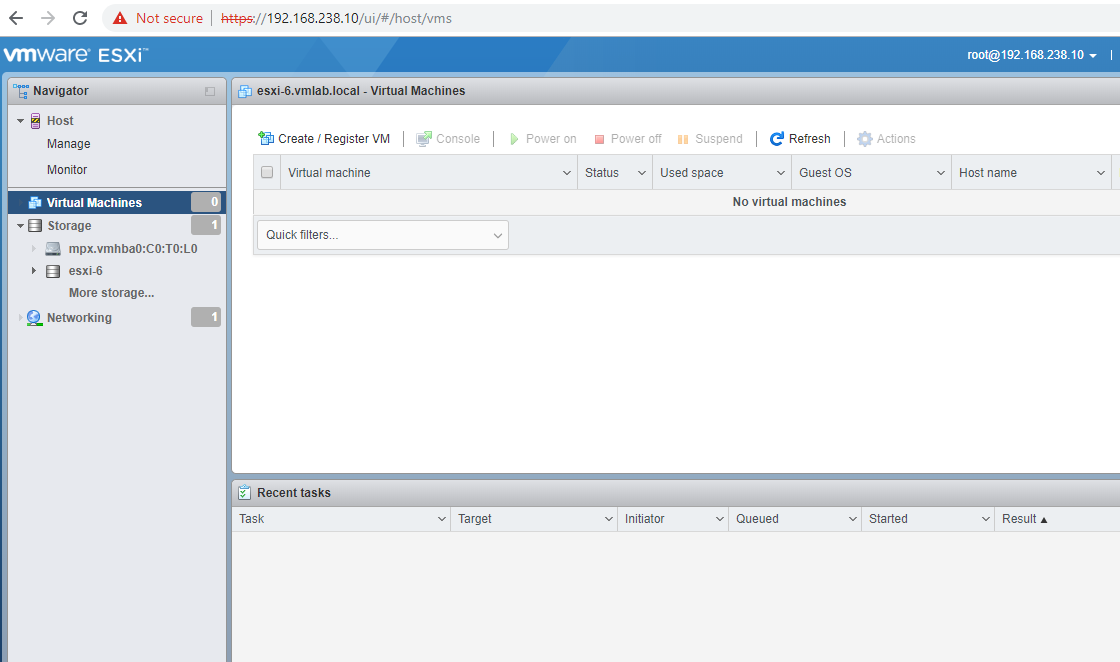
* Jika sudah berhasil upload, maka file ISO sudah masuk di kolom pada menu window



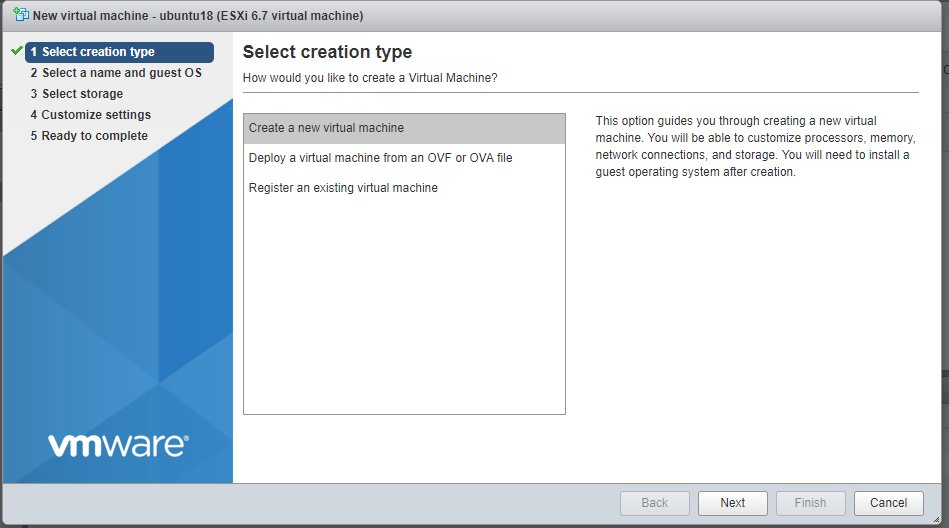
* Langkah selanjutnya adalah mengaktifkan SSH, pilih menu ***action*** > ***services*** > ***Enable Secure Shell (SSH)***



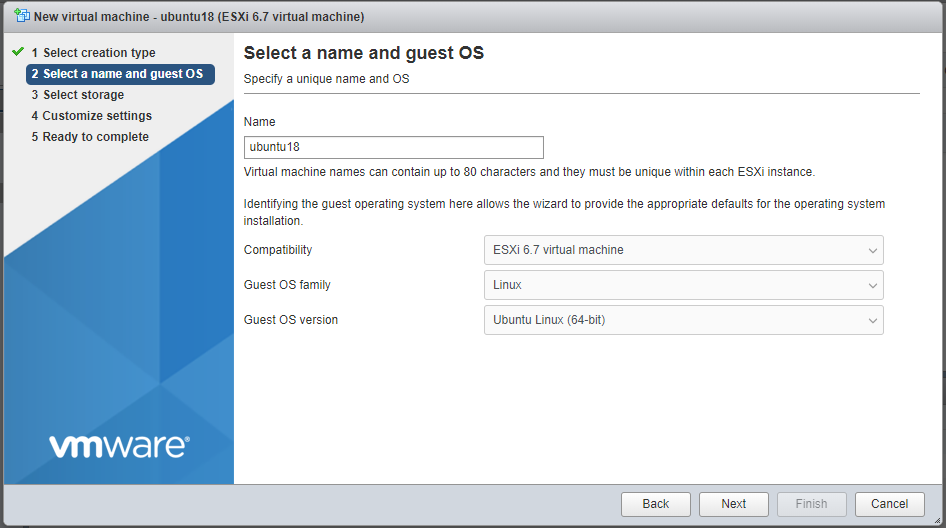
* Pilih menu ***Virtual Machine*** lalu pilih ***Create/Register VM***



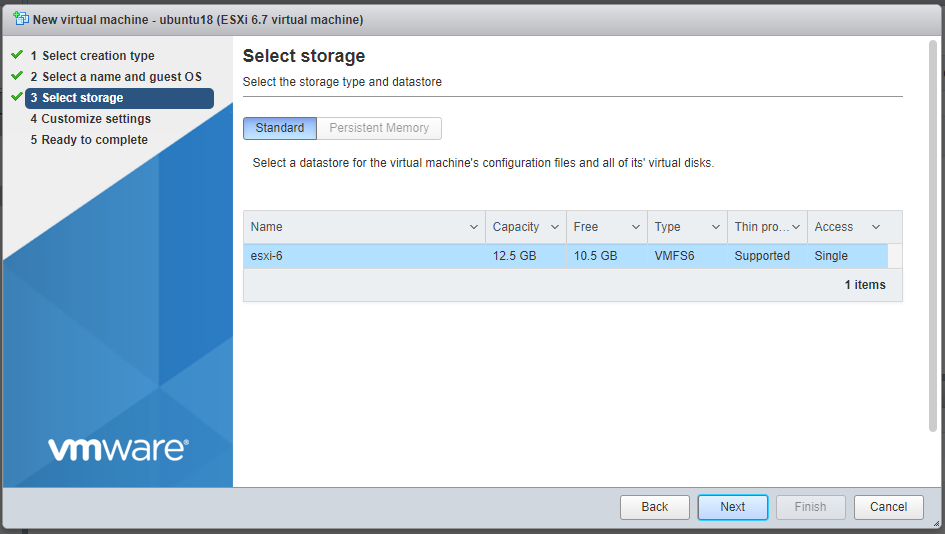
* Pilih ***Create a new virtual machine***



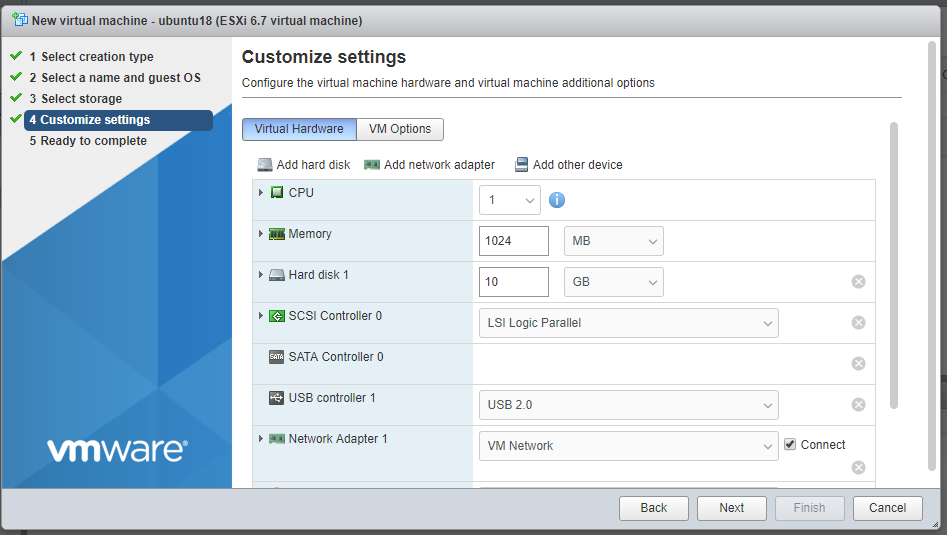
* Berikan nama virtual machine dan pilih guest OS sesuai dengan Sistem operasi yang akan di-*Install*



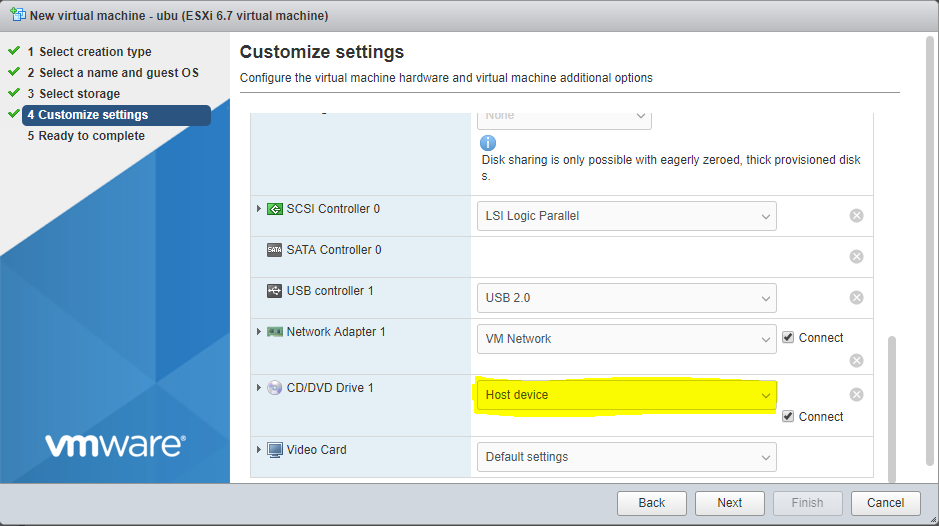
* Pilih ***storage*** lalu klik next



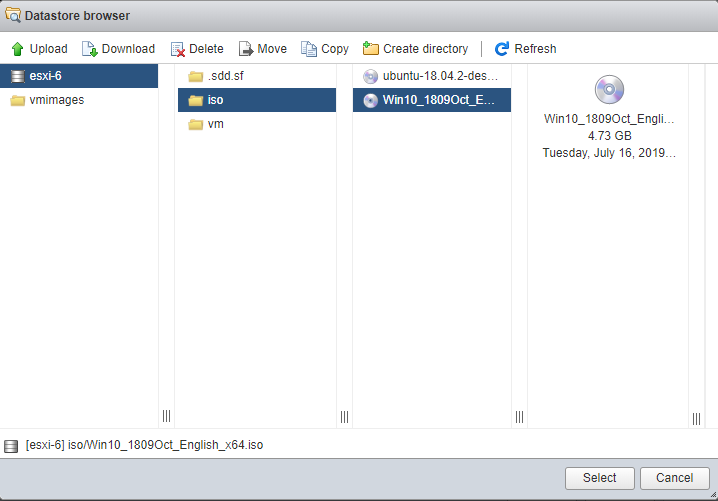
* Pilih ***storage*** lalu klik next



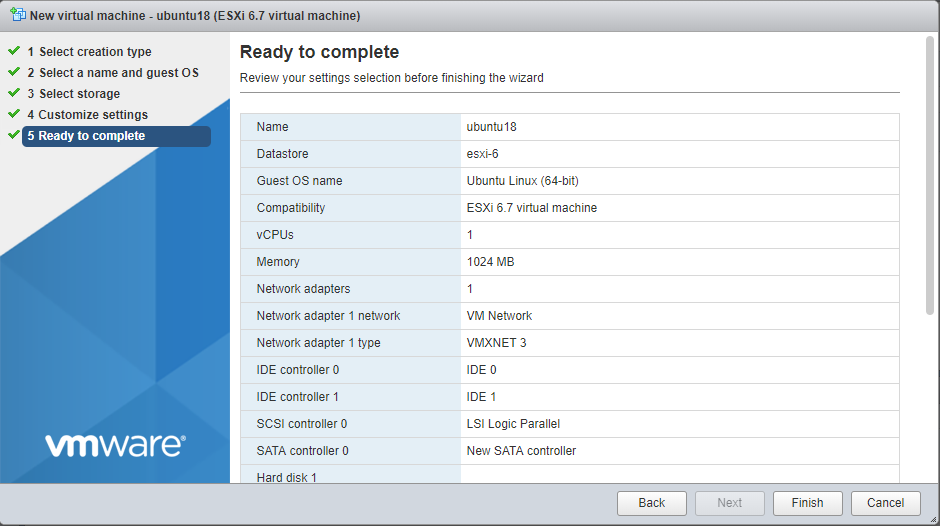
* Pilih ***Host Device*** lalu pilih ***Datastore ISO file***

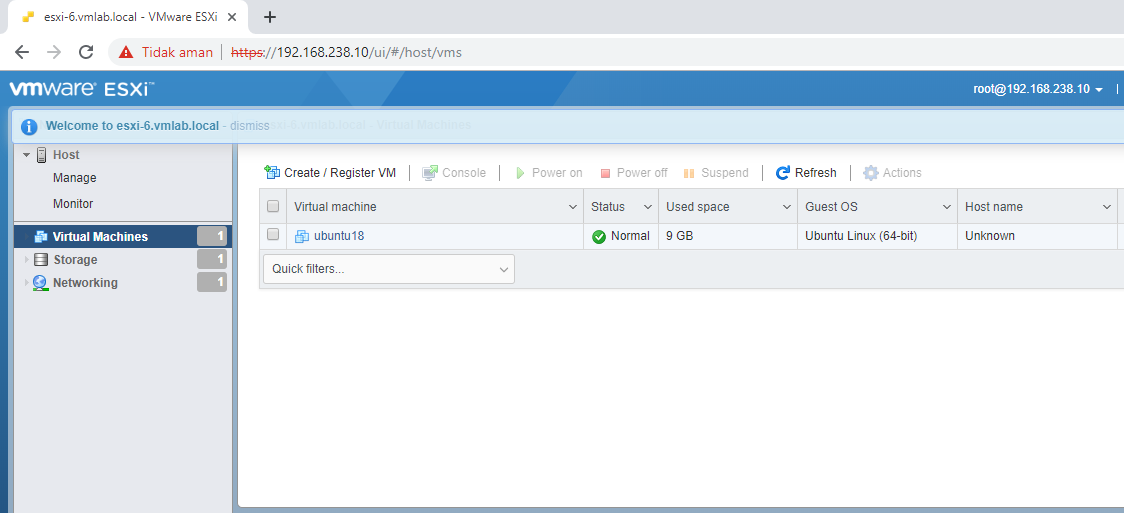


* Pada window Datastore browser pilih folder dan file ISO yang akan di-*install*



* Jika sudah selesai klik ***finish***





|  |
| --- |
| modul iv Instalasi dan Konfigurasi Layanan Hosting dengan LAMPP (SaaS) |

## Tujuan praktikum

* Memahami layanan hoting dengan LAMPP
* Mampu melakukan instalasi dan konfigurasi LAMPP

## Alokasi waktu

* 1 x pertemuan = 120 menit

## Dasar teori

Lampp merupakan singkatan dari Linux, Apache, MySQL, perl/php/python. Merupakan sebuah paket perangkat lunak bebas yang digunakan untuk menjalankan sebuah aplikasi secara lengkap.

Adapun beberapa komponen penting dalam LAMPP adalah beberapa hal berikut ini:

* \*\*htdoc adalah folder di mana Anda meletakkan file yang akan dijalankan, seperti file PHP, HTML dan script lainnya.
* \*\*phpMyAdmin adalah bagian untuk mengelola database MySQL yang dikomputer.  
  Untuk membukanya, membuka browser dan ketik alamat http: // localhost /phpMyAdmin, atau juga dapat menggunakan ip http://172.0.0.1/phpmyadmin maka secara otomatis halaman phpMyAdmin akan muncul.
* \*\*Control Panel yang berfungsi untuk mengelola layanan (service) XAMPP. Seperti stop service (berhenti), atau mulai (mulai).

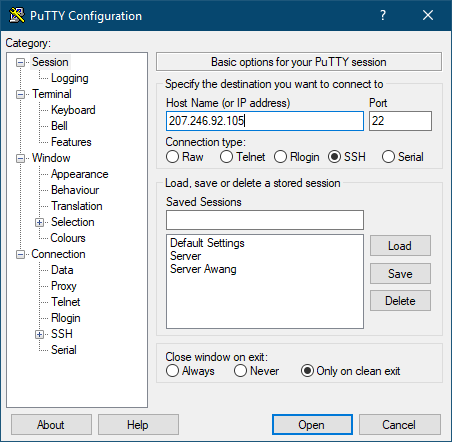
## Langkah Praktikum

Pada proses pembuatan layanan SaaS berbasiskan layanan Apache, PHP, MySQL, dan PHPMyAdmin, maka diperlukan tahapan sebagai berikut yang terbagi menjadi beberapa tahapan besar:

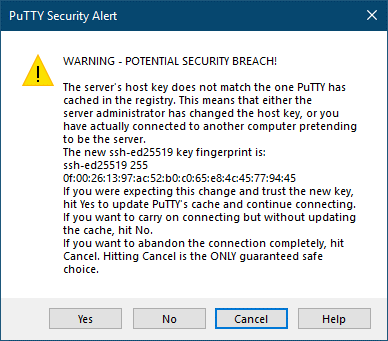
### Koneksi dengan Server

Sebelum memulai tahapan instalasi layanan SaaS, maka dilakukan koneksi terhadap server yang akan dikerjakan. Tahapannya adalah sebagai berikut ini:

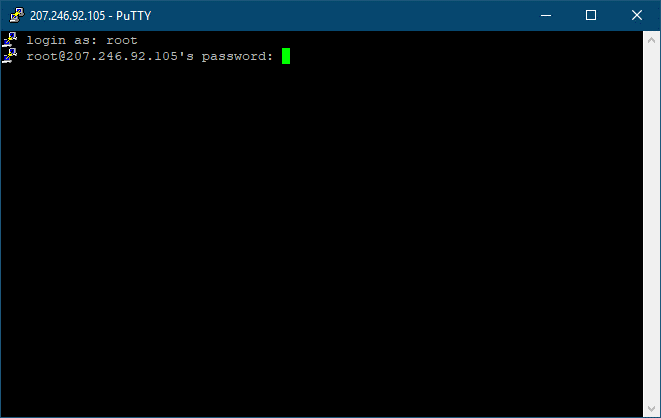
* Buka aplikasi PuTTY, masukkan IP server dan pilih mode koneksi SSH dengan *port* 22 kemudian lakukan *open connection* sebagaimana gambar berikut:



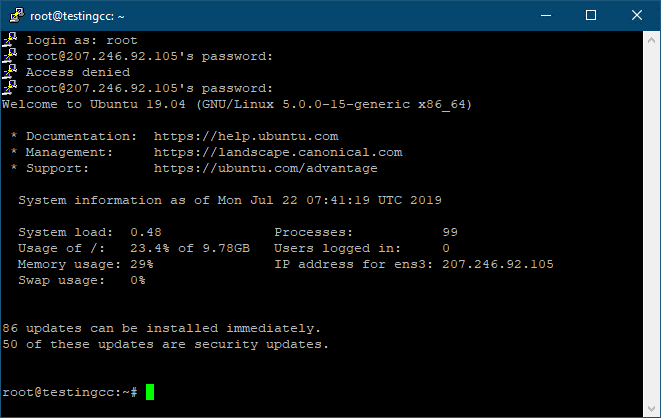
* Setelah itu, akan muncul tampilan *certificate approval*, pilih Yes atau Update bila muncul.



* Kemudian adalah jendela input username dan password. Isikan username dan password sesuai dengan yang telah dibuat pada bab sebelumnya (bab instalasi Linux Server). Berikut tampilannya:



* Bila proses login telah berhasil maka akan muncul welcome screen Linux sebagamana gambar berikut ini:

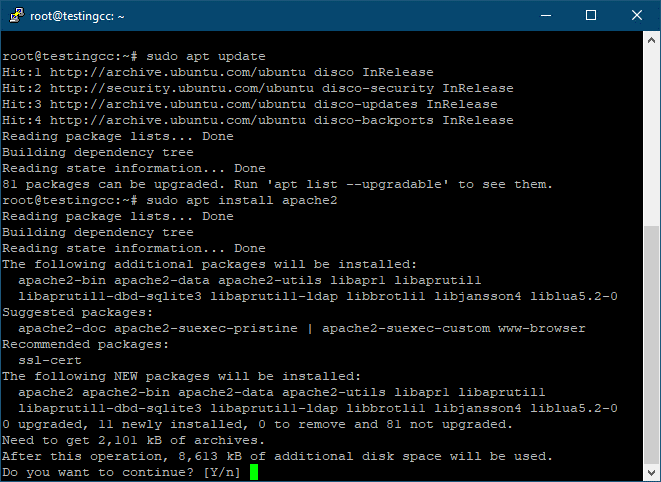


### Instalasi Apache dan Pengaturan Firewall

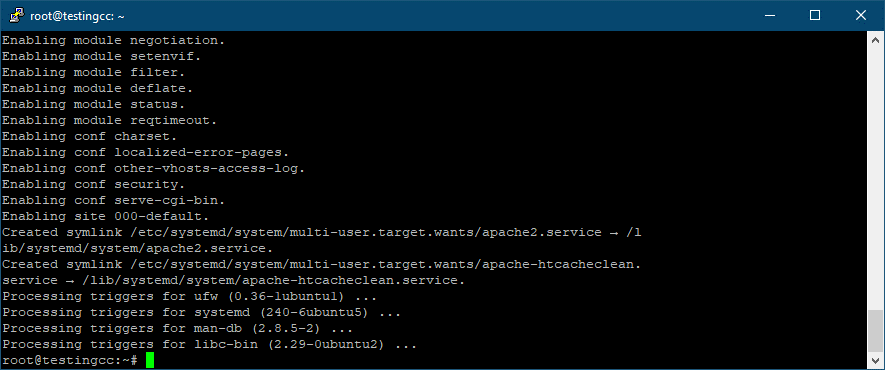
Apache Web Server merupakan web server yang pada umumnya digunakan di seluruh dunia. Selain terdokumentasi dengan baik dan dapat digunakan untuk berbagai hal, Apache Web Server cocok digunakan untuk model pelatihan di praktikum Cloud Computing ini. Berikut tahapan instalasi Apache dan pengaturan Firewall:

* Lakukan proses pembaruan package di Ubuntu dilanjutkan dengan instalasi package apache2 sebagaimana sintak perintah dan gambar berikut ini:

|  |
| --- |
| $ sudo apt update  $ sudo apt install apache2 |

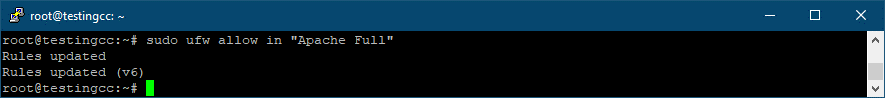


Pastikan proses instalasi telah berhasil dengan tidak adanya pesan error sebagaimana gambar berikut:



Kemudian dilanjutkan pengaturan Firewall dikarenakan pada Ubuntu versi baru memiliki pengaturan default bahwa Firewall otomatis aktif. Berikut sintak yang digunakan untuk mengatur Firewall sehingga lalu lintas data dari Apache Web Server dengan tipe protokol HTTP (port 80) dan HTTPS (port 443) diijinkan:

|  |
| --- |
| $ sudo ufw allow in "Apache Full" |



Setelah itu lakukan pengecekan dengan membuka browser pada Workstation yang digunakan saat ini. Buka alamat IP server dengan mengetikkan <http://IP_Server_Anda> Bila telah berhasil terbuka sempurna seperti pada gambar berikut ini, maka proses instalasi dan pengaturan Firewall telah berhasil dijalankan dengan sempurna:

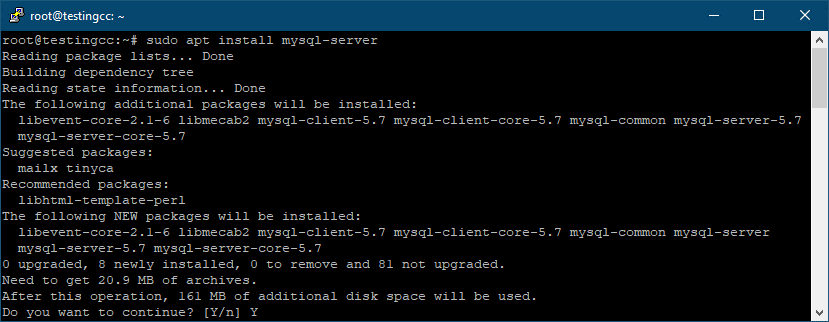


### Instalasi MySQL

Setelah Web Server dipastikan menyala (running), maka dilanjutkan dengan instalasi layanan MySQL sebagai basis layanan manajemen basis data. Berikut adalah langkah instalasinya:

* Lakukan proses pembaruan package di Ubuntu dilanjutkan dengan instalasi package apache2 sebagaimana sintak perintah dan gambar berikut ini:

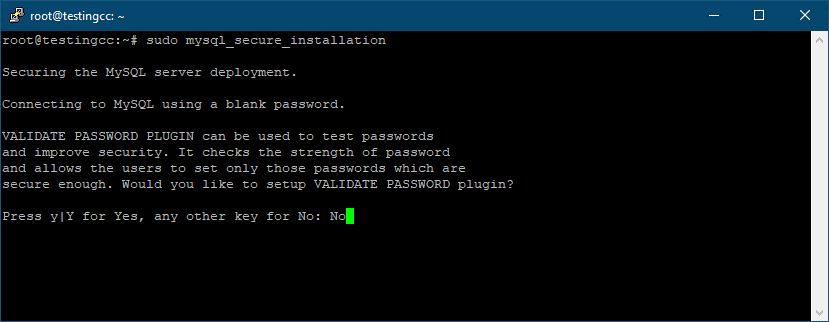
|  |
| --- |
| $ sudo apt install mysql-server |



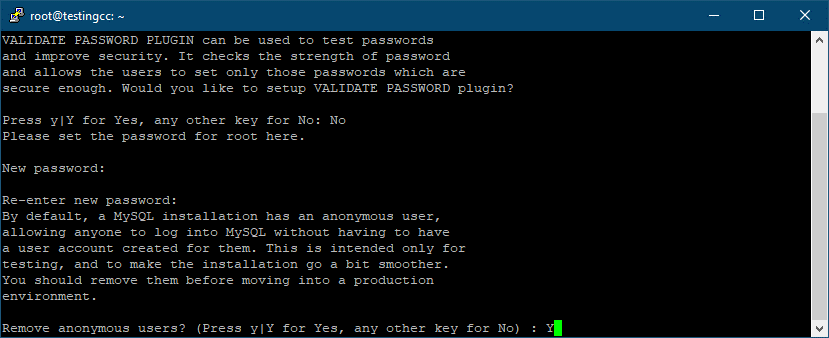
* Lanjutkan dengan proses pengaturan dasar pengamanan MySQL yang berupa users, kata sandi, hak akses, dan sebagainya. Berikut adalah sintaks wizard yang digunakan:

|  |
| --- |
| $ sudo mysql\_secure\_installation |

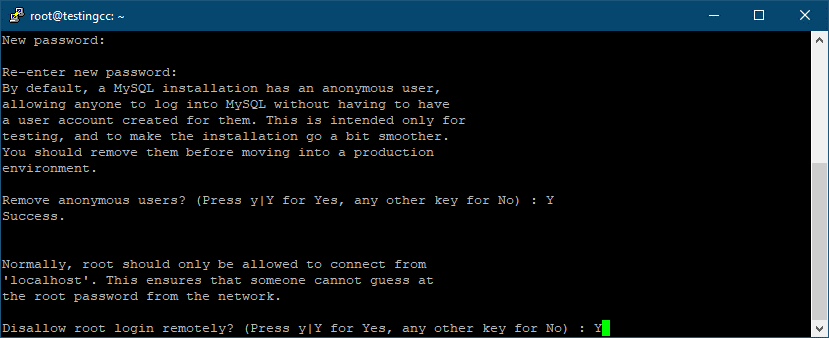
* Wizard akan dimulai dengan menanyakan perihal validasi penggunaan kata sandi pada MySQL apakah kata sandi yang digunakan harus menggunakan kombinasi yang kuat atau tidak. Untuk mempermudah praktikum Cloud Computing ini, maka validasi kata sandi dimatikan dengan memberikan masukan nilai No sebagaimana gambar berikut:



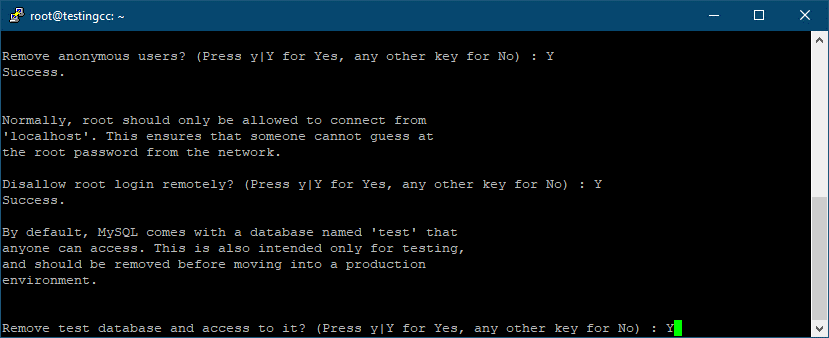
* Dilanjutkan dengan proses pengaturan kata sandi untuk akun root pada MySQL. Isilah kata sandi yang mudah diingat dikarenakan akan digunakan pada materi berikutnya. Dilarang mengisi nilai kosong atau kata sandi yang random, Anda wajib mengingatnya selama praktikum ini. Untuk memudahkan, maka isilah NIM Anda pada kata sandi ini.
* Selanjutnya adalah peringatan mengenai Anonymous User pada MySQL. Masukkan parameter Y yang merupakan Yes untuk menghapus anonymous user sebagai standar keamanan MySQL Cloud. Berikut tampilan parameter yang dimasukkan:



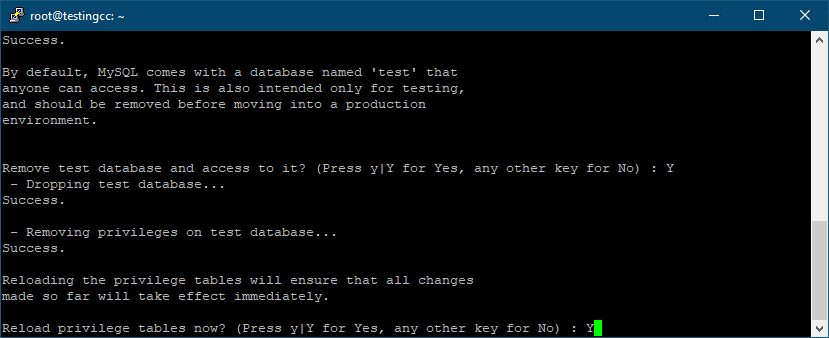
* Kemudian dilanjutkan peringatan mengenai Remote Login ke basis data MySQL dari luar jaringan localhost (Ubuntu Server), misal dari Workstation. Pada menu ini, isi dengan perintah Y sehingga kemampuan Remote Login dengan akun root (yang sedang dilakukan pengaturan saat ini) pada server Cloud tidak diaktifkan, dikarenakan mempengaruhi keamanan server Cloud. Berikut tampilan parameter yang dimasukkan:



* Tahap berikutnya adalah peringatan mengenai database dengan nama test yang secara default terpasang pada MySQL. Untuk alasan keamanan, maka lakukan penghapusan database tersebut dengan memasukkan perintah Y sebagaimana tampilan berikut ini:



* Tahap terakhir adalah proses dari refresh/reload tabel privilege atau hak akses dari MySQL. Masukkan parameter Y pada menu ini sehingga tabel privilage memiliki definisi nilai yang terbaru dan dapat diakses dengan username dan kata sandi yang telah diatur pada tahap sebelumnya. Berikut tampilan parameter yang dimasukkan:



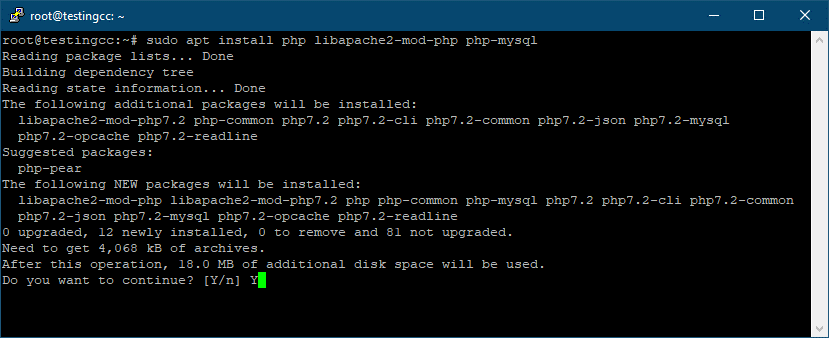
* Hasil akhir dari tahap wizard tersebut adalah informasi “All done!” yang menandakan proses akhir dari wizard pengaturan MySQL.

### Instalasi PHP

Tahap berikutnya adalah instalasi layanan PHP sebagai prosesor konten Web Server dengan jenis PHP (ekstensi PHP). Layanan tersebut diperlukan, dikarenakan layanan Apache hanya dapat memproses konten Web Server HTML dan sejenisnya (CSS, JS, dsb). Selain itu, PHP mampu memberikan layanan konektivitas antara Web Server dengan layanan MySQL. Berikut adalah proses instalasi yang perlu dilakukan:

* Lakukan proses instalasi package php libapache2-mod-php php-mysql sebagaimana sintak perintah dan gambar berikut ini:

|  |
| --- |
| $ sudo apt install php libapache2-mod-php php-mysql |



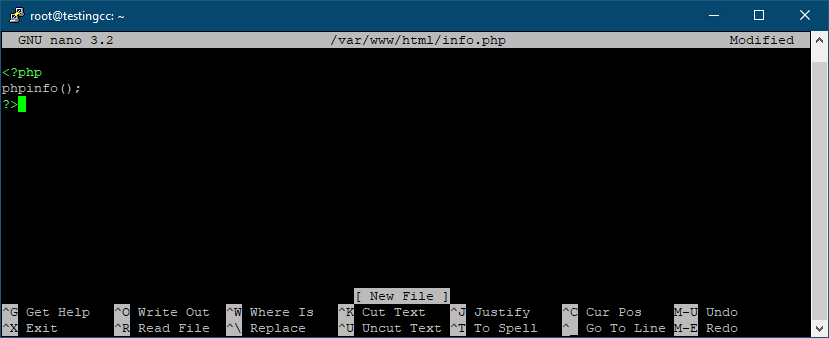
Pada umumnya, perintah tersebut akan melakukan instalasi PHP dan mengatur dengan pengaturan default sehingga siap digunakan. Berikut akan dibuat suatu kodingan sederhana sehingga dapat dibuktikan hasil instalasi PHP dapat berjalan dengan sempurna. Pertama adalah membuat berkas dengan format .php pada lokasi direktori Web Server Apache yakni di /var/www/html/ menggunakan media aplikasi nano (sejenis aplikasi Notepad pada Windows namun berbasiskan CLI). Masukkan sintaks berikut pada jendela terminal PuTTY:

|  |
| --- |
| $ sudo nano /var/www/html/info.php |

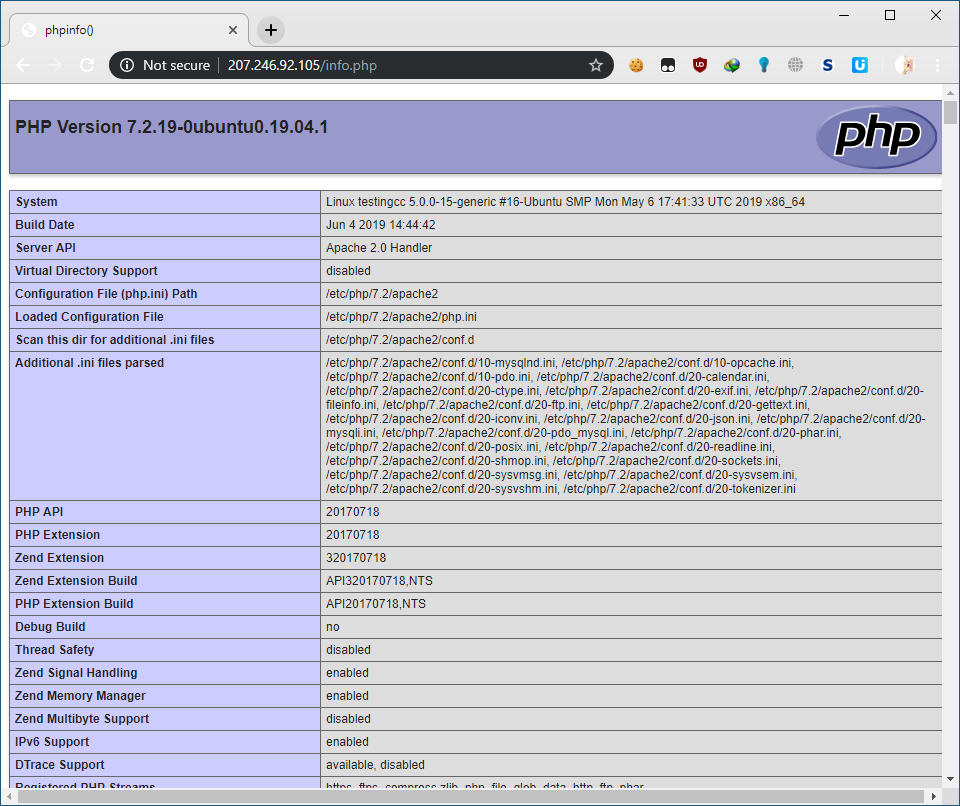


* Kemudian akan muncul aplikasi GNU nano yang berisikan lembar kerja dari berkas info.php. Pada lembar kerja tersebut, ketikan kodingan sebagaimana berikut, dilanjutkan dengan proses penyimpanan dengan cara menekan tombol CTRL+O untuk proses Write Out (Save) lalu Enter untuk konfirmasi penyimpanan. Setelah itu, tekan tombol CTRL+X untuk keluar dari aplikasi nano. Berikut kodingan dan tampilan lembar kerja yang dibuat:

|  |
| --- |
| <?php  phpinfo();  ?> |



* Langkah berikutnya adalah mengecek pada jendela browser Workstation dengan cara akses alamat Server Cloud dilanjutkan dengan nama berkas yang telah dibuat pada tahap sebelumnya. Sintaksnya adalah <http://IP_Server_Cloud/info.php>. Berikut adalah tampilan bilamana layanan PHP telah aktif dan dapat diproses dengan benar oleh Apache Web Service untuk disajikan pada *user*:

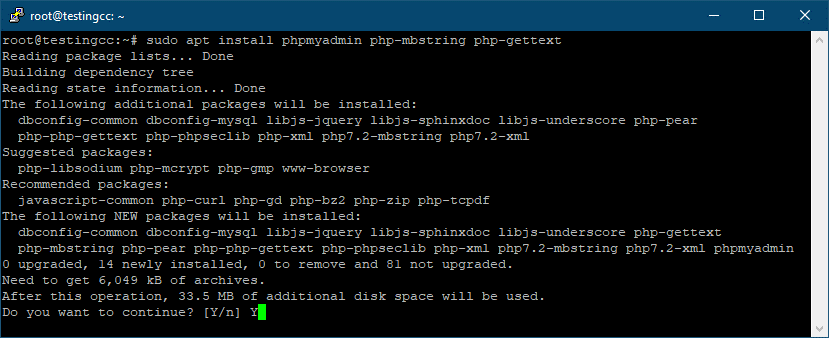


### Instalasi PHPMyAdmin

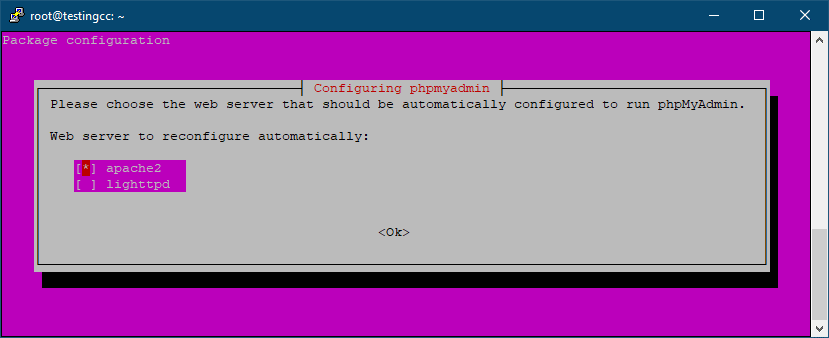
Pada tahap terakhir adalah proses instalasi PHPMyAdmin untuk layanan manajemen database MySQL berbasiskan Web GUI. Tahap ini bersifat opsional, dikarenakan MySQL tetap dapat dilakukan manajemen melalui Terminal dengan berbasiskan Command Line (CLI) MySQL. Berikut tahapan instalasi PHPMyAdmin pada Cloud Server:

* Lakukan instalasi package phpmyadmin php-mbstring php-gettext sebagaimana sintak perintah dan gambar berikut ini:

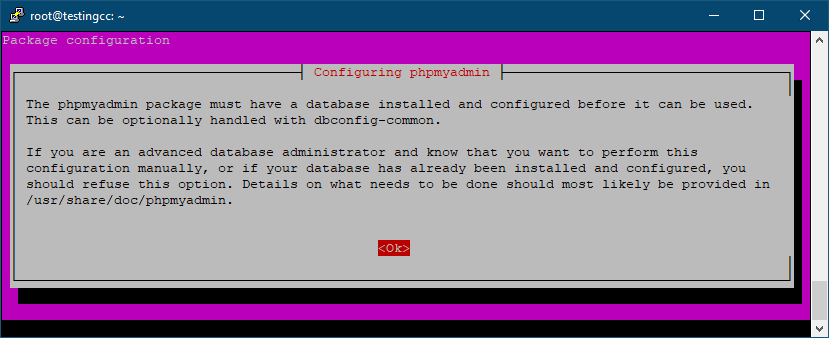
|  |
| --- |
| $ sudo apt install phpmyadmin php-mbstring php-gettext |

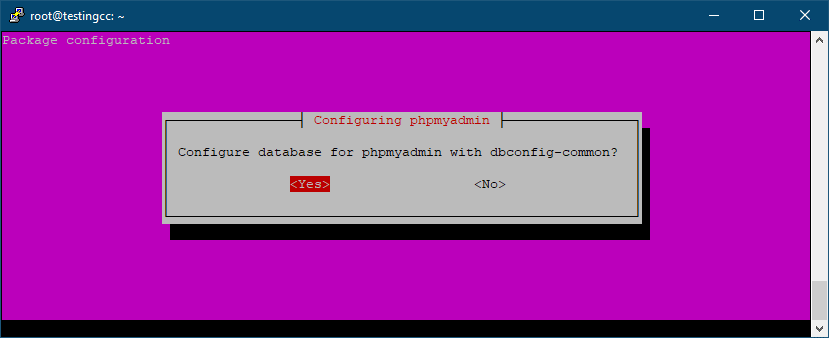


* Pada tahap instalasi berikut, berhati-hatilah sebelum menekan tombol Enter dikarenakan opsi secara default tidak terpilih, antara apache2 maupun lighttpd. Arahkan pada apache2 lalu tekan tombol Spasi sekali hingga muncul simbol asterik (\*) menandakan opsi telah terpilih. Setelah itu tekan Enter. Berikut adalah tampilannya:

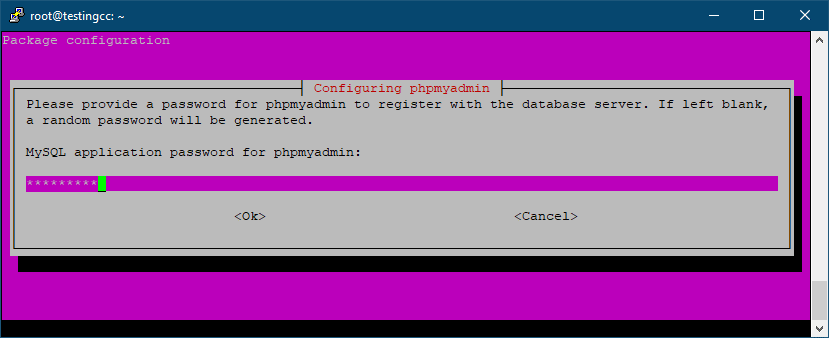


* Selanjutnya akan muncul peringatan mengenai database untuk PHPMyAdmin dikarenakan PHPMyAdmin memerlukan database internal untuk fungsionalitas layanan tersebut. Tekan Enter untuk melanjutkan. Kemudian pada tampilan konfigurasi untuk database dengan konfigurasi default dbconfig-common, tekan Enter pada pilihan Yes. Berikut tampilannya:

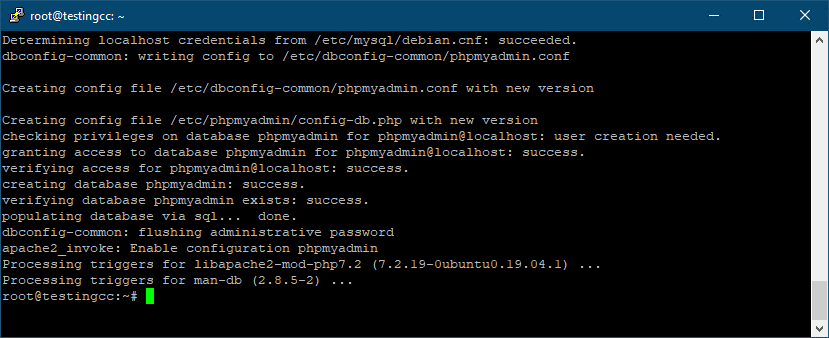




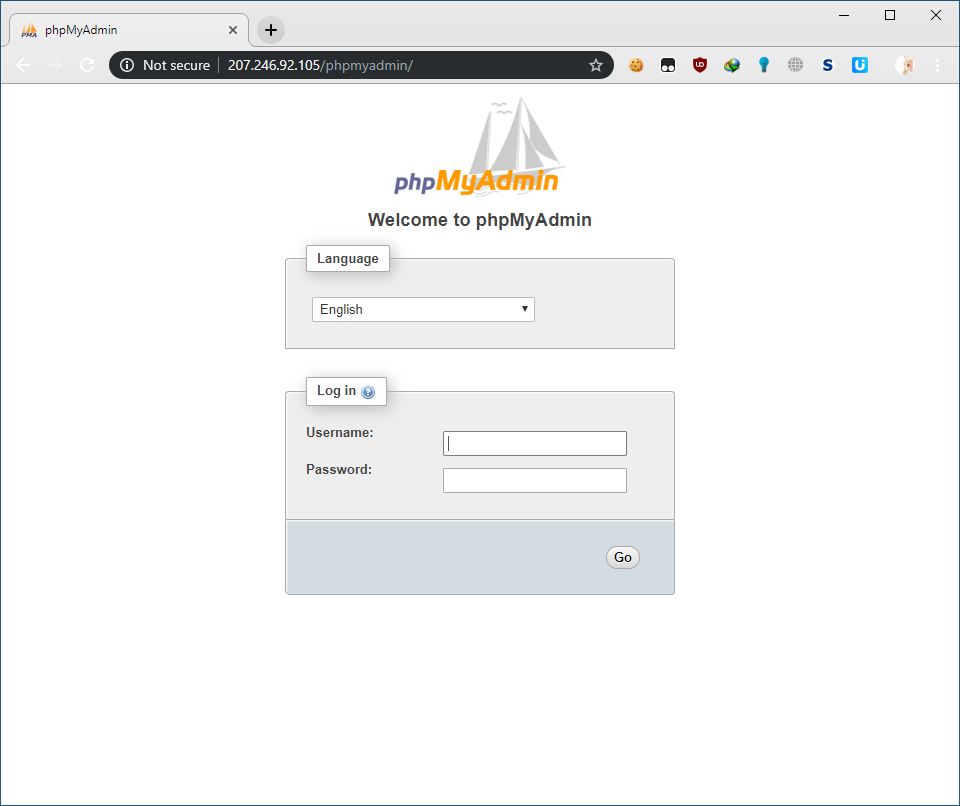
* Langkah selanjutnya adalah menentukan kata sandi untuk database PHPMyAdmin, isi kata sandi yang mudah diingat sama dengan kata sandi MySQL. Untuk mempermudah, masukkan NIM Anda pada kolom input kata sandi. Berikut tampilannya:



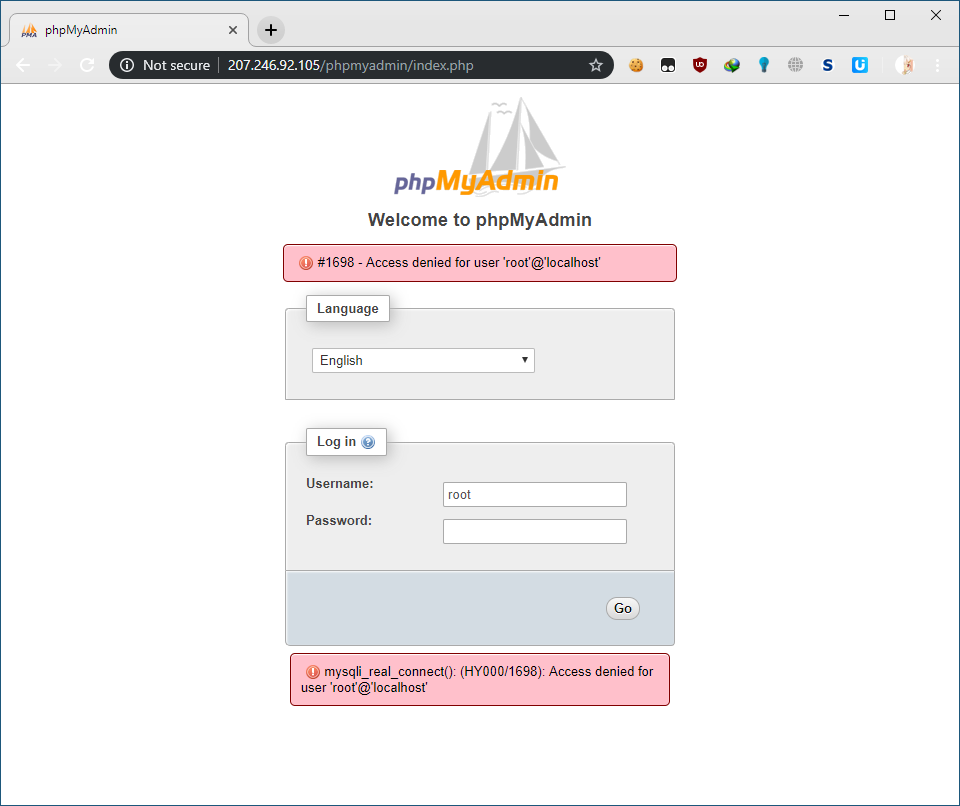
* Setelah proses tersebut maka akan didapatkan tampilan hasil akhir proses instalasi sebagai berikut yang menandakan proses instalasi telah berhasil:



* Untuk membuktikan hasil instalasi telah berhasil maka lakukan pengecekan pada browser Workstation dengan membuka alamat berikut <http://IP_Server_Cloud/phpmyadmin>. Berikut tampilannya:

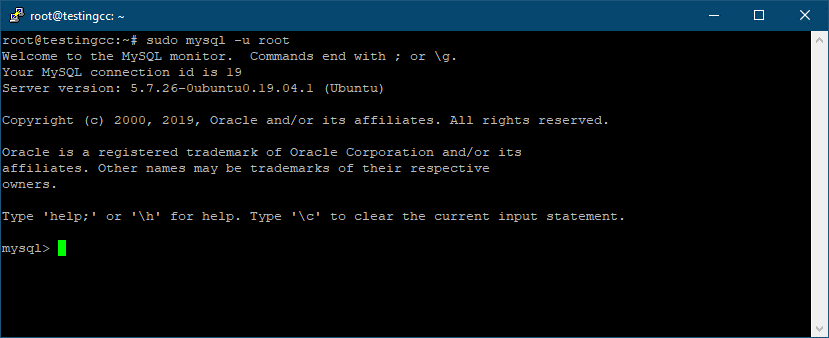


* Bila dilakukan proses akses login ke database dengan username dan password yang telah diatur pada MySQL maka akan didapat pesan error. Pesan error tersebut dikarenakan konfigurasi user root yang digunakan untuk login diatur dengan model plugin Unix Socket based authentication, sehingga perlu diatur plugin menggunakan Native MySQL authentication. Berikut tampilan error yang dimaksud:



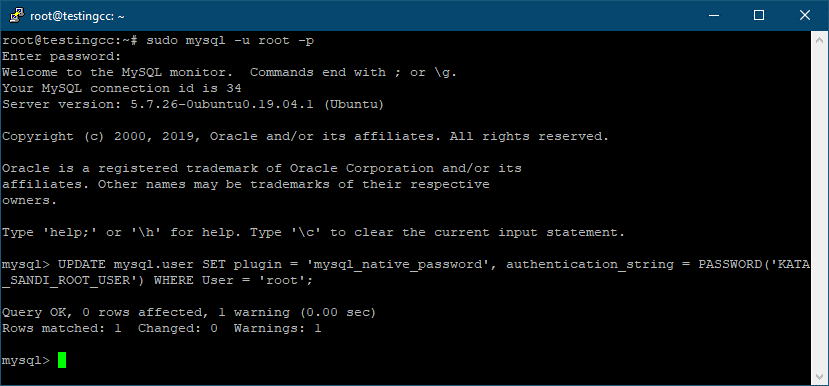
* Untuk memperbaiki error tersebut, maka ketikan sintak berikut untuk masuk ke CLI dari konfigurasi MySQL sebagai user root. Setelah sintak tersebut dimasukkan, ketikkan kata sandi dari user root yang telah diatur sebelumnya. Berikut tampilannya:

|  |
| --- |
| $ sudo mysql -u root  ATAU  $ sudo mysql -u root -p |

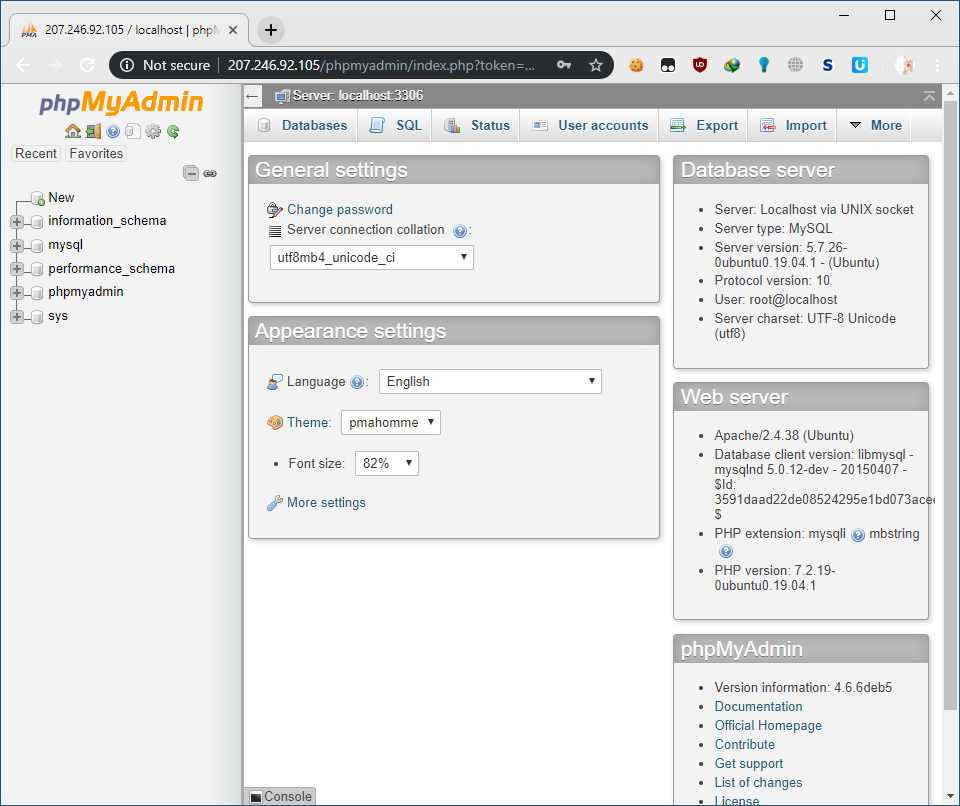


* Kemudian ketikan sintak berikut untuk mengatur parameter plugin dari user root. Atur parameter KATA\_SANDI\_ROOT\_USER dengan kata sandi akun root pada MySQL yang telah dibuat sebelumnya (NIM Anda). Berikut sintak yang digunakan dan tampilannya:

|  |
| --- |
| UPDATE mysql.user SET plugin = 'mysql\_native\_password', authentication\_string = PASSWORD('KATA\_SANDI\_ROOT\_USER') WHERE User = 'root'; |



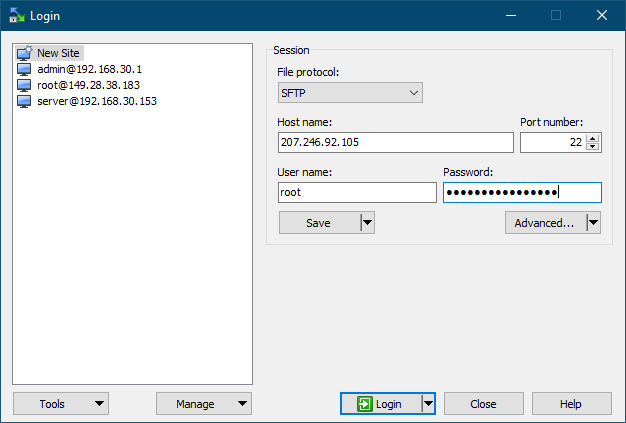
* Setelah itu untuk keluar dari jendela pengaturan CLI MySQL, ketikkan perintah exit lalu tekan Enter. Kemudian buka browser pada Workstation dan buka alamat PHPMyAdmin. Akses dengan username dan kata sandi dari MySQL yang telah dibuat sebelumnya. Bila berhasil masuk dan mendapatkan tampilan sebagaimana berikut ini maka proses instalasi PHPMyAdmin telah selesai:



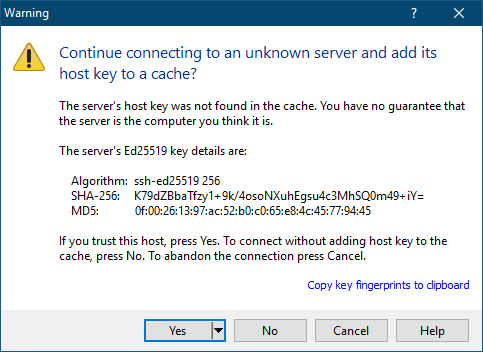
### Proses Upload Berkas dengan WinSCP

Untuk mengupload berkas menggunakan aplikasi WinSCP, berikut langkah-langkahnya:

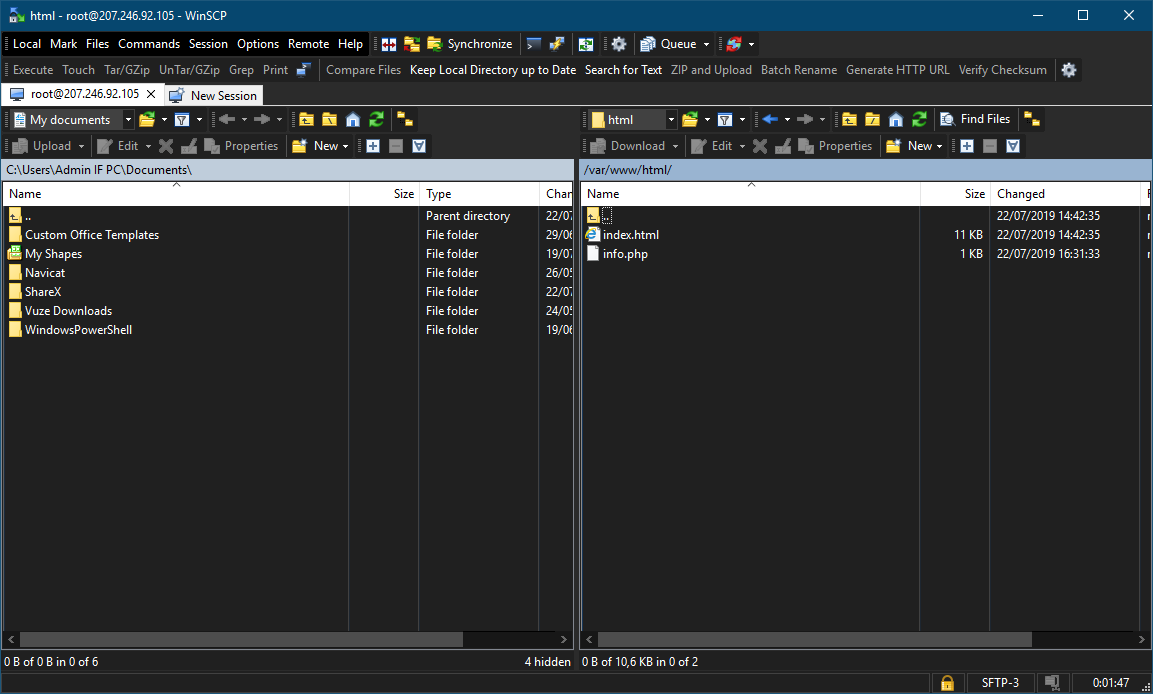
* Buka aplikasi WinSCP pada Workstation. Setelah itu, pada dropdown File Protocol pastikan terpilih Secure File Transfer Protocol (SFTP), kemudian isi Hostname dengan IP Server Cloud, dilanjutkan pastikan pada Port Number menggunakan 22 (port SSH), lalu isikan username dan password dari Server Cloud yang telah dibuat Hasil akhir input parameter koneksi akan didapatkan tampilan awal sebagai berikut ini:



* Kemudian setelah ditekan tombol Login akan muncul peringatan konfirmasi sertifikat dari Server. Pilih Yes untuk mengonfirmasi sertifikat sebagaimana tampilan berikut:



* Kemudian pada kolom sebelah kanan merupakan tampilan dari berkas yang berada pada Server, sedangkan yang kiri merupakan berkas yang ada pada Workstation. Arahkan pada direktori /var/www/html/ untuk menaruh berkas dari Workstation ke dalam Server. Berikut adalah tampilannya:



* Untuk proses upload berkas kodingan maka digunakan cara drag and drop dari Window Kiri ke dalam Window Kanan sehingga berkas akan diproses upload oleh WinSCP. Untuk melakukan pengecekan hasil upload, dapat dicek pada Server melalui browser Workstation dengan cara mengetikkan URL IP server diikuti nama berkas yang dimaksud.

|  |
| --- |
| Modul V Implementasi Layanan API pada Cloud Computing |

## Tujuan Praktikum

* Mampu membuat suatu REST API
* Penerapan REST-API menggunakan PHP CodeIgniter

## Alokasi Waktu

* 1 x pertemuan = 1 x 120 menit = 120 menit

## Dasar Teori

### REST API

REST merupakan singkatan dari *Representational State Transfer* yang berarti setiap request maupun response memiliki suatu representasi informasi dalam format tertentu. Maksudnya dalam pembuatanya harus menggunakan metode HTTP yang tepat dan respon harus berada dalam suatu format tertentu seperti JSON atau XML.

Berikut ini adalah prinsip penggunaan REST:

* Meyediakan id unik untuk setiap resource, misalnya URI.
* Mampu membangun hubungan antar resources.
* Resoursce bias digunakan pada state yang berbeda pada aplikasi.
* Menggunakan HTTP untuk status request dan respon dalam bentuk JSON atau XML.

### Metode HTTP

HTTP merupakan suatu protocol penting dalam web service. HTTP dapat digunakan untuk mengakses resource, tidak hanya berupa halaman HTML melainkan semua resource termasuk didalamnya adalah gambar, musik, video, dan aplikasi. Ketika melakukan request menggunakan HTTP perlu memperhatikan URI dan perintah HTTP yang akan digunakan. Berikut adalah perintah HTTP yang digunakan dalam pemanggilan resource:

* **Get**: Mengembalikan resource berdasarkan URI yang dipanggil atau dengan kata lain membaca record.
* **POST**: Mengirim resource ke server atau dengan kata lain menambah record.
* **PUT**: Mengirim resource ke server untuk disimpan di URI yang dituju atau bisa disebt juga dengan update record.
* **DELETE**: Menghapus resource berdasarkan URI yang dipanggil atau menghapus record.
* **HEAD**: Mengembalikan atau membaca metadata dari resource yang dipanggil.

## Praktikum

Untuk membuat suatu REST API menggunakan PHP CodeIgniter dibutuhkan langkah-langkah seperti membuat database, setting autoload, dan setting URI.

#### Pembuatan Database

Buatlah database dengan nama **akademik.** Kemudian buat table **mahasiswa** dengan ketentuan seperti berikut:

|  |
| --- |
| nim varchar (8) not null;  nama varchar (45) not null;  kelas varchar (5) not null;  tglahir date not null;  primary key (nim); |

Untuk settingan database yang akan digunakan: **username = root** dan **password = password**.

#### Setting Autoload

Berikutnya adalah setting autoload. File autoload.php berada pada folder application/config/. Ubahlah file tersebut menjadi seperti berikut:

|  |
| --- |
| $autoload[‘libraries’] = array();  //diubah menjadi |
| $autoload[‘libraries’] = array(‘database’); |

|  |
| --- |
| $autoload[‘helper’] = array();  //diubah menjadi |
| $autoload[‘libraries’] = array(‘url’); |

#### Setting URI

Setting URI berfungsi untuk menghilangkan index.php pada URI resource serta pengaktifan file autoload.php. Setting URI dilakukan dengan mengakases file config.php pada folder application/config.

|  |
| --- |
| $autoload[‘composer\_autoload’] = FALSE;  //diubah menjadi |
| $autoload[‘composer\_autoload’] = realpath(APPPATH.‘../vendor/autoload.php’); |

|  |
| --- |
| $autoload[‘index\_page’] = index.php;  //diubah menjadi |
| $autoload[‘index\_page’] = index.php; |

#### Setting Koneksi Database

Sebelum membuat REST-API, sebaiknya setting file database.php untuk melakukan koneksi antara php dan database yang telah dibuat. Untuk melakukan setting koneksi database, buka file database.php pada folder application/config/ kemudian konfigurasi username, password, database, dan localhost menjadi seperti berikut:

|  |
| --- |
| ...  'hostname' => 'localhost',  'password' => 'password',  'database' => 'akademik', |
| 'username' => 'root',  ... |

#### Koding Model

CodeIgniter merupakan framework MVC (Model View Controller) sehingga perlu untuk melakukan koding pada Model, View, dan Controller. Model biasanya berisi query atau proses manipulasi database.

Untuk membuat model pada framework CI, buatlah sebuah file model\_mhs.php pada folder application/models/ dan masukkan kodingan berikut.

|  |
| --- |
| <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');  class Model\_mhs extends CI\_Model {  public function getCountMahasiswa()  {  return $this->db->count\_all\_results('mahasiswa', FALSE);  }  public function getMahasiswa($page, $size)  {  return $this->db->get('mahasiswa', $size, $page);  }  public function insertMahasiswa($dataMahasiswa)  {  $val = array(  'nim' => $dataMahasiswa['nim'],  'nama' => $dataMahasiswa['nama'],  'kelas' => $dataMahasiswa['kelas'],  'tglahir' => $dataMahasiswa['tglahir']  );  $this->db->insert('mahasiswa', $val);  }  public function updateMahasiswa($dataMahasiswa, $nim)  {  $val = array(  'nama' => $dataMahasiswa['nama'],  'kelas' => $dataMahasiswa['kelas'],  'tglahir' => $dataMahasiswa['tglahir']  );  $this->db->where('nim', $nim);  $this->db->update('mahasiswa', $val);  }  public function deleteMahasiswa($nim)  {  $val = array(  'nim' => $nim  );  $this->db->delete('mahasiswa', $val);  }  } |

Pada Model yang dibuat terdapat beberapa function dengan kegunaan sebagai berikut:

* **getCountMahasiswa**: untuk mengambil data banyak mahasiswa.
* **getMahasiswa**: untuk mengambil data dari table mahasiswa.
* **insertMahasiswa**: untuk menyimpan data mahasiswa.
* **updateMahasiswa**: untuk melakukan update atau edit data mahasiswa.
* **deleteMahasiswa**: untuk menghapus data mahasiswa.

#### Koding Controller

Setelah melakukan koding model, berikutnya adalah koding controller yang berfungsi untuk meload dan memproses model yang telah dibuat. Buatlah file Mahasiswa.php pada folder application/controllers/ dan masukkan kodingan berikut.

|  |
| --- |
| <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');  class MahasiswaController extends CI\_Controller {  public function \_\_construct()  {  parent::\_\_construct();  $this->load->model('model\_mhs');  }  public function getMahasiswa($page, $size)  {  $response = array(  'content' => $this->model\_mhs->getMahasiswa(($page - 1) \* $size, $size)->result(),  'totalPages' => ceil($this->model\_mhs->getCountMahasiswa() / $size));  $this->output  ->set\_status\_header(200)  ->set\_content\_type('application/json', 'utf-8')  ->set\_output(json\_encode($response, JSON\_PRETTY\_PRINT))  ->\_display();  exit;  }  public function saveMahasiswa()  {  $data = (array)json\_decode(file\_get\_contents('php://input'));  $this->model\_mhs->insertMahasiswa($data);  $response = array(  'Success' => true,  'Info' => 'Data Tersimpan');  $this->output  ->set\_status\_header(201)  ->set\_content\_type('application/json', 'utf-8')  ->set\_output(json\_encode($response, JSON\_PRETTY\_PRINT))  ->\_display();  exit;  }  public function updateMahasiswa($nim)  {  $data = (array)json\_decode(file\_get\_contents('php://input'));  $this->model\_mhs->updateMahasiswa($data, $nim);  $response = array(  'Success' => true,  'Info' => 'Data Berhasil di update');  $this->output  ->set\_status\_header(200)  ->set\_content\_type('application/json', 'utf-8')  ->set\_output(json\_encode($response, JSON\_PRETTY\_PRINT))  ->\_display();  exit;  }  public function deleteMahasiswa($nim)  {  $this->model\_mhs->deleteMahasiswa($nim);  $response = array(  'Success' => true,  'Info' => 'Data Berhasil di hapus');  $this->output  ->set\_status\_header(200)  ->set\_content\_type('application/json', 'utf-8')  ->set\_output(json\_encode($response, JSON\_PRETTY\_PRINT))  ->\_display();  exit;  }  } |

Adapun function yang ada pada controller\_mhs memiliki fungsi berikut.

* **\_\_construct**: untuk meload model yang sudah dibuat.
* **getMahasiswa**: mengambil seluruh data mahasiswa yang direquest oleh client. Hasilnya kemudian diencode kedalam format JSON yang bisa dilihat pada json\_encode.
* **saveMahasiswa**: menyimpan data mahasiswa yang diambil dari response JSON. Data didecode menjadi object array kemudian dikirim ke model untuk disimpan ke database.
* **updateMahasiswa**: memperbarui data mahasiswa berdasarkan nim dengan proses seperti saveMahasiswa.
* **deleteMahasiswa**: tidak ada parameter JSON yang dikirim tetapi melakukan penghapusan berdasarkan nim yang diterima.

#### Setting Routes

Routing dilakukan untuk menjelaskan perintah HTTP yang akan dieksekusi oleh controller pada mahasiswa.php. Untuk melakukan konfigurasi routes, buka file routes.php pada application/config/ dan masukkan kodingan berikut.

|  |
| --- |
| $route['api/mahasiswa/(:num)/(:num)']['GET'] = 'MahasiswaController/getMahasiswa/$1/$2';  $route['api/mahasiswa']['POST'] = 'MahasiswaController/saveMahasiswa';  $route['api/mahasiswa/(:any)']['PUT'] = 'MahasiswaController/updateMahasiswa/$1';  $route['api/mahasiswa/(:any)']['DELETE'] = 'MahasiswaController/deleteMahasiswa/$1'; |

Ketika melakukan setting routes.php sebenarnya secara tidak langsung sudah melakukan custom URI untuk resource yang tersedia/dimiliki.

* Route pertama mengambil data mahasiswa menggunakan metode GET berdasarkan protokol HTTP dengan 2 parameter, yaitu page dan size.
* Route kedua berfungsi untuk menyimpan data dalam bentuk JSON dengan menggunakan metode POST pada protokol HTTP.
* Route ketiga berfungsi untuk memperbarui data JSON yang dikirim berdasarkan nim dengan menggunakan metode PUT pada protokol HTTP.
* Route keempat berfungsi untuk menghapus data mahasiswa dengan menggunakan metode DELETE pada protokol HTTP.

|  |
| --- |
| Modul VI Instalasi dan Konfigurasi Layanan Docker dengan Dokku (PaaS) |

1. **Tujuan praktikum**

Tujuan dari praktikum pada bab ini adalah sebagai berikut:

* Mendapatkan pemahaman mengenai arsitektur *Platform as a Service* (*PaaS*) pada *Cloud Computing*.
* Mendapatkan pemahaman mengenai salah satu layanan *PaaS* (Docker dan Dokku) yang kemudian dikombinasikan dengan layanan *SaaS* untuk digunakan pada *Cloud Computing*.
* Mendapatkan pemahaman mengenai konfigurasi dan *troubleshooting* yang terjadi pada tahap *deploying PaaS architecture.*

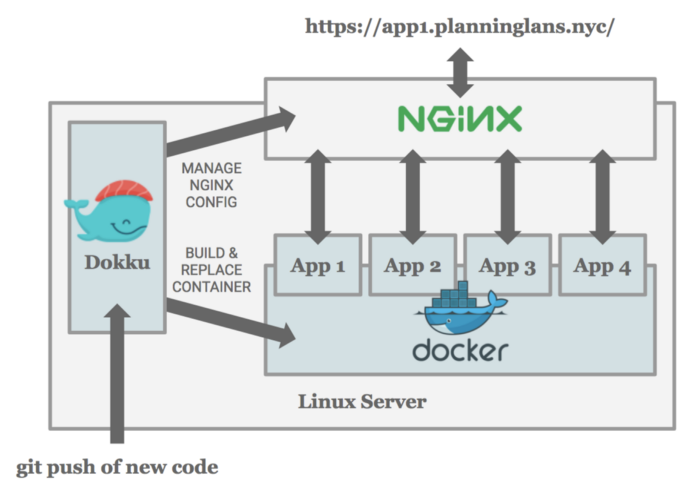
1. **Alokasi waktu**

Alokasi waktu yang dibutuhkan pada bab ini adalah 2 x 60 menit yang terbagi menjadi:

* Sesi demonstrasi dan penyampaian materi sebanyak 3 x 30 menit.
* Sesi latihan mandiri sebanyak 1 x 30 menit.

1. **Dasar teori**

*Docker* merupakan perangkat lunak yang memudahkan dalam proses pembuatan, penciptaan (*deploy*), dan menjalankan aplikasi (*running*) yang salah satunya menggunakan konsep *container*. *Container* pada *Docker* memungkinkan *developer* membuat paket aplikasi dengan berbagai komponen yang dibutuhkan seperti *library* dan *dependency*/ketergantungan dengan paket aplikasi yang lain dalam satu kesatuan *package* yang utuh. Dengan demikian, *developer* tidak perlu khawatir terhadap aplikasi yang dibuat dan akan dijalankan pada mesin Linux lainnya (*end user*), meskipun memiliki lingkungan (pengaturan) Linux yang berbeda dengan yang digunakan *developer* saat menciptakan aplikasi tersebut.



**Gambar 8.1** Interaksi antar komponen Docker, Dokku, dan Nginx

Secara garis besar, *Docker* merupakan sesuatu yang menyerupai *Virtual Machine*. Namun melainkan *Virtual Machine* yang memuat bermacam-macam komponen, *Docker* memungkinkan aplikasi untuk menggunakan interaksi dengan *kernel* *Linux* dan hanya dengan aplikasi pendukung (*dependency*) yang diikutsertakan dengan aplikasi, tanpa mengganggu *host machine*. Dengan demikian, akan terdapat peningkatan performa baik dari segi performa aplikasi maupun ukuran aplikasi dibandingkan dengan menggunakan *Virtual Machine* secara utuh untuk distribusi aplikasi dari developer. Selain itu, proses *deploy* cukup mudah dan cepat, hanya dengan beberapa baris perintah, maka layanan *SaaS* akan tercipta dengan cepat dan bekerja dengan baik.

Hal terakhir dari *Docker* ialah sifatnya yang *open source*. Dengan sifat tersebut, setiap orang dapat berkontribusi terhadap pengembangan *Docker* dan memperluas kemampuan dari *Docker* sesuai dengan keinginan *user* bilamana saat ini belum tersedia di *Docker*.

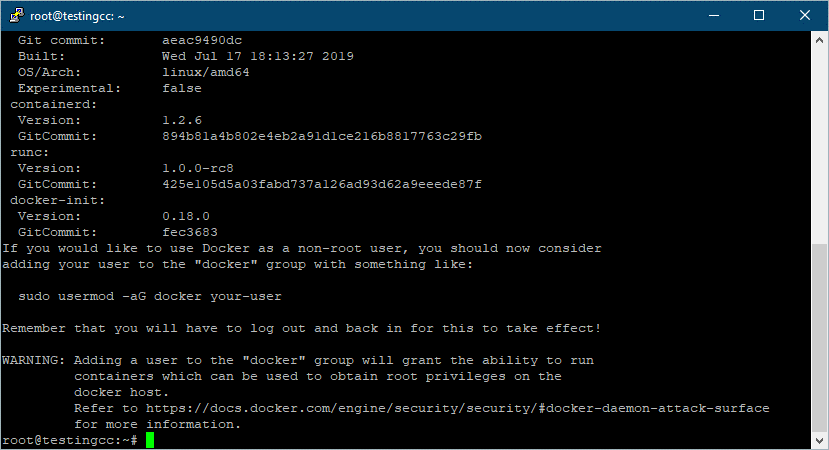
1. **Langkah Praktikum**

Untuk mengilustrasikan penggunaan *Docker*, maka akan digunakan basis layanan *Dokku*. *Dokku* merupakan *Platform as a Service* yang memungkinkan developer untuk mengembangkan aplikasi yang dibuatnya dalam lingkungan yang berbeda (misalnya *Production Environtment* atau *Development Environtment*) untuk keperluan sebelum tahap rilis atau kebutuhan lainnya. Berikut tahap yang digunakan:

1. Tahap pertama ialah instalasi *Docker* sebagai basis dukungan dari *Dokku*. Berikut parameter yang digunakan beserta tampilan ketika instalasi telah berhasil:

|  |
| --- |
| # PERINTAH UNTUK UNDUH DOCKER DAN INSTALASI OTOMATIS DOCKER  $ wget -nv -O - https://get.docker.com/ | sudo sh |

**Parameter 8.1** Proses unduh dan instalasi *Docker*

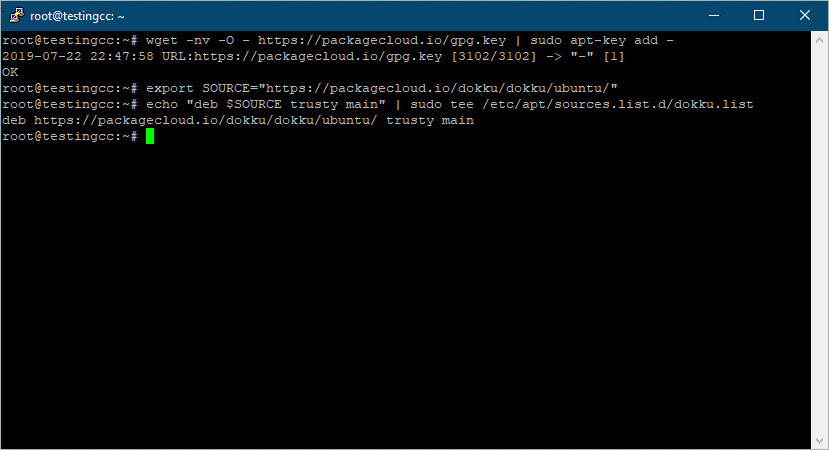


**Gambar 8.2** Tampilan proses instalasi *Docker*

1. Setelah proses instalasi *Docker* selesai, sebelum dapat melakukan instalasi *Dokku*, diperlukan penambahan *repository* baru pada daftar *repository* *Ubuntu* saat ini. Hal tersebut diperlukan karena *Dokku* untuk saat ini berada pada *repository* yang berbeda dengan *default* dari *Ubuntu*. Berikut parameter yang digunakan dan tampilan hasil proses penambahan *repository*:

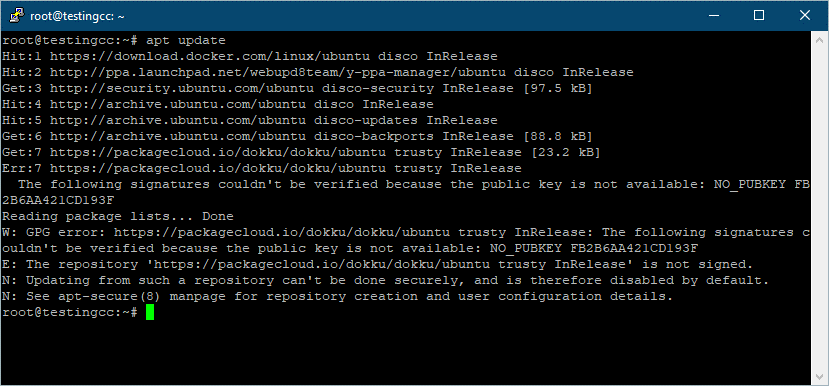
|  |
| --- |
| # PERINTAH UNTUK UNDUH REPOSITORY (GPG KEY) DOKKU DILANJUTKAN PENAMBAHAN KE DAFTAR REPOSITORY SAAT INI  $ wget -nv -O - https://packagecloud.io/gpg.key | sudo apt-key add -  $ export SOURCE="https://packagecloud.io/dokku/dokku/ubuntu/"  $ echo "deb $SOURCE trusty main" | sudo tee /etc/apt/sources.list.d/dokku.list |

**Parameter 8.2** Perintah untuk instalasi *GPG Key*



**Gambar 8.3** Proses instalasi *GPG Key*

1. Tahap berikutnya adalah memuat ulang daftar aplikasi yang tersedia pada *Linux* dengan cara apt update. Pada awalnya, akan didapat pesan *error* dikarenakan *GPG key* tidak ditemukan/tidak valid dari *repository* *Dokku*. Berikut tampilannya:

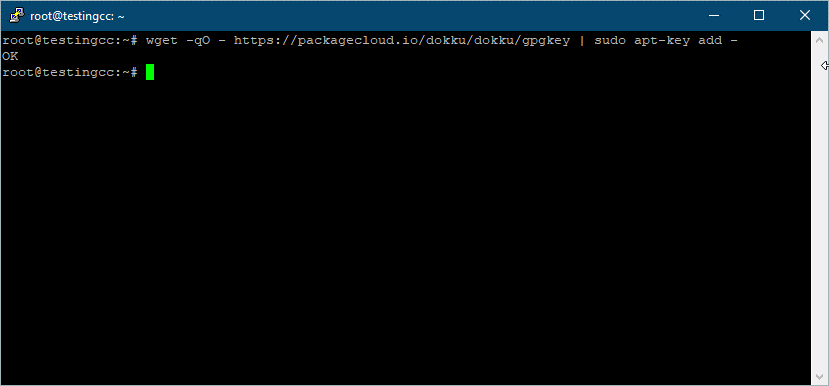


**Gambar 8.4** Pesan kesalahan ketika pembaruan *repository* aplikasi

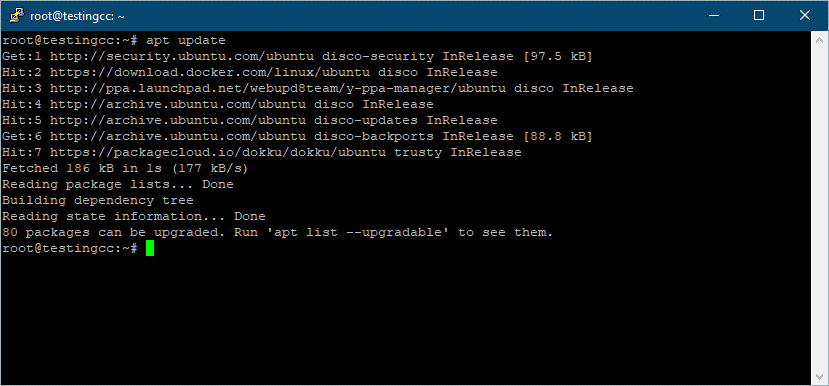
1. Untuk memperbaiki *error* tersebut, maka digunakan proses penambahan *GPG key* yang baru. Setelah perbaikan *GPG key*, kemudian dilanjutkan dengan proses pembaruan daftar aplikasi pada Ubuntu. Parameter yang digunakan adalah sebagai berikut:

|  |
| --- |
| $ wget -qO - https://packagecloud.io/dokku/dokku/gpgkey | sudo apt-key add –  $ sudo apt update |

**Parameter 8.3** Proses pembaruan *GPG Signature* yang *invalid*

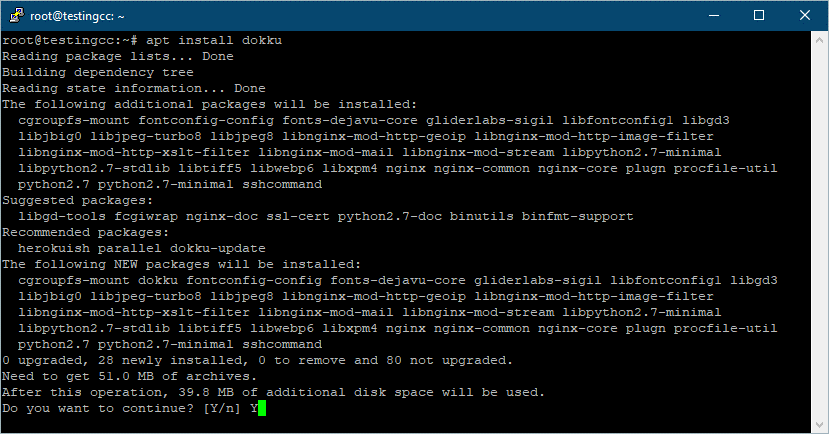


**Gambar 8.5** Tampilan proses pembaruan *GPG Signature*



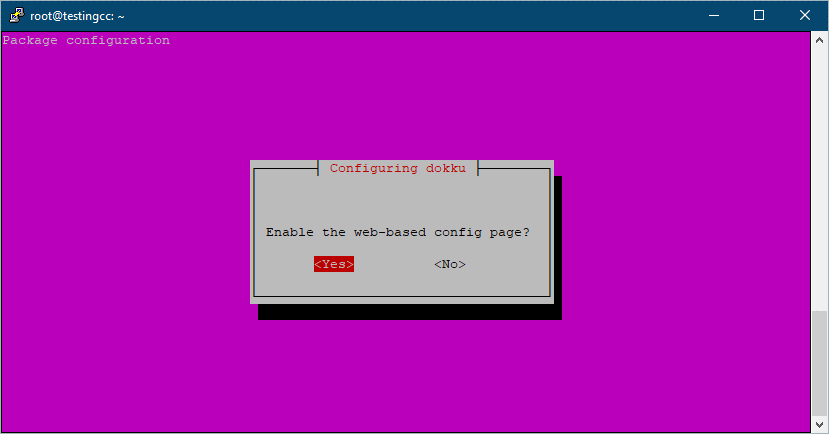
**Gambar 8.6** Hasil proses pembaruan *repository* berhasil

1. Langkah berikutnya setelah *repository* berhasil ditambahkan dan berhasil dimuat dengan sempurna ialah melakukan instalasi aplikasi Dokku. Untuk instalasi digunakan perintah sudo apt install dokku. Berikut tampilannya:



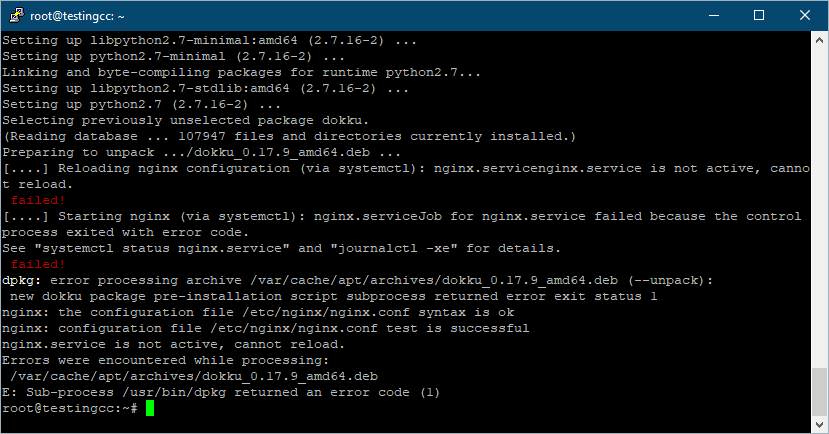
**Gambar 8.7** Proses unduh *Dokku*

1. Pada pesan peringatan untuk instalasi *Web-based Config Page*, pilih opsi Yes sehingga didapatkan modul konfigurasi *Dokku* berbasiskan *Web*. Berikut tampilannya:



**Gambar 8.8** Pesan konfirmasi instalasi *web-based config page*

1. Ketika proses instalasi akan berakhir, maka akan ditemukan *error* mengenai aktivasi layanan *Nginx* (*Web Server* semacam *Apache*) gagal. Berikut tampilan *error*-nya:

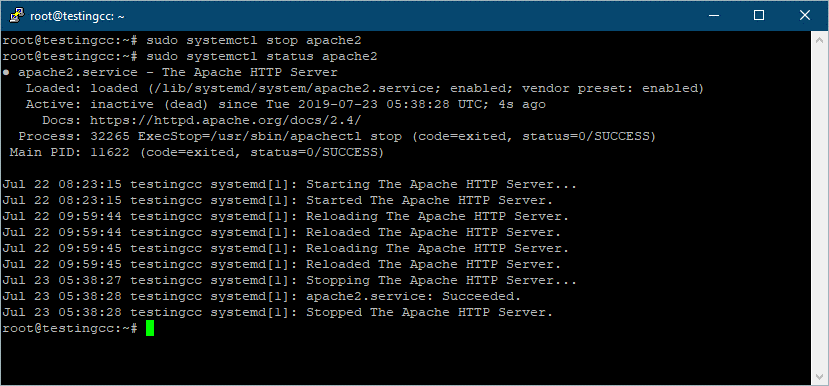


**Gambar 8.9** Pesan kegagalan instalasi *Dokku*

Hal tersebut terjadi dikarenakan layanan *Apache* saat ini sedang aktif dan mengalokasikan pada *port* 80 (HTTP), sehingga terjadi bentrok dengan *Nginx* yang akan menggunakan layanan *port* 80 (HTTP). Untuk mengatasi *error* tersebut, maka digunakan perintah untuk sementara mematikan layanan *Nginx* pada *Ubuntu Server*. Berikut parameter yang digunakan dan tampilannya:

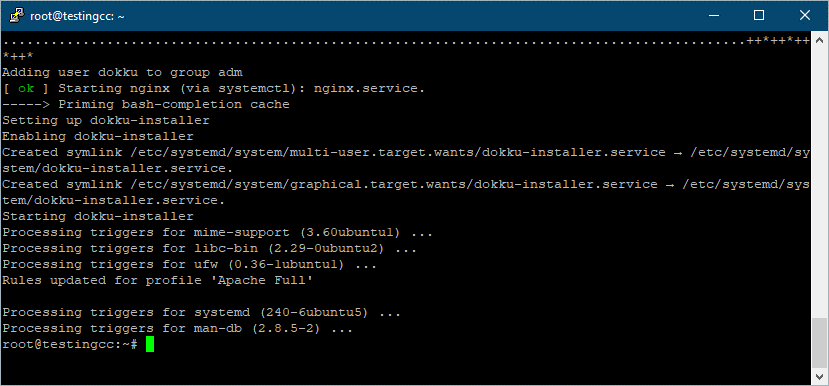
|  |
| --- |
| $ sudo systemctl stop apache2  $ sudo systemctl status apache2 |

**Parameter 8.4** Perintah yang digunakan untuk mengatur layanan *Apache*



**Gambar 8.10** Tampilan proses mematikan dan mengecek status layanan *Apache*

1. Setelah dipastikan layanan *Apache* telah mati (*inactive*) maka proses instalasi *Dokku* diulangi kembali sesuai parameter pada tahap instalasi *Dokku*. Berikut tampilan bilamana instalasi telah berhasil tanpa pesan *error*:

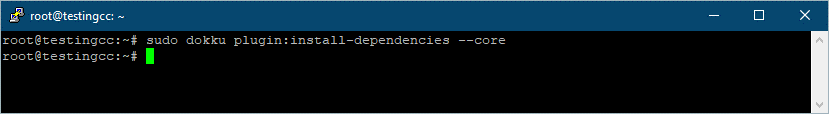


**Gambar 8.11** Hasil akhir proses instalasi *Dokku*

1. Setelah itu dilanjutkan dengan instalasi *plugin* & *dependency* utama *Dokku* menggunakan perintah berikut:

|  |
| --- |
| $ sudo dokku plugin:install-dependencies --core |

**Parameter 8.5** Perintah untuk instalasi *plugin* & *dependency* *Dokku*



**Gambar 8.12** Tampilan instalasi *plugin* & *dependency* *Dokku*

1. Untuk memastikan layanan *Dokku* telah terinstal dan dapat dibuka dengan benar, maka digunakan perintah sederhana yakni pengecekan versi *Dokku*. Parameter pengecekan versi yakni:

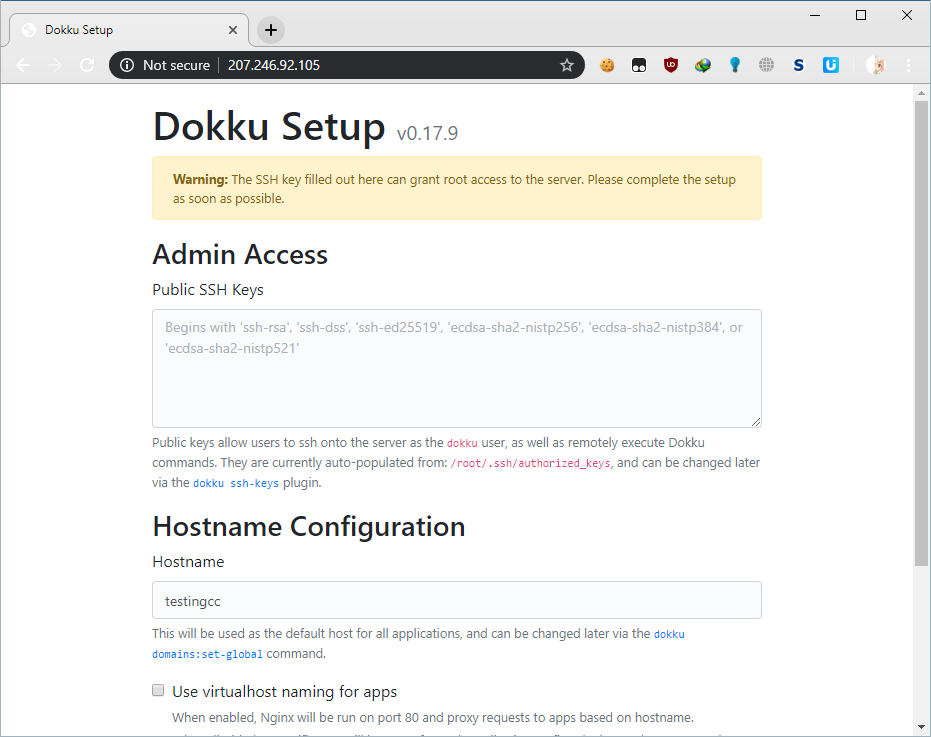
|  |
| --- |
| $ dokku version |

**Parameter 8.6** Perintah untuk pengecekan versi *Dokku*



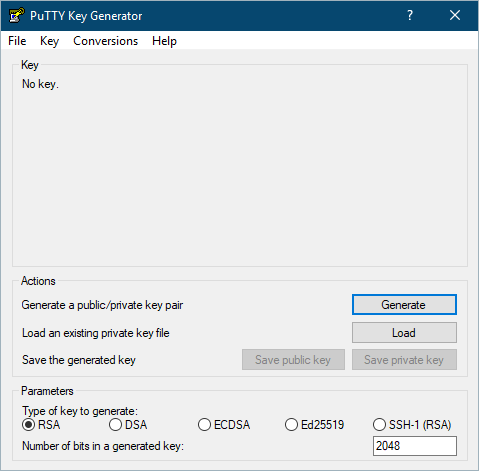
**Gambar 8.13** Tampilan hasil pengecekan versi *Dokku*

1. Tahap berikutnya ialah buka *browser* *Workstation*, arahkan pada *IP Ubuntu* *Server*, maka akan didapat tampilan berikut:



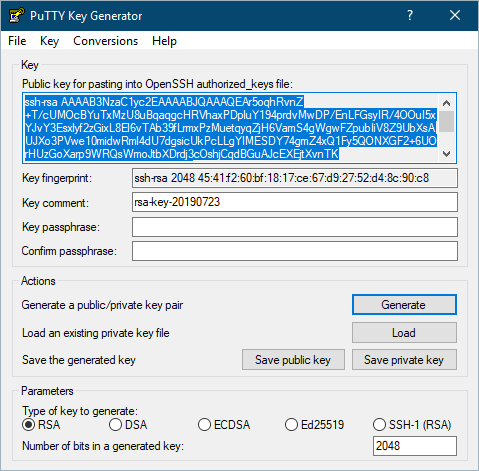
**Gambar 8.14** Tampilan *dashboard* *Dokku Web-config Setup*

Pada tahap ini, diperlukan pengaturan *SSH Key* guna keperluan interaksi dengan *Dokku* menggunakan kunci *SSH*. Pada *Workstation* buka aplikasi *PuTTY* *Gen* sehingga akan didapatkan tampilan sebagaimana berikut ini:



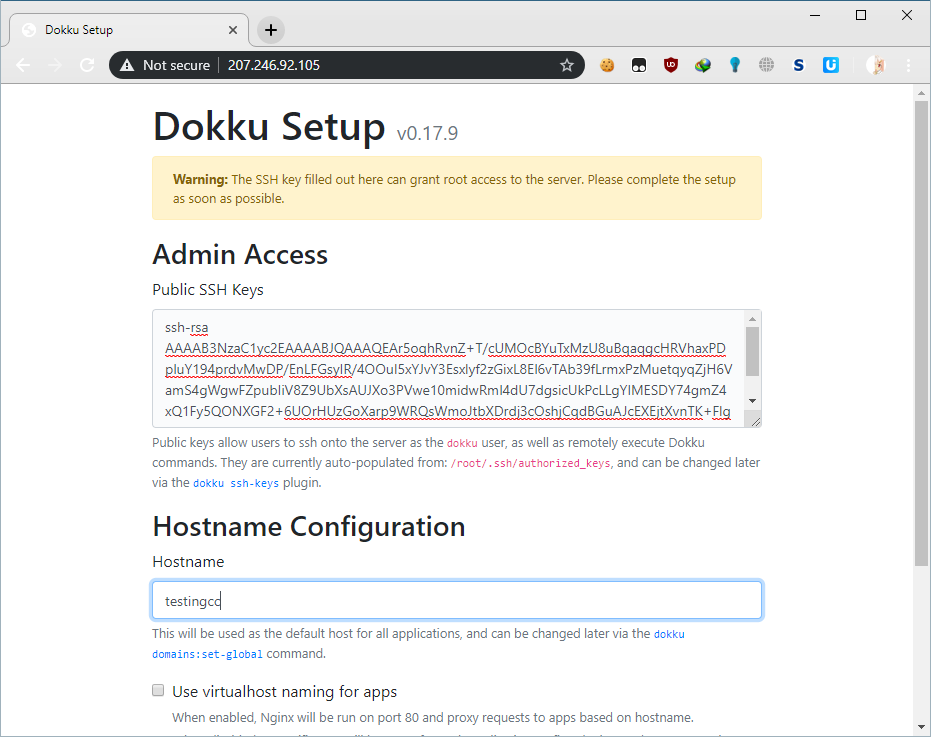
**Gambar 8.15** Tampilan *PuTTY Gen*

Klik tombol Generate untuk membuat kunci *random* baru. Pada *window* atas, gerak-gerakkan mouse sehingga didapatkan pola *random* untuk keperluan *generate key*. Setelah proses *genereate*, akan didapat *key* SSH sebagaimana gambar berikut:



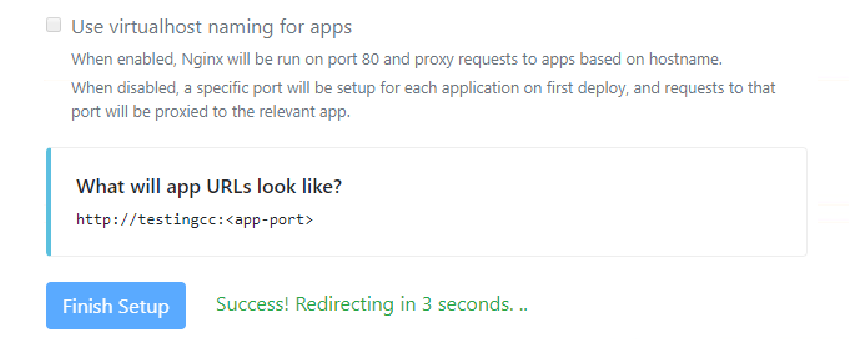
**Gambar 8.16** Hasil *generate SSH key*

Langkah berikutnya yakni simpan *Public Key* dan *Private Key* dengan cara klik tombol yang tersedia dan simpan di tempat yang aman (pada folder kerja Cloud Computing **Logbook 8**). Gunakan format penyimpanan pubkey\_dokku untuk ***Public Key*** dan privkey\_dokku.ppk untuk ***Private Key***. Kedua *key* tersebut akan digunakan untuk keperluan tahap berikutnya. Kemudian lakukan ***copy*** hasil *generate key* tersebut dan ***paste*** pada browser di kolom Public SSH Keys sebagaimana gambar berikut ini:



**Gambar 8.17** Proses *input* *Public SSH Key*

Pastikan kolom Hostname terisi sesuai dengan pengaturan di saat instalasi Ubuntu Server. Bilamana masih kosong, isi dengan format ubuntu-NIM. Kemudian pada bagian bawah **hilangkan** *checklist* pada *virtualhost* dikarenakan pada praktikum ini tidak menggunakan *virtualhost*, melainkan menggunakan *port* *based*. Terakhir klik tombol Finish Setup. Berikut tampilannya ketika berhasil menyelesaikan pengaturan:



**Gambar 8.18** Tampilan pengaturan SSH *key* telah berhasil

1. Setelah pengaturan tersebut, maka Dokku telah siap digunakan. Anda dapat membuat *container* untuk menambahkan aplikasi (contoh *Node.js*) maupun *plugin* (contoh *Postgres*) sesuai kebutuhan.

|  |
| --- |
| Modul VII Logbook |

## Logbook 1

Berdasarkan pemahaman Anda terhadap materi yang telah diajarkan, kerjakan soal berikut pada lembar kerja **Logbook 1** yang telah disediakan.

1. Apa itu *Cloud* *Computing*? Apa bedanya dan mengapa diperlukan dibandingkan komputasi konvensional (non *Cloud* *Computing*)?
2. Aplikasi apa yang digunakan untuk virtualisasi sistem operasi FreeNAS pada praktikum *Cloud* *Computing*?
3. Apa yang Anda ketahui dari FreeNAS? Apa fungsinya dan kelebihannya dibandingkan media *sharing* lokal (*Samba*, *File* *Sharing*, *ShareIT*, *Bluetooth*, dsb)?
4. Dengan proses instalasi dan terciptanya (*deploy*) FreeNAS pada VMware, maka akan terbentuk arsitektur *Cloud* seperti apa? Apakah termasuk *Private*/*Public*/*Community* *Cloud*? Jelaskan alasannya!
5. Ilustrasikan pada lembar kerja mengenai arsitektur *Cloud Computing* yang digunakan dalam praktikum (VMware dan FreeNAS) dimulai dari *layer hardware* atau komputer laboratorium hingga keterkaitannya dengan *layer* sistem operasi virtual yakni FreeNAS.

## Logbook 2

Berdasarkan pemahaman Anda terhadap penggunaan dan konfigurasi FreeNAS yang telah diajarkan, kerjakan studi kasus berikut pada komputer laboratorium. Ketentuan pengerjaan:

1. Simpan pada drive **D** (**Data Lab**). Buat folder baru dengan nama “**Latihan Cloud Computing”** atau gunakan folder yang sudah ada bila telah tersedia. Lanjutkan dengan pembuatan folder “**NIM-Logbook2**” di dalam folder “Latihan Cloud Computing”. Simpan pekerjaan Anda di dalam folder tersebut.
2. Kerjakan dalam waktu 30 menit dan patuhi petunjuk soal yang telah diberikan. Anda dapat membuat sendiri pemisalan/asumsi seperti penamaan dan kata sandi bila tidak dijelaskan pada soal.
3. Bila pekerjaan telah selesai, kerjakan **Logbook 2** pada lembar yang telah disediakan dalam waktu 15 menit.
4. Setelah pengisian Logbook selesai, segera ajukan penilaian pada asisten laboratorium untuk diberikan penilaian.

Berikut adalah soal yang harus Anda kerjakan pada komputer laboratorium:

Perusahaan Maju Mundur memiliki masalah dengan pembagian alokasi media penyimpanan pada aplikasi FreeNAS yang mereka gunakan. Terdapat beberapa divisi yang akan menggunakan FreeNAS namun dengan berbagai ketentuan yang telah dirumuskan. Dapatkah Anda sebagai seorang Administrator FreeNAS membantu perusahaan Maju Mundur dengan ketentuan sebagai berikut:

1. Terdapat 5 user (divisi) dengan username dan password yang berbeda-beda dan memiliki jatah storage mereka secara mandiri.
2. Kapasitas maksimal storage yang dimiliki perusahaan hanya sebesar 25 GB namun harus dibagi sesuai kebutuhan setiap divisi. Divisi Data Center membutuhkan kapasitas yang lebih dibandingkan divisi lain.

Berikut logbook yang harus Anda kerjakan pada **Logbook 2** yang telah disediakan:

1. Apakah semua divisi mendapatkan jatah storagenya sesuai ketentuan yang ada? Mengapa demikian?
2. Apa solusi yang dapat Anda sampaikan pada Perusahaan Maju Mundur blah blah

## Logbook 3

Berdasarkan praktikum yang anda lakukan, buatlah sebuah instalasi VMware vSphere ESXi dan mengkonfigurasinya. Setelah itu lakukan hal berikut:

1. Lakukan instalasi lebih dari satu (minimal 2) sistem operasi pada VMware anda.

2. Jelaskan bagaimana pembagian kegunaan dari OS yang anda install.

3. Bagaimana proses pengelolaannya?

4. Jika telah selesai, kerjakan pada **Logbook 3** yang telah disediakan.

## Logbook 4

Berdasarkan apa yang telah anda pelajari, jelaskan secara rinci tentang hal-hal berikut ini:

1. Apa kegunaan SaaS?

2. Sebutkan dan jelaskan minimal 3 layanan SaaS yang anda ketahui!

3. Jelaskan apa itu LAMPP!

4. Jelaskan kegunaan Apache!

5. Jelaskan kegunaan MySQL!

6. Jelaskan kegunaan PHP!

7. Jelaskan cara kerja LAMPP pada OS yang anda gunakan!

Kerjakan dalam waktu **15 menit** dan ditulis pada **Logbook 4** yang telah disediakan.

## Logbook 5

Berdasarkan praktikum yang telah anda lakukan, buatlah sebuah REST API seperti berikut:

1. Buatlah sebuah database **produk** yang memiliki id, nama barang, tempat pembuatan, tahun keluaran!

2. Buatlah fungsi CRUD untuk REST-API yang anda bangun!

3. Jelaskan tentang fungsi yang telah anda buat!

Kerjakan dalam waktu **20 menit** Tuliskan pada **Logbook** **5** yang telah disediakan.