

Pengenalan Linux dan Pemrograman Dasar

Detail Materi

Indikator :
Memahami fungsi utama dari server,
dapat mengoperasikan Linux dan
membuat bash scripting untuk
kebutuhan automasi environment



Mengetahui hubungan Linux
dan Server



Linux Begitu Penting di
Dunia DevOps



Terbiasa dengan Comman Line
di Linux



Setup Webserver dan
Web Aplikasi



Skill Pemrograman Dasar
Bash

Modul Sekolah DevOps Cilsy

Hak Cipta © 2020 **PT. Cilsy Fiolution Indonesia**

Hak Cipta dilindungi Undang-Undang. Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun, baik secara elektronik maupun mekanis, termasuk mecropy, merekam atau dengan sistem penyimpanan lainnya, tanpa izin tertulis dari Penulis dan Penerbit.

Penulis : Adi Saputra, Irfan Herfiandana & Tresna Widiyaman
Editor: Rizal Rahman, Tresna Widiyaman, & Muhammad Fakhri Abdillah
Revisi Batch 4

Penerbit : **PT. Cilsy Fiolution Indonesia**

Web Site : <https://cilsyfiolution.com> , <https://devops.cilsy.id>

Sanksi Pelanggaran Pasal 113 Undang-undang Nomor 28 Tahun 2014 tentang Hak Cipta

1. Setiap orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam pasal 9 ayat (1) huruf i untuk penggunaan secara komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan atau pidana denda paling banyak Rp100.000.000,00 (seratus juta rupiah).
2. Setiap orang yang dengan tanpa hak dan atau tanpa izin pencipta atau pemegang hak cipta melakukan pelanggaran hak ekonomi pencipta sebagaimana dimaksud dalam pasal 9 ayat (1) huruf c, huruf d, huruf f, dan atau huruf h, untuk penggunaan secara komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah)
3. Setiap orang yang dengan tanpa hak dan atau tanpa izin pencipta atau pemegang hak melakukan pelanggaran hak ekonomi pencipta sebagaimana dimaksud dalam pasal 9 ayat (1) huruf a, huruf b, huruf e, dan atau huruf g, untuk penggunaan secara komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan atau pidana denda paling banyak Rp1.000.000.000,00 (satu miliar rupiah)
4. Setiap orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan atau pidana denda paling banyak Rp4.000.000.000,00 (empat miliar rupiah)

Daftar Isi

Cove.....	1
Daftar Isi.....	3
3. Pengenalan Linux dan Pemrograman Dasar.....	9
Learning Outcomes.....	9
Outline Materi.....	9
3.1. Mengetahui Server.....	10
3.1.1. Penjelasan Konsep.....	10
3.1.2. Exercise.....	12
3.2. Mengetahui Linux (GNU/Linux).....	12
3.2.1. Mengetahui Sistem Operasi.....	12
3.2.2. Apa itu Linux?.....	14
3.2.3. Kenapa Linux bisa cocok untuk Server?.....	15
3.2.4. Distribusi/Distro Linux.....	16
3.2.5. Distro Terkenal di Indonesia.....	17
3.2.6. Parent Distro.....	17
3.2.7. CentOS.....	18
3.2.8. Ubuntu.....	18
3.2.9. Exercise.....	20
3.3. Instalasi Vagrant pada Sistem Operasi Linux Ubuntu.....	20
3.3.1. Instalasi Virtual Box.....	21
3.3.2. Instalasi Vagrant.....	24
3.3.3. Instalasi Box Vagrant.....	24
3.3.4. Membuat Project.....	26
3.4. Struktur Direktori GNU/Linux.....	31
3.4.1. Apa itu struktur direktori?.....	31
3.4.2. Penjelasan Struktur Direktori.....	32
3.4.3. Penulisan Struktur Direktori.....	34

3.4.3.1. Penulisan Alamat Lengkap.....	35
3.4.3.2. Penulisan Alamat berdasarkan posisi kita berada.....	37
3.4.4. Exercise.....	40
3.5. Perintah-perintah dasar Command Line Linux.....	41
3.5.1. Apa itu Command Line?.....	41
3.5.2. Remote Linux.....	41
3.5.3. Belajar Login Logout.....	43
3.5.4. Perintah Navigasi.....	44
3.5.5. Perintah pengelolaan Direktori & File.....	45
3.5.6. Perintah Editing File.....	46
3.5.7. Perintah Bantuan Manual Guide.....	47
3.5.8. Exercise.....	47
3.6. Belajar manipulasi Teks & File.....	48
3.6.1. Apa itu manipulasi teks & file?.....	48
3.6.2. Belajar redirection & Pipeline.....	50
3.6.2.1. Apa itu Redirection & Pipeline?.....	50
3.6.2.2. Contoh Penggunaan Redirection.....	50
3.6.2.3. Contoh Pipeline.....	51
3.6.3. Belajar grep.....	52
3.6.3.1. Apa itu Grep?.....	52
3.6.3.2. Menampilkan Beberapa baris setelah atau sebelum pola yang dicari.....	52
3.6.3.3. Menampilkan teks dengan pola kata kunci tertentu.....	53
3.6.3.4. Membalik Hasil yang muncul untuk memfilter.....	54
3.6.4. Exercise.....	55
3.6.5. Belajar Regular Expression.....	56
3.6.5.1. Apa itu Regular Expression?.....	56
3.6.5.2. Penggunaan (.).....	57
3.6.5.3. Penggunaan (*).....	58
3.6.5.4. Penggunaan +.....	59
3.6.5.5. Penggunaan ?.....	59

3.6.5.6. Penggunaan ^ dan 	59
3.6.5.7. Penggunaan \$ dan [karakter].....	60
3.6.6. Exercise.....	60
3.6.7. Belajar sed.....	62
3.6.7.1. Apa itu Sed?.....	62
3.6.7.2. Contoh penggunaan Sed.....	62
3.6.8. Belajar cut.....	64
3.6.8.1. Apa itu Cut?.....	64
3.6.8.2. Contoh penggunaan Cut.....	64
3.6.9. Belajar Sort & Uniq.....	65
3.6.9.1. Apa itu Sort & Uniq?.....	65
3.6.9.2. Contoh penggunaan Sort dan Uniq.....	65
3.6.10. Exercise.....	66
3.7. Belajar archiving.....	68
3.7.1. Apa itu Archiving?.....	68
3.7.2. Jenis dan Opsi kompresi.....	68
3.7.3. Melakukan Kompresi.....	70
3.7.4. Melihat isi file yang sudah di kompres.....	70
3.7.5. Mengupdate dan menghapus file didalam Arsip.....	70
3.7.6. Mengekstrak File.....	71
3.7.7. Exercise.....	71
3.8. Belajar finding files.....	71
3.8.1. Apa itu Finding Files?.....	71
3.8.2. Mencari file berdasarkan nama dan ekstensi.....	72
3.8.3. Mencari file berdasarkan tipe dan ukuran.....	72
3.8.4. Mencari file berdasarkan waktu dimodifikasi.....	72
3.8.5. Mencari file berdasarkan permission.....	73
3.8.6. Exercise.....	73
3.9. Belajar Manajemen User & Group.....	73
3.9.1. Apa itu manajemen user & group?.....	73
3.9.2. Perintah-perintah manajemen User.....	76

3.9.2.1. Membuat User Baru.....	76
3.9.2.2. Memberi/mengganti password user.....	76
3.9.2.3. Mengganti nama user.....	76
3.9.2.4. Menghapus User.....	76
3.9.2.5. Disable/lock user.....	76
3.9.2.6. Unlock user.....	76
3.9.2.7. Perintah-perintah manajemen user lainnya.....	77
3.9.3. Perintah-perintah manajemen Group.....	77
3.9.3.1. Membuat Group Baru.....	77
3.9.3.2. Merubah nama Group.....	77
3.9.3.3. Menghapus Group.....	77
3.9.3.4. Menampilkan group suatu user.....	77
3.9.3.5. Menambahkan user ke suatu grup.....	77
3.9.3.6. Mengeluarkan user dari suatu grup.....	78
3.9.4. Exercise.....	78
3.10. Belajar Permission.....	79
3.10.1. Apa itu Permission?.....	79
3.10.2. Penjelasan Konsep Permission.....	79
3.10.2.1. Aturan Umum Permission.....	79
3.10.2.2. Cara Menghitung Nilai Oktal Permission.....	81
3.10.3. Praktek Permission.....	83
3.11. Belajar Manajemen Paket.....	84
3.11.1. Apa itu Manajemen Paket?.....	84
3.11.2. Melakukan Update Repository.....	84
3.11.3. Melakukan Instalasi Aplikasi/Service.....	85
3.11.4. Mengecek apakah aplikasi sudah terinstall atau belum.....	85
3.11.5. Menghapus Aplikasi/Service.....	85
3.11.6. Mencari Aplikasi dengan kata kunci tertentu.....	86
3.11.7. Upgrade aplikasi.....	86
3.11.8. Exercise.....	86
3.12. Belajar Manajemen Layanan.....	87

3.12.1. Apa itu Manajemen Layanan?	87
3.12.2. Mengecek Status Layanan	87
3.12.3. Merestart Layanan	88
3.12.4. Menghentikan Layanan	88
3.12.5. Menjalankan Layanan	88
3.12.6. Mengecek & Membuat layanan aktif/tidak saat booting	89
3.12.7. Exercise	90
3.13. Roadmap Topologi yang akan dibangun	90
3.13.1. Roadmap Pembelajaran	90
3.13.2. Topologi yang akan dibangun	91
3.14. Setup Webserver dan Aplikasi	92
3.14.1. Install Webserver dan Database Server	92
3.14.2. Setup Web Aplikasi Sosial Media	94
3.15. Pemrograman Dasar Shell	98
3.15.1. Apa itu Shell ?	98
3.15.2. Pemrograman Shell	98
3.15.3. Perintah Dasar Shell	99
3.15.4. Bash Scripting Sederhana	99
3.15.5. Pemakaian Variable	101
3.15.5.1. Positional Parameter	101
3.15.5.2. User Defined Variable	102
3.15.6. Exercise	105
3.15.7. Belajar Logika Pemrograman pada Bash Script	106
3.15.7.1. If Conditional	106
3.15.7.2. While Looping	109
3.15.7.3. For Looping	111
3.15.8. Exercise	114
3.16. Membuat Automasi Setup Webserver dengan Bash	114
3.16.1. Automasi instalasi webserver	114
3.16.2. Automasi Setup Web Aplikasi	115
3.16.3. Exercise	117

3.17. Summary.....	117
--------------------	-----

3.

Pengenalan Linux dan Pemrograman Dasar

Learning Outcomes

Setelah selesai mempelajari bab ini, peserta mampu :

1. Mengetahui apa hubungan Linux dan Server.
2. Memahami kenapa Linux begitu penting dalam dunia DevOps.
3. Terbiasa menggunakan command line dalam melakukan berbagai hal di Server Linux.
4. Dapat melakukan setup webserver dan Aplikasi di Server Local
5. Mempunyai skill dasar untuk melakukan scripting dan pemrograman dasar.

Outline Materi

1. Belajar apa itu Server, Sistem Operasi, dan Linux
2. Struktur direktori di Linux
3. Dasar-dasar command line Linux
4. Manipulasi Teks dan File
5. Archiving & Finding File
6. Manajemen User & Grup
7. Permission
8. Manajemen Paket dan Manajemen Layanan di Linux
9. Setup Webserver dan Web Aplikasi


10. Pemrograman Dasar Shell

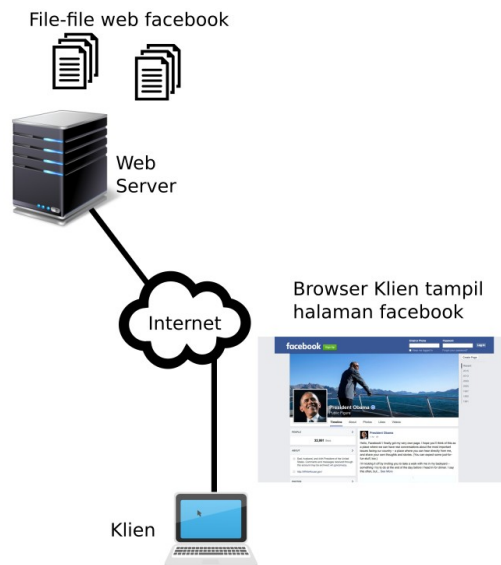
3.1. Mengenal Server

3.1.1. Penjelasan Konsep

Pada pembahasan bab sebelumnya, sering sekali disebut istilah Server dan Linux. Kedua hal ini kaitannya memang sangat erat dengan DevOps. Dan kedua hal ini adalah hal yang juga sangat berkaitan.

Pada bab ini kita akan fokus mempelajari terkait Linux. Apa itu Linux, kenapa harus Linux, untuk apa Linux, bagaimana pengaplikasiannya, dll. Akan tetapi, sebelum kita mempelajari apa itu Linux dan bagaimana penggunaannya pada dunia DevOps, kita harus mengenal dulu tentang Server.

Server sebenarnya hanyalah sebuah komputer sama seperti komputer pada umumnya. Namun bedanya, komputer server memiliki tujuan untuk memberikan suatu layanan yang bisa dinikmati oleh pengguna. Layanan ini bisa bermacam-macam bentuknya. Ada layanan web, layanan email, layanan database, layanan akses remote jarak jauh, dll. Misalnya kalau kita mau punya website seperti Bhinneka.com, maka itu perlu layanan web. Istilahnya adalah Web Server. Misalnya kalau mau punya aplikasi untuk kirim terima email seperti , maka itu perlu layanan email. Istilahnya adalah Mail Server. Begitu seterusnya.



Ilustrasi Web Server Facebook

Karena diperuntukkan untuk melayani pengguna, dan biasanya jumlah pengguna sangat besar (bisa bayangkan website Bukalapak.com itu diakses berapa juta pengguna perhari?), maka spesifikasi dari komputer untuk server ini biasanya sangat bagus. Prosesornya misalnya memiliki 32 Core, RAM nya 32GB, dan Harddisknya 4TB. Inilah yang menyebabkan seakan-akan komputer server dengan komputer biasa itu berbeda, padahal sebenarnya sama. Sama-sama komputer. Bedanya hanyalah di spesifikasinya saja.

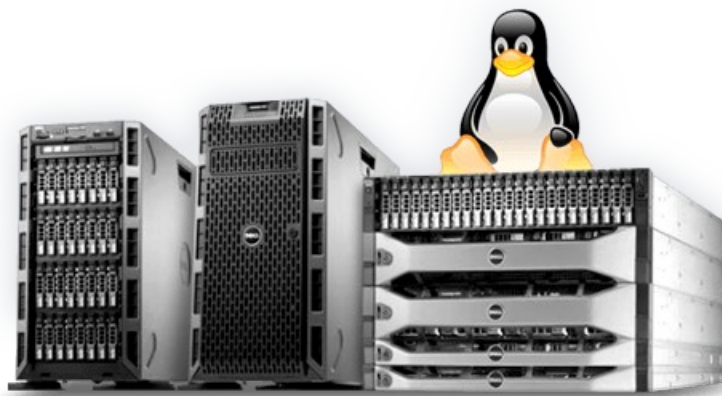


Dell PowerEdge R720 Rackmount Server

Server Rackmount

Seluruh kegiatan DevOps nantinya pasti berkaitan dengan Server ini. Seluruh aplikasi maupun web di perusahaan tempat DevOps bekerja, itu pasti berjalan di server. Misalnya Tokopedia.com yang memiliki aplikasi berbentuk android dan web. Maka sebenarnya aplikasi-aplikasi itu adalah bentuk “layanan” yang disediakan oleh Server. Ada komputer-komputer Server yang berjalan dibalik itu semua. Dan itulah mengapa DevOps pasti selalu berkuat dengan Server.

Lalu apa hubungannya Server dengan Linux? Seperti yang sudah dijelaskan sebelumnya, pada dasarnya Server hanyalah sebuah komputer. Sebuah komputer ketika ingin bisa berjalan, maka perlu yang namanya Sistem Operasi. Nah, Sistem Operasi yang paling cocok untuk Server adalah Linux dibanding dengan Sistem Operasi yang lain seperti MacOS atau Windows. Kita akan bahas ini secara mendalam pada bagian berikutnya. Intinya disini kita cukup mengetahui bahwa seorang DevOps pasti berkuat selalu dengan yang namanya Server. **Dan Sistem Operasi yang paling cocok untuk Server, adalah Linux.**




Server menggunakan OS Linux

Sebagai tambahan, Server tidak harus selalu berbentuk komputer fisik. Karena ada Server yang bentuknya virtual/cloud. Server virtual/cloud ini kita tidak pernah mengetahui bentuk fisiknya seperti apa, akan tetapi secara spesifikasi

sudah sesuai dengan komputer server berbentuk fisik pada umumnya, yaitu memiliki RAM, CPU, dan harddisk tertentu.

3.1.2. Exercise

1. Apa hubungan DevOps dengan Server? 
2. Sebutkan Layanan-layanan server lainnya yang pernah Anda ketahui?

3.2. Mengenal Linux (GNU/Linux)

3.2.1. Mengenal Sistem Operasi

Sebuah komputer atau sebuah Server pada dasarnya hanyalah sekumpulan perangkat keras yang belum bisa dipakai. Nah agar seluruh komponen perangkat keras pada komputer bisa bekerja dan dinikmati oleh manusia, maka perlu adanya sebuah perantara untuk menjembatani. Perantaranya ini disebut Sistem Operasi. Sistem Operasi membuat seluruh komponen perangkat keras pada komputer bekerja untuk kemudian menyediakan tampilan dan fungsi-fungsi yang bisa dinikmati oleh pengguna. Seluruh tampilan di laptop kita yang bagus dengan berbagai fungsi dan aplikasi yang kita pakai sehari-hari, itu adalah peran dari Sistem Operasi.



Contoh dari Sistem Operasi yang paling terkenal adalah Windows, MacOS dan Linux. Masing-masing Sistem Operasi didesain dengan peruntukannya masing-masing dengan berbagai kekuatan dan kelemahannya.

Biasanya Windows lebih cocok untuk keperluan penggunaan kegiatan sehari-hari. Seperti bermain game, untuk ketik-ketik dokumen, untuk bekerja, dll. MacOS lebih untuk kegiatan terkait desain. Biasanya para desainer, pembuat film, pembuat musik lebih suka menggunakan Mac OS. Nah Linux, ini lebih cocok untuk dipakai di Server. Yaitu untuk memberikan layanan-layanan kepada pengguna.

Ibarat mobil, masing-masing merk mobil pun ada peruntukannya masing-masing. Misalnya Toyota untuk keperluan mobil keluarga sehari-hari, Limousin untuk keperluan gaya pariente, hingga Range Rover untuk keperluan off-road. Begitupun pada Sistem Operasi.

3.2.2. Apa itu Linux?

*Logo Linux*

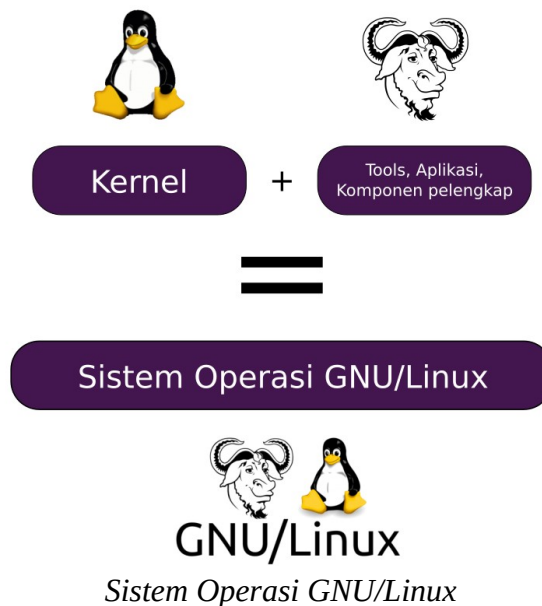
Pada materi sebelumnya kita sudah bahas bahwa Linux adalah sebuah Sistem Operasi yang cocok untuk Server. Pengertian ini sebenarnya hanya untuk memberikan gambaran umum saja, karena pengertian tersebut tidak sepenuhnya benar.

Lalu sebenarnya apa itu Linux? Linux lebih tepatnya hanyalah sebuah kernel, bukan sistem operasi itu sendiri. **Kernel ini adalah inti atau mesin dari sistem operasi.** Ibarat motor, kernel ini adalah mesin motor, bukan keseluruhan dari si motornya itu sendiri.

Kernel Linux pertama kali dikembangkan oleh seorang mahasiswa bernama Linus Torvalds pada tahun 1991. Dan terus berkembang pesat sejak saat itu hingga sekarang karena seluruh kode sumber dari Linux dapat diakses dan dimodifikasi secara bebas tanpa membayar sepeserpun. Ini sangat berbeda dengan Windows maupun Mac OS yang kode sumbernya sama sekali tidak boleh diakses dan diubah-ubah tanpa seijin mereka.

Bagaimana Linux bisa berjalan dengan baik kalau hanya terdiri dari mesinnya saja? Bukankah sepeda motor tidak akan bisa berjalan kalau hanya ada mesinnya saja (tidak ada stang, ban, kursi, dll)?

Disinilah peran yang namanya GNU. GNU ini adalah sebuah proyek dari seorang bernama Richard Stallman berupa sekumpulan tools, software, aplikasi, dan komponen-komponen pelengkap sebuah Sistem Operasi. GNU inilah ibaratnya stang, kursi, roda, dll untuk melengkapi mesin sehingga jadilah sebuah sepeda motor yang utuh.



Kesimpulannya Linux bukanlah sistem operasi. Namun ketika Linux ditambah GNU, barulah menjadi sebuah sistem operasi yang lengkap dan bisa digunakan. Sistem Operasi gabungan GNU dan Linux yang lengkap disebut GNU/Linux. Sayangnya karena penyebutan GNU/Linux terlalu susah, akhirnya orang-orang lebih suka menyebutnya Linux saja.

3.2.3. Kenapa Linux bisa cocok untuk Server?

Berikut kira-kira beberapa faktor mengapa Linux paling cocok digunakan untuk Server :

1. Lebih aman
2. Lebih stabil
3. Banyaknya tools dan aplikasi yang tersedia untuk kebutuhan Server
4. Mayoritas tools dan aplikasi yang tersedia di Linux itu Free.
5. Banyaknya dokumentasi yang tersebar di internet
6. Besarnya forum yang siap membantu menjawab dan mendokumentasikan segala hal tentang Linux.



Seluruh poin diatas hanya disebabkan oleh 1 hal. Karena kode sumber Linux yang bebas untuk diakses dan di modifikasi.

Hal inilah yang menyebabkan tumbuhnya basis pengembang aplikasi dan tools di Linux yang sangat besar di seluruh dunia. Mulai dari perorangan hingga perusahaan-perusahaan besar macam Intel, Oracle, Google, hingga Samsung juga turut berkontribusi. Mereka menyukai pembuatan aplikasi di Linux karena kebebasan modifikasinya. Mereka bisa turut berkontribusi bersama-sama untuk mengembangkan aplikasi, memperbaiki, hingga menggunakannya. Semuanya secara bebas.

Company	Changes	Total	Company	Changes	Total
None	12,550	13.6%	NVidia	1,192	1.3%
Red Hat	9,483	10.2%	Freescale	1,127	1.2%
Intel	8,108	8.8%	Ingics Technology	1,075	1.2%
Texas Instruments	3,814	4.1%	Renesas Electronics	1,010	1.1%
Linaro	3,791	4.1%	Qualcomm	965	1.0%
SUSE	3,212	3.5%	Cisco	871	0.9%
Unknown	3,032	3.3%	The Linux Foundation	840	0.9%
IBM	2,858	3.1%	Academics	831	0.9%
Samsung	2,415	2.6%	AMD	820	0.9%
Google	2,255	2.4%	Inktank Storage	709	0.8%
Vision Engraving Systems	2,107	2.3%	NetApp	707	0.8%
Consultants	1,529	1.7%	LINBIT	705	0.8%
Wolfson Microelectronics	1,516	1.6%	Fujitsu	694	0.7%
Oracle	1,248	1.3%	Parallels	684	0.7%
Broadcom	1,205	1.3%	ARM	664	0.7%

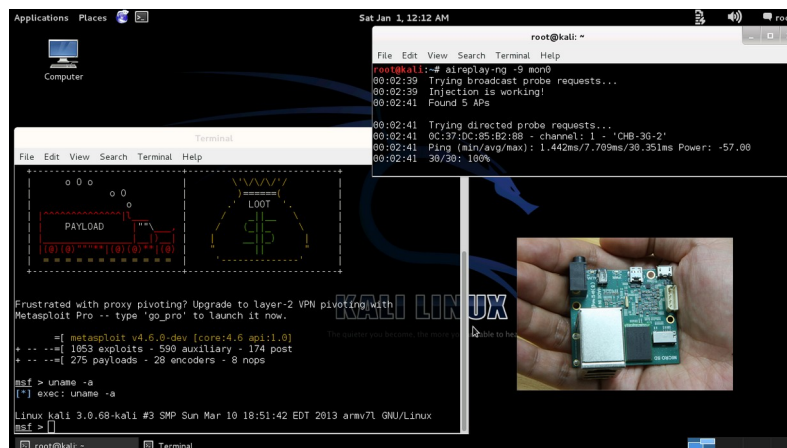
Pengembang Linux dalam Angka

Saking banyaknya orang dan perusahaan yang berkontribusi, akibatnya ketika ada sebuah virus maupun celah keamanan yang muncul di Linux akan sangat cepat di perbaiki. Ini yang menyebabkan Linux lebih aman, karena hampir tak ada kesempatan bagi hacker untuk menyerang Linux dengan sukses. Mereka sudah capek-capek membobol, tapi dalam waktu yang singkat berhasil di perbaiki. Membuat usaha mereka tidak sebanding dengan hasilnya. Untuk apa riset mencari celah berbulan-bulan tapi berhasil diperbaiki hanya dalam hitungan jam?

3.2.4. Distribusi/Distro Linux

Distribusi Linux atau yang sering disebut sebagai distro Linux, merupakan sistem operasi GNU/Linux yang sudah di modifikasi untuk tujuan penggunaan tertentu.

Misalnya ada Distro Kali Linux yang cocok untuk tujuan hacking, atau ada distro **Linux Mint** yang cocok untuk tujuan kegiatan sehari-hari (printing, office, dll), atau ada **distro Centos** yang lebih cocok untuk digunakan sebagai Server, hingga Android yang merupakan distro untuk kebutuhan Smartphone.



Contoh distro Kali Linux untuk Hacking

Analogi Distro ini bisa diibaratkan tipe sepeda motor. Misalnya Honda Beat, sepeda motor yang tujuannya untuk penggunaan sehari-hari keliling kompleks saja. Atau ada Kawasaki Ninja, yaitu sepeda motor yang tujuannya untuk balapan. **Sepeda Motornya itu sendiri adalah Sistem Operasi GNU/Linux, nah segala macam modifikasi aksesoris, stiker motornya, letak spionnya, model bodynya, sehingga terciptalah tipe motor Honda Beat, itulah Distro.**

3.2.5. Distro Terkenal di Indonesia

Beberapa distro yang populer di Indonesia adalah **Debian, Ubuntu, Fedora, OpenSuse, Blankon, Linux Mint, hingga Android.**

3.2.6. Parent Distro

Distro Linux ini jumlahnya ada ribuan di seluruh dunia. Akan tetapi seluruh distro yang ada di dunia sejauh sebenarnya hanya merujuk pada 3 buah distro besar, yaitu :

1. Debian
2. RedHat
3. Slackware

Secara konsep, ketiga distro diatas dengan seluruh distro yang ada diseluruh dunia adalah sama. Sekalipun ada perbedaan, itu hanya beda nama dan teknik saja. Sehingga jika kita sudah menguasai salah satu distro, maka ketika pindah ke distro lain itu tidak akan ada masalah. Cukup melakukan beberapa penyesuaian saja.

Agar lebih fokus, distro yang digunakan selama pembelajaran Sekolah DevOps Cilsy menggunakan distro CentOS dan Ubuntu.

3.2.7. CentOS



Logo CentOS

CentOS (Community ENTERprise Operating System) merupakan Distro Linux yang cocok dipergunakan dalam skala Server Enterprise selain itu juga gratis. CentOS dikembangkan oleh sebuah komunitas yang disebut CentOS Project, yang di buat benar-benar persis dari kode sumber distro Red Hat Enterprise (RHEL). Distro RHEL ini sendiri adalah distro yang cukup disegani dan cukup banyak dipakai di perusahaan-perusahaan enterprise karena dikenal akan

kestabilan dan kehandalannya serta didukung tim support yang mumpuni. Sayangnya RHEL ini adalah distro yang berbayar.

CentOS hadir sebagai solusi dimana orang yang ingin merasakan kualitas distro yang stabil dan handal persis seperti RHEL namun dengan harga yang gratis.

CentOS dirilis setiap versi RHEL juga dirilis. Biasanya selang 2-3 bulan setelah versi RHEL terbaru dirilis, CentOS juga akan ikut dirilis.

Paket aplikasi-aplikasi yang digunakan di CentOS menggunakan ekstensi .rpm.

3.2.8. Ubuntu



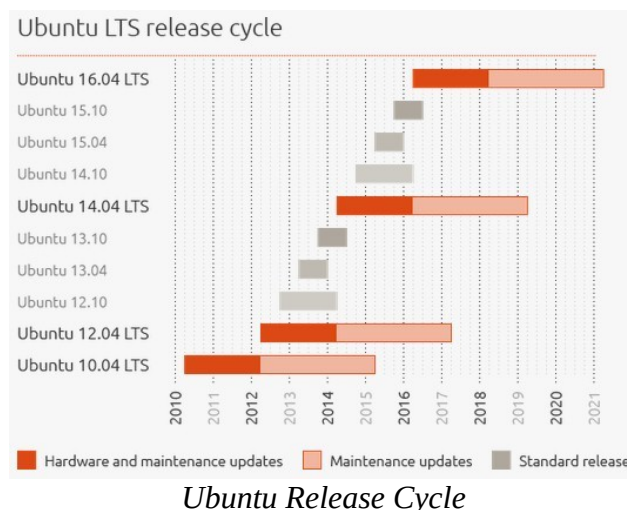
Ubuntu merupakan salah satu distro Linux berbasis distro Debian dan didistribusikan sebagai distro yang bebas dan gratis. Nama Ubuntu berasal dari filosofi dari Afrika Selatan yang berarti "kemanusiaan kepada sesama". Ubuntu dirancang untuk kepentingan penggunaan pribadi, namun versi server Ubuntu juga tersedia, dan juga telah dipakai secara luas.

Proyek Ubuntu resmi disponsori oleh Canonical Ltd. yang merupakan sebuah perusahaan yang dimiliki oleh pengusaha Afrika Selatan Mark Shuttleworth. Tujuan dari distribusi Linux Ubuntu adalah membawa semangat yang terkandung di dalam filosofi Ubuntu ke dalam dunia perangkat lunak. Ubuntu adalah sistem operasi lengkap berbasis Linux, tersedia secara bebas, dan mempunyai dukungan baik yang berasal dari komunitas maupun tenaga ahli profesional.

Ubuntu merupakan distro yang cukup populer di dunia karena dikenal akan kemudahan, ringan, handal, dan banyaknya support di internet.

Ubuntu memiliki dua buah versi rilis yang berbeda, yaitu versi LTS (Long Term Support) dan versi biasa. Letak perbedaan kedua versi ini adalah pada jangka waktu support yang disediakan oleh Canonical. Versi LTS disupport selama 3 tahun untuk versi Desktop dan 5 tahun untuk versi Server. Namun semenjak versi Ubuntu 12.04 LTS, kedua versi sama-sama disupport selama 5 tahun. Versi biasa hanya disupport selama 18 bulan atau 1,5 tahun.

Versi LTS lebih ditujukan kepada pengguna Server, karena penggarapan versi LTS ini lebih diutamakan pada kestabilan dan kehandalan sistem. Versi LTS di rilis tiap 2 tahun sekali, sedangkan versi biasa dirilis tiap 6 bulan sekali.



3.2.9. Exercise

1. Jelaskan kembali perbedaan Linux, GNU, Sistem Operasi, dan Distro?
2. Kenapa Linux lebih cocok untuk server dibanding sistem operasi lain?



3.3. Instalasi Vagrant pada Sistem Operasi Linux Ubuntu

Ketika kita melakukan development terhadap sebuah aplikasi terkadang kita membutuhkan tool - tool penting untuk membantu pada developer untuk membuat sebuah aplikasi. Ada beberapa masalah yang sering ditemui oleh seorang developer diantaranya adalah :

- Kesulitan ketika antar developer menggunakan sistem operasi yang berbeda.
- Kesulitan ketika server yang digunakan menggunakan sistem operasi yang berbeda.
- Kesulitan ketika project yang sedang dikerjakan tidak dapat berjalan di PC yang lain.

Dari beberapa kesulitan tersebut kita dapat meminimalkannya dengan menggunakan **Vagrant**. Apa itu vagrant ?

Vagrant adalah sebuah software yang menggunakan teknologi virtual machine dimana kita dapat membuat lingkungan development secara portable, konsisten dan lebih fleksible.



Dikarenakan vagrant menggunakan teknologi virtual machine maka kita membutuhkan software seperti virtual box dan VmWare. Tujuannya adalah kita ingin membuat sebuah lingkungan development secara portable, contohnya misalnya pada saat production kita akan menggunakan sistem operasi ubuntu maka pada saat development kita akan menggunakan ubuntu sebagai sistem operasi sehingga pada saat proses deploy ke production diharapkan tidak ada lagi permasalahan yang muncul.

Ketika ingin melakukan duplikasi atau dibagikan kepada team developer yang lain maka kita tinggal menggunakan perintah vagrant sehingga secara otomatis vagrant akan melakukan konfigurasi pada PC masing - masing team developer sehingga mempermudah dalam development sebuah aplikasi.

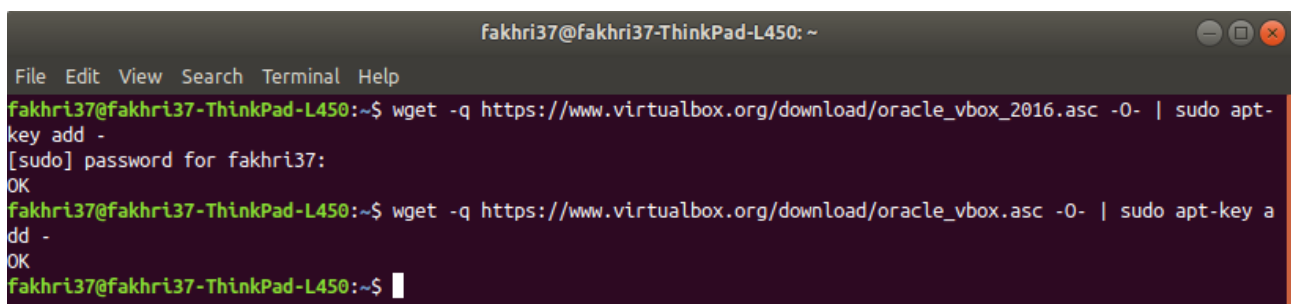
3.3.1. Instalasi Virtual Box

Pada modul ini, kita akan menggunakan virtual box dikarenakan virtual box merupakan software open source dan lebih mudah dikonfigurasi. Pertama-tama, buka terminal pada OS Ubuntu Anda dengan menekan tombol CTRL + ALT + T.

Setelah terminal terbuka masukkan public key dengan menjalankan perintah berikut.

```
wget -q https://www.virtualbox.org/download/oracle_vbox_2016.asc -O- | sudo apt-key add -
```

```
wget -q https://www.virtualbox.org/download/oracle_vbox.asc -O- | sudo apt-key add -
```

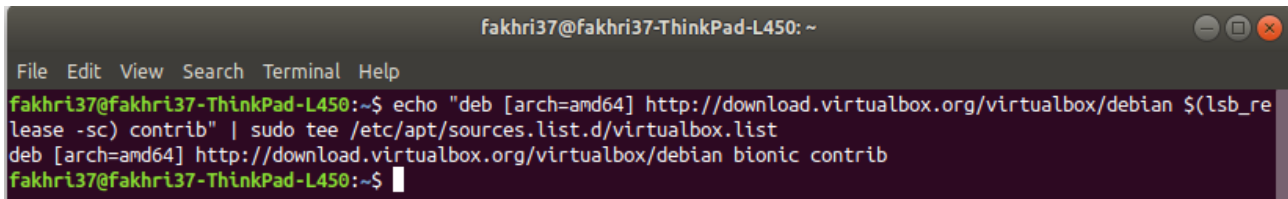


```
fakhri37@fakhri37-ThinkPad-L450: ~
File Edit View Search Terminal Help
fakhri37@fakhri37-ThinkPad-L450:~$ wget -q https://www.virtualbox.org/download/oracle_vbox_2016.asc -O- | sudo apt-key add -
[sudo] password for fakhri37:
OK
fakhri37@fakhri37-ThinkPad-L450:~$ wget -q https://www.virtualbox.org/download/oracle_vbox.asc -O- | sudo apt-key add -
OK
fakhri37@fakhri37-ThinkPad-L450:~$
```

Setelah itu tambahkan repository virtualbox menggunakan perintah berikut.



```
echo "deb [arch=amd64] http://download.virtualbox.org/virtualbox/debian $(lsb_release -sc) contrib" | sudo tee /etc/apt/sources.list.d/virtualbox.list
```

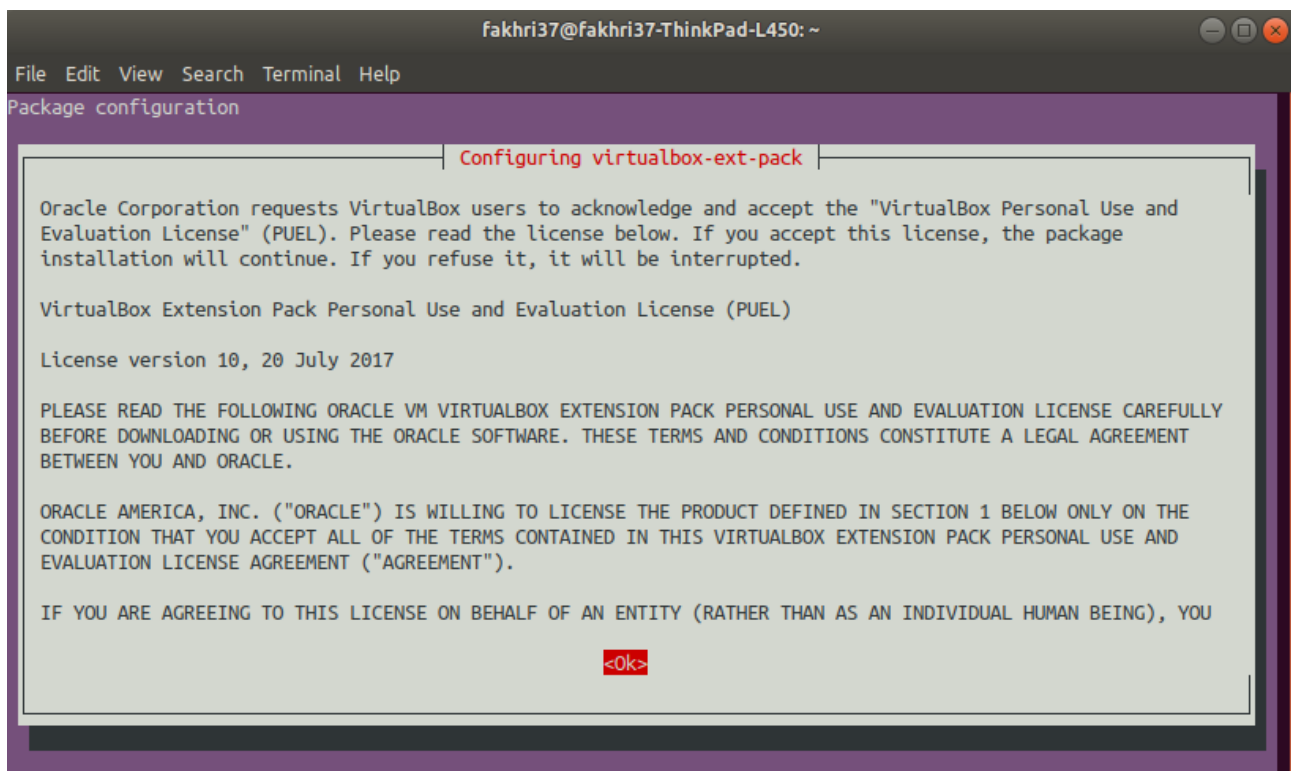


```
fakhri37@fakhri37-ThinkPad-L450: ~
File Edit View Search Terminal Help
fakhri37@fakhri37-ThinkPad-L450:~$ echo "deb [arch=amd64] http://download.virtualbox.org/virtualbox/debian $(lsb_release -sc) contrib" | sudo tee /etc/apt/sources.list.d/virtualbox.list
deb [arch=amd64] http://download.virtualbox.org/virtualbox/debian bionic contrib
fakhri37@fakhri37-ThinkPad-L450:~$
```

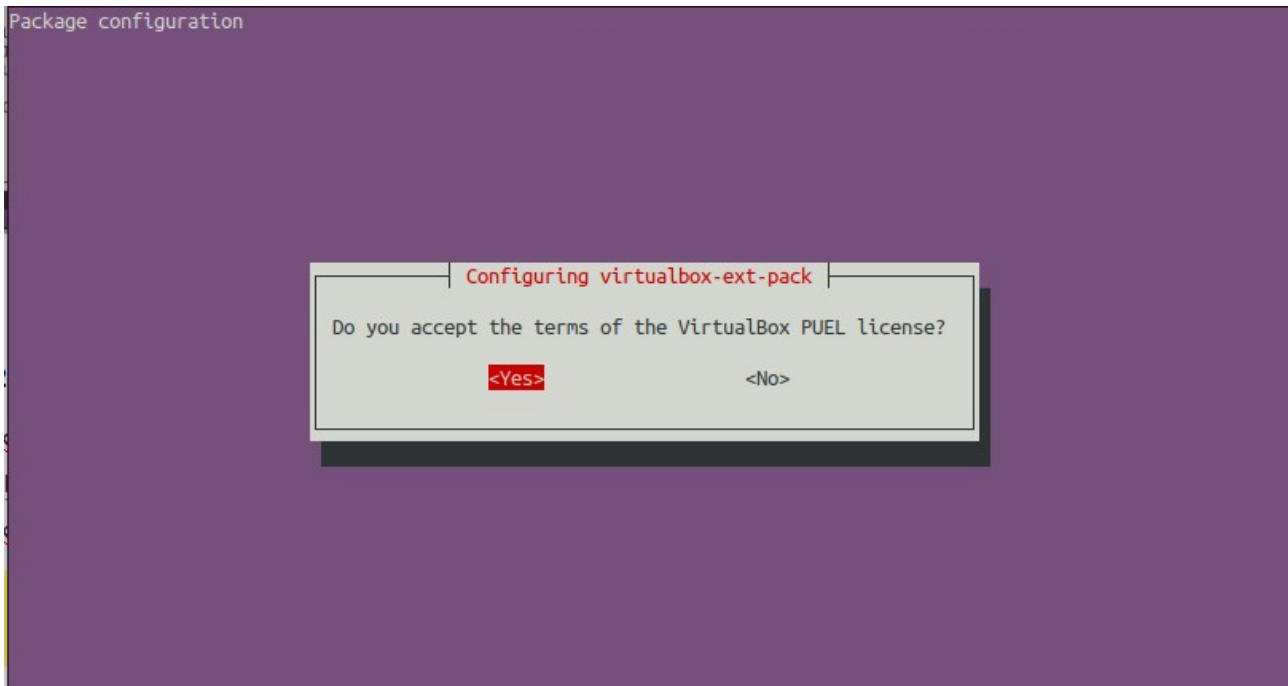
Kemudian gunakan perintah berikut untuk melakukan instalasi virtual box.

```
sudo apt update
sudo apt install linux-headers-$(uname -r) dkms
sudo apt install virtualbox virtualbox-ext-pack
```

Pilih OK pada user agreement license seperti pada gambar dibawah. Caranya adalah dengan menekan tombol TAB pada keyboard hingga tulisan OK terdapat highlight, kemudian tekan enter.



Pilih Yes pada prompt berikut, kemudian tunggu hingga instalasi selesai.



3.3.2. Instalasi Vagrant

Untuk melakukan instalasi vagrant silahkan download vagrant terlebih dahulu di [vagrant download](#), disini kita menggunakan ubuntu 18.04 64 bit maka kita pilih versi debian 64 bit. Untuk melakukan instalasinya silahkan jalankan perintah.

Sudo apt install curl ###(lakukan perintah ini untuk menginstallkan curl apabila belum terinstall)

curl -O https://releases.hashicorp.com/vagrant/2.2.6/vagrant_2.2.6_x86_64.deb

```
fakhri37@fakhri37-ThinkPad-L450:~$ curl -O https://releases.hashicorp.com/vagrant/2.2.6/vagrant_2.2.6_x86_64.deb
% Total    % Received % Xferd Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left  Speed
100 39.4M  100 39.4M    0     0  113k      0  0:05:54  0:05:54 --:--:-- 335k
fakhri37@fakhri37-ThinkPad-L450:~$
```

Setelah file .deb terdownload, gunakan perintah berikut untuk melakukan instalasi vagrant.

sudo apt install ./vagrant_2.2.6_x86_64.deb

```
fakhri37@fakhri37-ThinkPad-L450:~$ sudo apt install ./vagrant_2.2.6_x86_64.deb
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'vagrant' instead of './vagrant_2.2.6_x86_64.deb'
The following NEW packages will be installed:
  vagrant
0 upgraded, 1 newly installed, 0 to remove and 73 not upgraded.
Need to get 0 B/41,4 MB of archives.
After this operation, 117 MB of additional disk space will be used.
Get:1 /home/fakhri37/vagrant_2.2.6_x86_64.deb vagrant amd64 1:2.2.6 [41,4 MB]
Selecting previously unselected package vagrant.
(Reading database ... 196244 files and directories currently installed.)
Preparing to unpack .../vagrant_2.2.6_x86_64.deb ...
Unpacking vagrant (1:2.2.6) ...
Setting up vagrant (1:2.2.6) ...
fakhri37@fakhri37-ThinkPad-L450:~$
```

3.3.3. Instalasi Box Vagrant

Box pada vagrant berfungsi sebagai virtual machine yang memuat sistem operasi dan disana juga terdapat seluruh konfigurasi yang kita lakukan. Dengan demikian maka secara tidak langsung sebenarnya kita menggunakan virtual machine maka oleh karena itu kita membutuhkan virtual box sebagai tempat virtual machine.

Mengapa kita menggunakan vagrant ? mengapa tidak menggunakan virtual box saja ?

Salah satu alasannya adalah ketika menggunakan vagrant maka vagrant akan melakukan semua konfigurasi hanya dengan satu perintah pada saat sistem operasi start up sedangkan jika kita menggunakan virtual box maka kita harus melakukan konfigurasi secara manual.

Repository box bisa dilihat di [repository box](#). Berikut adalah bentuk umum untuk melakukan instalasi box.

```
vagrant box add user/box
```

bisa dilihat bahwa untuk melakukan instalasi box sangatlah mudah yaitu kita hanya menggunakan perintah diatas dan anda hanya perlu mengganti perintah `user/box`. Perintah `user` berarti adalah pembuat box sedangkan perintah `box` adalah box yang ingin kita gunakan. Disini kita akan menggunakan box `ubuntu/trusty64` maka jalankan perintah berikut.

```
vagrant box add ubuntu/trusty64
```

```
fakhri37@fakhri37-ThinkPad-L450:~$ vagrant box add ubuntu/trusty64
==> box: Loading metadata for box 'ubuntu/trusty64'
box: URL: https://vagrantcloud.com/ubuntu/trusty64
==> box: Adding box 'ubuntu/trusty64' (v20190514.0.0) for provider: virtualbox
box: Downloading: https://vagrantcloud.com/ubuntu/boxes/trusty64/versions/20190514.0.0/providers/virtualbox.box
box: Download redirected to host: cloud-images.ubuntu.com
box: Progress: 91% (Rate: 324k/s, Estimated time remaining: 0:01:43)
```

PERHATIAN !

Untuk melakukan instalasi box membutuhkan koneksi internet dan membutuhkan instalasi yang lama dikarenakan kita akan mendownload sebuah sistem operasi yang berkisar 1 GB > tergantung dari sistem operasi yang akan digunakan.

3.3.4. Membuat Project

Untuk membuat project, silahkan buat sebuah folder misalnya disini kita akan membuat folder `vagrantproject`. Lakukan perintah berikut untuk membuat direktori `vagrantproject`.

```
fakhri37@fakhri37-ThinkPad-L450:~$ mkdir vagrantproject
fakhri37@fakhri37-ThinkPad-L450:~$ cd vagrantproject/
fakhri37@fakhri37-ThinkPad-L450:~/vagrantproject$
```

Kemudian jalankan perintah berikut di dalam folder tersebut.

```
vagrant init
```

maka akan muncul output seperti berikut.

```
fakhri37@fakhri37-ThinkPad-L450:~/vagrantproject$ vagrant init
A 'Vagrantfile' has been placed in this directory. You are now
ready to 'vagrant up' your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
'vagrantup.com' for more information on using Vagrant.
fakhri37@fakhri37-ThinkPad-L450:~/vagrantproject$
```

Setelah proses diatas, maka akan muncul sebuah file bernama Vagrantfile. Kita bisa lihat dengan perintah

ls

```
GNU nano 2.9.3                                Vagrantfile

# -*- mode: ruby -*-
# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure("2") do |config|
  # The most common configuration options are documented and commented below.
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

  # Every Vagrant development environment requires a box. You can search for
  # boxes at https://vagrantcloud.com/search.
  config.vm.box = "base"

  # Disable automatic box update checking. If you disable this, then
  # boxes will only be checked for updates when the user runs
  # `vagrant box outdated`. This is not recommended.
  # config.vm.box_check_update = false

  # Create a forwarded port mapping which allows access to a specific port
  [ Read 70 lines ]
^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify
^X Exit          ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell
^C Cur Pos       ^_ Go To Line    ^- Undo         ^E Redo
```

Kemudian buka file `Vagrantfile` dengan perintah

```
nano Vagrantfile
```

Kemudian, ubah codingannya menjadi seperti berikut.

```
Vagrant.configure(2) do |config|
```

```
#konfigurasi box untuk sistem operasi ubuntu trusty 64 bit
config.vm.box = "ubuntu/trusty64"
```

```
config.vm.provider "virtualbox" do |vb|
```

```
# konfigurasi virtual box dengan ram 1 GB
```

```
vb.memory = "1024"
```

end

end

```
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure("2") do |config|
  # The most common configuration options are documented and commented below.
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

  # Every Vagrant development environment requires a box. You can search for
  # boxes at https://vagrantcloud.com/search.
  config.vm.box = "ubuntu/trusty64"

  # Disable automatic box update checking. If you disable this, then
  # boxes will only be checked for updates when the user runs
  # `vagrant box outdated`. This is not recommended.
  # config.vm.box_check_update = false
```

Setelah selesai, save dengan menekan tombol CTRL + X, lalu tekan Y untuk menyimpan. Selanjutnya, untuk menjalankan silahkan jalankan perintah berikut.

```
vagrant up
```

Maka akan tampil output berikut pada terminal.

```
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ubuntu/trusty64'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'ubuntu/trusty64' version '20190514.0.0' is up to
date...
==> default: Setting the name of the VM:
vagrantproject_default_1578556466493_14337
==> default: Clearing any previously set forwarded ports...
Vagrant is currently configured to create VirtualBox synced folders with
the `SharedFoldersEnableSymlinksCreate` option enabled. If the Vagrant
guest is not trusted, you may want to disable this option. For more
information on this option, please refer to the VirtualBox manual:

https://www.virtualbox.org/manual/ch04.html#sharedfolders

This option can be disabled globally with an environment variable:
```

```
VAGRANT_DISABLE_VBOXSYMLINKCREATE=1
```

or on a per folder basis within the Vagrantfile:

```
config.vm.synced_folder '/host/path', '/guest/path',
SharedFoldersEnableSymlinksCreate: false
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2222
    default: SSH username: vagrant
    default: SSH auth method: private key
    default:
    default: Vagrant insecure key detected. Vagrant will automatically replace
    default: this with a newly generated keypair for better security.
    default:
    default: Inserting generated public key within guest...
    default: Removing insecure key from the guest if it's present...
    default: Key inserted! Disconnecting and reconnecting using new SSH key...
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
    default: The guest additions on this VM do not match the installed version
of
    default: VirtualBox! In most cases this is fine, but in rare cases it can
    default: prevent things such as shared folders from working properly. If you
see
    default: shared folder errors, please make sure the guest additions within
the
    default: virtual machine match the version of VirtualBox you have installed
on
    default: your host and reload your VM.
```

```
default:
default: Guest Additions Version: 4.3.40
default: VirtualBox Version: 5.2
==> default: Mounting shared folders...
default: /vagrant => /home/fakhri37/vagrantproject
```

```
fakhri37@fakhri37-ThinkPad-L450:~/vagrantproject$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ubuntu/trusty64'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'ubuntu/trusty64' version '20190514.0.0' is up to date...
==> default: Setting the name of the VM: vagrantproject_default_1578556466493_14337
==> default: Clearing any previously set forwarded ports...
Vagrant is currently configured to create VirtualBox synced folders with
the 'SharedFoldersEnableSymlinksCreate' option enabled. If the Vagrant
guest is not trusted, you may want to disable this option. For more
information on this option, please refer to the VirtualBox manual:

  https://www.virtualbox.org/manual/ch04.html#sharedfolders

This option can be disabled globally with an environment variable:

  VAGRANT_DISABLE_VBOXSYMLINKCREATE=1

or on a per folder basis within the Vagrantfile:

  config.vm.synced_folder '/host/path', '/guest/path', SharedFoldersEnableSymlinksCreate: false
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
  default: Adapter 1: nat
==> default: Forwarding ports...
  default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
  default: SSH address: 127.0.0.1:2222
  default: SSH username: vagrant
  default: SSH auth method: private key
  default:
  default: Vagrant insecure key detected. Vagrant will automatically replace
  default: this with a newly generated keypair for better security.
  default:
  default: Inserting generated public key within guest...
  default: Removing insecure key from the guest if it's present...
  default: Key inserted! Disconnecting and reconnecting using new SSH key...
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
  default: The guest additions on this VM do not match the installed version of
  default: VirtualBox! In most cases this is fine, but in rare cases it can
  default: prevent things such as shared folders from working properly. If you see
```

Jika box telah jalan, langkah selanjutnya adalah login ke dalam box yang telah kita jalankan. Untuk login ke dalam box silahkan jalankan perintah berikut.

```
vagrant ssh
```

Jika berhasil maka akan muncul output pada terminal seperti berikut.


```
fakhri37@fakhri37-ThinkPad-L450:~/vagrantproject$ vagrant ssh
Welcome to Ubuntu 14.04.6 LTS (GNU/Linux 3.13.0-170-generic x86_64)

* Documentation:  https://help.ubuntu.com/

System information as of Thu Jan  9 07:54:55 UTC 2020

System load:  0.9               Processes:            79
Usage of /:   3.6% of 39.34GB    Users logged in:     0
Memory usage: 25%              IP address for eth0: 10.0.2.15
Swap usage:   0%

Graph this data and manage this system at:
https://landscape.canonical.com/

UA Infrastructure Extended Security Maintenance (ESM) is not enabled.

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

Enable UA Infrastructure ESM to receive 64 additional security updates.
See https://ubuntu.com/advantage or run: sudo ua status

New release '16.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

vagrant@vagrant-ubuntu-trusty-64:~$
```

Instalasi box pada vagrant telah berhasil. Sekarang kita bisa memulai belajar Linux tanpa takut error berpengaruh pada sistem operasi kita.

3.4. Struktur Direktori GNU/Linux

3.4.1. Apa itu struktur direktori?

Struktur direktori ini adalah ilmu untuk mempelajari seperti apa hierarki dan kegunaan dari masing-masing folder yang ada pada sistem operasi GNU/Linux.

Kita perlu mempelajari hal ini agar kita benar-benar tahu seperti apa isi dari sistem operasi GNU/Linux yang akan kita kelola saat menjadi DevOps nanti.

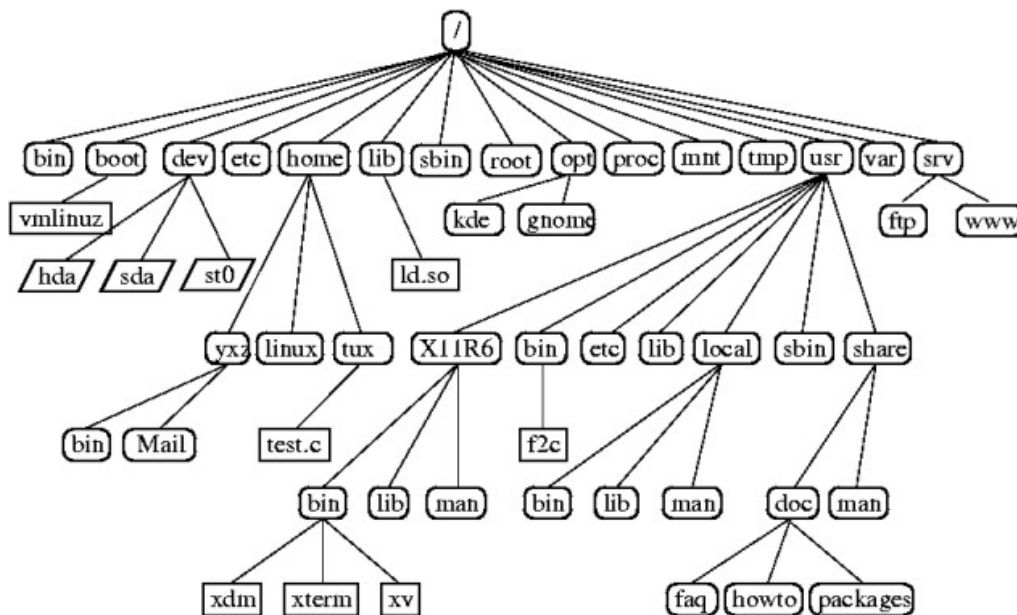
Kita harus tahu folder A untuk apa, folder B untuk menyimpan file apa, folder C sebaiknya tidak boleh dihapus, dst.

Misalnya jika di Windows, kita harus tahu bahwa letak aplikasi yang sudah diinstall itu ada di folder C:\Program Files, kalau mau simpan data di D:, kalau mau edit sistem maka ada di C:\Windows, dst. Begitupun di Linux, kita juga harus tahu ketika install aplikasi di Linux maka masuk ke folder apa, kalau mau edit konfigurasi web server ada dimana, kalau mau lihat data user ada dimana, dll.

Ibarat rumah, kita harus tau masing-masing ruangan di rumah tersebut fungsinya untuk apa. Agar kita tidak tersesat, agar kita tidak salah meletakkan barang, maupun agar kita tidak salah membuang barang yang ada didalamnya. Sebagai pemilik rumah, kita wajib menguasai seluk beluk ruangan dari rumah kita sendiri. Sebagai DevOps kita wajib mengetahui seluk beluk struktur direktori dari server Linux kita sendiri.

3.4.2. Penjelasan Struktur Direktori

Struktur direktori GNU/Linux sangat berbeda dengan struktur direktori windows/dos dimana dalam GNU/Linux tidak akan ditemukan drive a, drive c dan drive lainnya karena GNU/Linux menganut satu direktori utama yaitu direktori / (dibaca root). Dimana nantinya keseluruhan direktori dan file lainnya akan berada di dalam/dibawah direktori / ini.



Struktur direktori di Linux

Seluruh folder yang ada di bawah /, memiliki fungsi nya masing-masing sebagai berikut :

/	Dibaca root, merupakan direktori utama sistem Linux. Ibaratnya seperti C: pada windows.
/bin	Direktori yang berisi binary files yang dapat dieksekusi oleh users system. Direktori ini handle perintah-perintah standar GNU/Linux diantaranya ls, cp, mv dan beberapa lainnya.
/boot	Direktori yang berisikan file-file yang dibutuhkan saat proses booting termasuk didalamnya adalah kernel image GNU/Linux.
/dev	Direktori yang merepresentasikan/mewakili device hardware. Dalam GNU/Linux everything is files, dimana harddisk, cdrom, dll itu dianggap sebagai file. Semua device hardware komputer di representasikan sebagai file pada direktori ini.
/etc	Direktori yang berisi file-file konfigurasi system dan layanan. File-file konfigurasi ini dibutuhkan oleh paket-paket lain untuk menjaga

	perilaku system.
/home	Merupakan home direktori atau area kerja dari masing-masing user GNU/Linux kecuali user root(super user). User-user biasa menyimpan data disini.
/lib	Merupakan direktori file-file library GNU/Linux yang mendukung binary files baik yang berada didirektori /bin maupun /sbin.
/mnt	Direktori yang disiapkan untuk mounting point device storage seperti flashdisk dan harddisk eksternal.
/media	Direktori memounting/mengaitkan media removable storages seperti harddisk eksternal, floppy disk, cdrom/dvdrom, flashdisk, digital kamera maupun media lainnya. Mirip seperti /mnt.
/opt	Direktori tempat instalasi paket yang diluar dari repository resmi.
/proc	Merupakan direktori yang unik yang dinamis karena direktori ini berisi virtual file system yang menyediakan akses informasi cpu, ram, kernel, suhu, dll secara realtime.
/root	Home direktori khusus super user (user root). Direktori ini sengaja dipisahkan dari direktori home user biasa di /home untuk handle jika partisi /home gagal di mounting maka system secara keseluruhan dapat diselamatkan oleh root. /root dibaca slash root.
/sbin	Direktori ini pada prinsipnya sama dengan direktori /bin yang berisi file-file binary. Namun direktori /sbin ini hanya dapat diakses oleh user root dalam kondisi sistem normal. Users biasa tidak memiliki akses untuk menjalankan binary files di direktori ini
/tmp	Direktori yang berisi temporary files artinya file-file yang sifatnya hanya dibutuhkan sementara dan akan hilang setiap komputer booting. Direktori ini dapat diakses oleh semua users sistem.
/usr	Bisa dikatakan sebagai direktori utama dibawah main direktori

	tempat diletakkannya file- file program aplikasi, dokumentasi, source kernel dan x-window system dari distribusi GNU/Linux yang digunakan. Mirip seperti Program Files pada Windows.
/var	Direktori yang diperuntukkan menyimpan file-file yang akan terus membesar ukurannya. Biasa digunakan untuk menyimpan file-file log system, file email, file-file web, file database.

Sampai tahap ini, minimal kita sudah sekilas mengetahui fungsi-fungsi dari masing-masing folder yang ada di Linux

3.4.3. Penulisan Struktur Direktori

Penulisan struktur direktori ini sangat penting dipahami untuk kepentingan penggunaan perintah-perintah command Line Linux nantinya. Karena salah satu huruf saja saat mengetikkan penulisan struktur direktori, maka dianggap oleh sistem Linux salah. Ketika dianggap salah, maka perintah-perintah yang ingin kita jalankan tidak akan bekerja.

Pada penulisan struktur direktori, terdapat 2 cara penulisan. Yaitu penulisan alamat folder secara lengkap, serta penulisan alamat folder berdasarkan posisi kita berada. Kita coba bahas satu persatu.

3.4.3.1. Penulisan Alamat Lengkap

Coba perhatikan gambar dibawah :

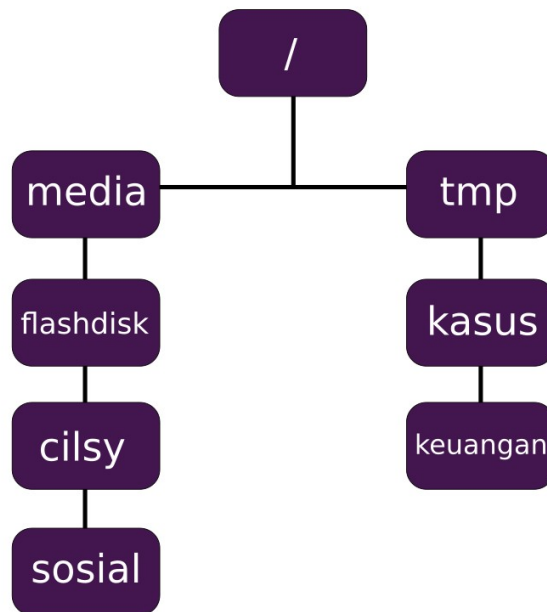


Alamat direktori 1

Pada gambar tersebut terlihat ada folder bernama musik yang berada di dalam folder home dan folder home ada di dalam folder utama /. Nah jika kita ditanya bagaimanakah penulisan dari folder musik? Penulisan hierarki folder ditulis lengkap dari folder utama (/) hingga folder yang dituju dengan dipisahkan tanda / di masing-masing foldernya. Sehingga penulisannya menjadi seperti ini :

`/home/musik`

Contoh lain :



Alamat direktori 2

Pada gambar tersebut ada dua hierarki yang terpisah. Maka jika ditanya bagaimana penulisan dari folder sosial dan keuangan? Maka penulisannya menjadi seperti ini :

Hierarki 1 : `/media/flashdisk/cilsy/sosial`

Hierarki 2 : `/tmp/kasus/keuangan`

Lalu jika ditanya lagi, bagaimanakah penulisan dari folder media dan kasus? Maka menjadi seperti ini :

Hierarki 1 : `/media`

Hierarki 2 : `/tmp/kasus`

Sebenarnya kita cukup membayangkan hierarki folder dan penulisan ini seperti layaknya kita melakukan klik folder, lalu di dalam folder tersebut ada folder lain lagi, di klik lagi ada folder lagi, dst.

Keuntungan dari teknik alamat lengkap ini adalah minim kesalahan. Selama kita menulis alamat lengkapnya secara benar, maka kita tidak perlu khawatir salah melakukan perintah command line tertentu nantinya.

Kekurangan dari teknik ini adalah kita akan cukup panjang saat mengetik alamat foldernya. Apalagi jika letak foldernya sangat dalam dan jauh. Misalnya saja :

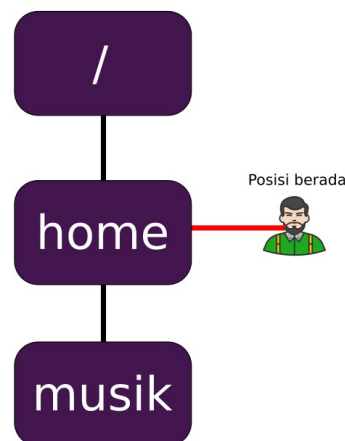
```
/var/www/html/bhinneka/admin/panel/configuration/beta1/access
```

Akan sangat melelahkan bukan jika harus terus mengetik sepanjang itu?

3.4.3.2. Penulisan Alamat berdasarkan posisi kita berada

Untuk mengatasi penulisan alamat yang terlalu panjang, kita bisa menggunakan teknik yang satu ini. Teknik ini memperdulikan posisi kita berada saat melakukan penulisan alamat. Sehingga file atau folder yang ada didalam posisi kita saat ini berada tidak perlu di tulis dengan tanda / dan folder-folder diatas posisi kita berada tidak perlu di tulis kembali.

Misalnya masih pada kasus yang sama dengan folder musik, namun disini dianggap kita berada di dalam folder home.



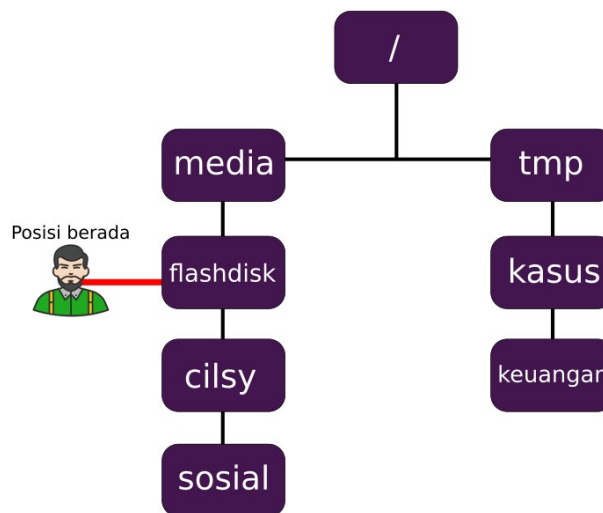
Alamat direktori 3

Maka ketika kita diminta seperti apakah penulisan folder musik, jawabannya seperti ini :

```
musik
```


Jadi sama sekali tidak ada penulisan lengkap mulai dari /home/musik. Melainkan cukup musik saja.

Kita ambil lagi contoh yang ini :



Alamat direktori 4

Pada gambar diatas kita diposisikan berada di dalam folder flashdisk. Sehingga jika ditanya penulisan dari folder sosial, adalah seperti ini :

`cilsy/sosial`

Didepan folder cilsy, sama sekali tidak ada tanda /. Karena folder cilsy berada didalam posisi kita berada sekarang.

Lalu bagaimana dengan folder keuangan? Bisakah kita menuliskannya cukup dengan begini ? :

`kasus/keuangan`

atau :

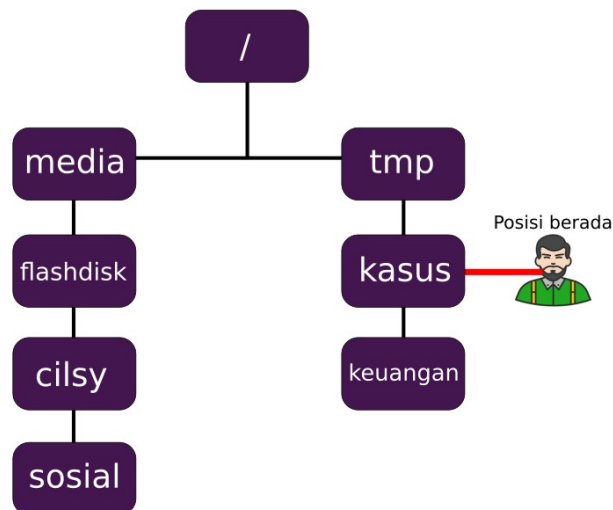
`keuangan`

Jawabannya adalah tidak bisa. Karena kita tidak berada didalam folder tmp maupun folder kasus. Sehingga jika posisi kita berada masih di folder flashdisk

tapi diminta untuk menuliskan letak folder keuangan, kita tetap harus menuliskannya lengkap seperti ini :

`/tmp/kasus/keuangan`

Namun ketika kita sudah pindah ke folder kasus seperti gambar berikut :



Alamat direktori 5

Barulah kita bisa menuliskan keuangan seperti ini :

`keuangan`

Kalau kita ada di folder tmp, bagaimana kah penulisan alamat folder keuangan?

`kasus/keuangan`

Kalau kita ada di folder /, bagaimana kah penulisan alamat folder keuangan?

`tmp/kasus/keuangan`

Kesimpulannya teknik ini dapat lebih mempersingkat penulisan, namun memang sangat rawan kesalahan. Teknik ini biasa digunakan ketika banyak command yang harus dilakukan didalam folder yang sangat dalam posisinya. Kalau belum terlalu paham betul dengan konsep ini, sebaiknya selalu gunakan teknik yang penulisan alamat lengkap.

3.4.4. Exercise

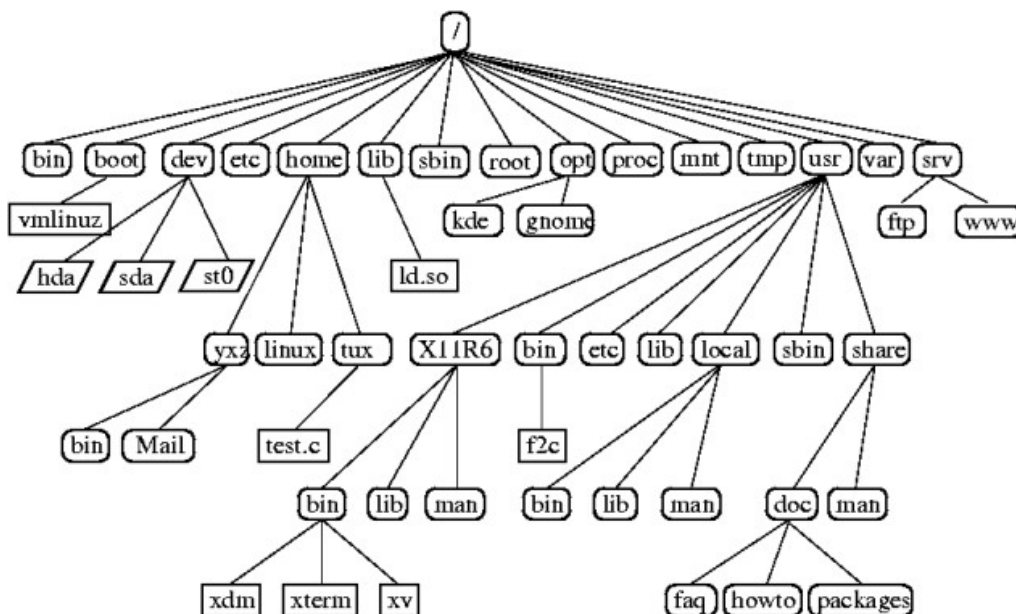
Teori

1. Kenapa kita perlu belajar Struktur direktori Linux?

Praktek

Waktu pengerjaan : 7 menit.

Masing-masing peserta selesaikan soal dibawah ini dengan cara mengkopi jawabannya ke kolom chat saat pembelajaran.



Struktur direktori di Linux

1. Tuliskan alamat dari folder **Mail** yang berada didalam folder **yxz** (abaikan posisi).
2. Tuliskan alamat folder **howto** yang berada didalam **doc** jika posisi kita sedang berada di dalam folder **share**.

3. Tuliskan alamat folder **packages** yang berada didalam **doc** jika posisi kita sedang berada di dalam folder **X11R6**.

3.5. Perintah-perintah dasar Command Line Linux

3.5.1. Apa itu Command Line?

Sebuah server dituntut agar beban kerjanya dapat se-ringan dan se-aman mungkin. Oleh karena itu biasanya hampir semua Server pasti tidak memiliki tampilan grafis untuk meringankan beban RAM dan CPU. Semua hal yang ingin dilakukan di server akan digantikan menggunakan perintah-perintah yang diketik, atau biasa disebut sebagai command line.

Aktifitas sesimpel membuat folder yang tinggal klik saja di komputer Windows, di Linux (dalam konteks server) harus dilakukan dengan mengetikkan sebuah command.

Seorang DevOps yang akan selalu mengelola server Linux wajib terbiasa dengan command line. Untuk konfigurasi dan setup server yang akan dikelola, maupun untuk membuat berbagai macam scripting untuk otomasi nantinya.

3.5.2. Remote Linux

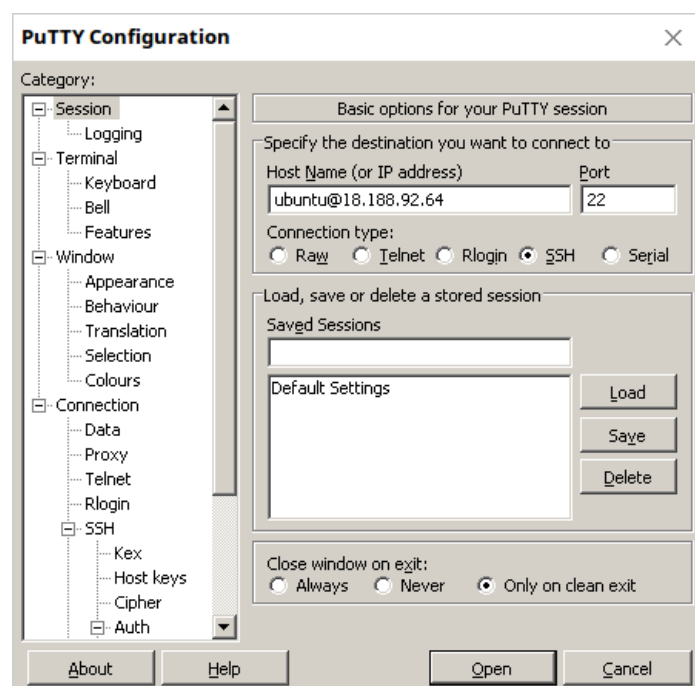
Pada bagian ini kita akan banyak menggunakan GNU/Linux, maka baiknya kalian menggunakan sistem operasi linux yang sudah terinstall di komputer. Apabila kalian belum menginstallkannya, maka bisa menggunakan ubuntu yang sudah disediakan oleh team Sekolah DevOps Cilsy. Untuk kawan-kawan yang ingin menginstallkan linuxnya langsung dikomputer, maka team cilsy akan menyiapkan modulnya secara terpisah.

Cara menggunakan linux yang sudah disediakan adalah dengan me-remote linux tersebut agar dapat langsung kita guakan, untuk melakukan remote

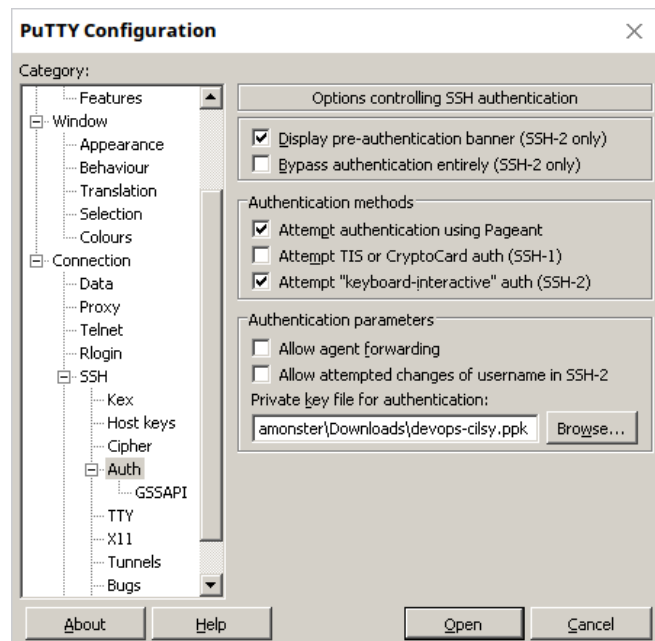
server pada linux tersebut kalian harus download terlebih dahulu aplikasi putty di link berikut.

<https://the.earth.li/~sgtatham/putty/latest/w32/putty-0.70-installer.msi>

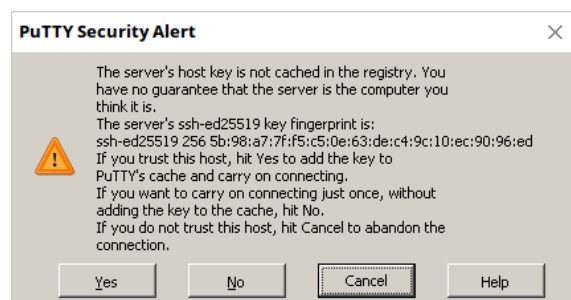
Install aplikasi tersebut sampai selesai, setelah itu buka aplikasi Putty untuk melakukan akses ke ubuntu tersebut, setelah itu isikan username dan ip public dari ubuntu yang akan kita remote. Username dan IP Public sudah diberikan sebelumnya melalui chat slack



Sebelum klik open, pilih terlebih dahulu menu Auth yang ada di kategori menu connection > SSH > Auth. Klik tombol browse dan cari file keypair.ppk yang sudah diberikan sebelumnya.



Klik open lalu kita akan diarahkan ke pop out security alert, pilih yes untuk melanjutkan.



Dengan begitu kita sudah selesai remote login ke ubuntu yang akan kita gunakan, dan hasilnya adalah sebagai berikut ini.

```
ubuntu@ip-172-31-0-250: ~
Using username "ubuntu".
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-1065-aws x86_64)

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

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

New release '18.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Sep  7 04:39:12 2018 from 180.246.29.161
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-0-250:~$
```

3.5.3. Belajar Login Logout

Saat kita berhasil masuk ke sistem Linux, maka pertama kali yang kita lihat adalah layar hitam dengan tulisan :

```
cilsy login:
```

```
Password:
```

Itu adalah tampilan prompt login default dari Linux. Kita tidak akan bisa melakukan apapun di Linux sebelum kita berhasil login. Sehingga yang pertama harus dilakukan adalah login sesuai user masing-masing yang sudah didapatkan saat instalasi maupun saat kita membuat server di AWS. Misalnya disini user yang digunakan adalah rizal :

```
cilsy login: rizal
```

```
Password:
```

```
rizal@cilsy:~$
```

Perhatikan tanda \$, tanda tersebut mewakili jenis user yang digunakan login, yaitu user biasa. Bukan user root (administrator).

Sekarang coba ketikkan perintah berikut :

```
rizal@cilsy:~$ sudo -i
```

```
root@cilsy :~#
```

Sudo -i adalah perintah untuk masuk ke mode root atau mode administrator. Setelah masuk mode root, tanda \$ berubah menjadi #. Landing direktori kedua user diatas juga berbeda berdasarkan home direktorinya masing-masing.

User biasa berada di : /home/rizal

User root berada di : /root

Perintah sudo sendiri membuat kita dapat bebas melakukan apapun di sistem kita. Hampir semua konfigurasi dan command-command yang terkait administrasi server mewajibkan kita untuk masuk mode root terlebih dahulu menggunakan perintah sudo.

Lalu bagaimana jika kita ingin keluar dari sistem Linux kita? Gunakan perintah logout atau exit. Bisa juga dengan menekan kombinasi tombol ctrl+d.

```
rizal@cilsy:~$ logout
```

```
rizal@cilsy:~$ exit
```

3.5.4. Perintah Navigasi

Perintah navigasi digunakan untuk menjelajahi direktori-direktori yang ada di Linux. Sama seperti ketika kita melakukan pindah-pindah folder dengan mengklik-klik di komputer Windows.

Berikut merupakan beberapa perintah navigasi pada GNU/Linux.

pwd : print working directory, untuk melihat alamat direktori kita berada saat ini

cd : change directory, untuk berpindah folder

ls : list, melihat daftar isi direktori saat ini

Contoh dari penggunaan perintah navigasi bisa dilakukan dengan cara berikut.

pwd : untuk melihat posisi kita berada sekarang

- **pwd**

cd : untuk berpindah direktori

- **cd /home/devops** (jika /home/devops berada diluar direktori sekarang)
- **cd home/devops** (jika home berada didalam direktori sekarang)
- **cd ..** (kembali satu tingkat ke direktori sebelumnya)
- **cd -** (kembali ke direktori yang dimasuki sebelumnya)

- **cd** (kembali ke home direktori)

ls : untuk melihat isi folder

- **ls** (jika ingin melihat isi direktori tempat berada sekarang)
- **ls /home/devops** (Melihat isi dari direktori devops)

3.5.5. Perintah pengelolaan Direktori & File

Perintah ini digunakan untuk aktifitas dasar dalam pengelolaan file dan folder di Linux. Yaitu untuk membuat file, membuat folder, mengkopi, cut-paste, menghapus, hingga merubah nama.

Berikut merupakan beberapa perintah pengelolaan direktori pada GNU/Linux.

cp : copy, untuk menyalin berkas atau folder (seperti copy paste)

mv : move, untuk memindahkan atau mengganti nama (seperti cut-paste)

rm : remove, untuk menghapus file atau folder

mkdir : make directory, untuk membuat folder

touch : untuk membuat file

Contoh dari penggunaan perintah Direktori bisa dilakukan dengan cara berikut.

mkdir : make directory, membuat folder

- **mkdir folderbaru** (folder yang dibuat ingin didalam direktori sekarang)
- **mkdir /tmp/folderbaru** (folder yang dibuat ingin diluar direktori sekarang)

touch : membuat file

- **touch filebaru** (file yang dibuat ingin didalam direktori sekarang)
- **touch /tmp/filebaru** (file yang dibuat ingin diluar direktori sekarang)

cp : copy, menyalin berkas atau folder

- **cp test1.txt test2.txt** (menyalin file dalam direktori ke dalam direktori)
- **cp test1.txt /tmp/test2.txt** (menyalin file dalam direktori ke luar direktori)
- **cp /tmp/test2.txt test3.txt** (menyalin file luar direktori ke dalam direktori)
- **cp /tmp/test2.txt /tmp/test3.txt** (menyalin file luar ke luar direktori)
- **cp -R /tmp/namafolder /home/namafolder** (untuk menyalin direktori yang ada isinya)

mv : move, memindahkan atau mengganti nama (seperti cut-paste)

- **mv test1.txt test2.txt** (mengganti nama test1 menjadi test2)

- **mv /tmp/test2.txt /home/rizal/test3.txt** (memindahkan file test2 lalu merubah namanya menjadi test3)

rm : remove, menghapus berkas atau folder

- **rm namabbenar.txt** (menghapus file namabbenar.txt)
- **rm -R namafolder** (untuk menghapus direktori)

3.5.6. Perintah Editing File

Perintah ini sama seperti kegiatan kita dalam mengedit dan membuat file teks di Windows menggunakan notepad. Perintah ini akan sangat sering kita gunakan, karena hampir seluruh file konfigurasi di Linux nantinya akan kita edit-edit, kita baca, kita modifikasi dengan perintah ini.

Berikut merupakan beberapa perintah file editing pada GNU/Linux.

nano : GNU nano, text editor berbasis konsol yang mudah dioperasikan

vim : vim, text editor berbasis konsol yang mudah dioperasikan

Contoh penggunaannya :

nano/vim = text editor berbasis konsol yang mudah dioperasikan

- **vim /home/rizal/test1.txt** (mengedit file diluar direktori)
- **nano rizal/test1.txt** (mengedit file di direktori kita berada sekarang)
- **nano test1.txt** (mengedit file didalam direktori kita berada)

Kita perlu mengingat beberapa kombinasi dasar saat pengoperasian text editor ini. Yaitu pada nano ada kombinasi Ctrl+X untuk menutup program (exit) dan Ctrl+O untuk menyimpan (save). Sedangkan untuk vim, sebelum menulis kita perlu mengetik karakter i (insert) lalu untuk menyimpan kita gunakan kombinasi Esc+:wq (write and quit). Adapun sudo diperlukan setiap pengguna menyunting berkas pada area sistem seperti pada /etc.

3.5.7. Perintah Bantuan Manual Guide

Pada dasarnya, di Linux kita sama sekali tidak perlu menghafal seluruh perintah yang ada. Ketika kita membutuhkan informasi detail terhadap suatu command, opsi-opsi yang digunakan apa saja, kita dapat memanfaatkan manual guide yang sudah disediakan pada setiap perintah.

Berikut merupakan beberapa perintah untuk membaca bantuan manual guide pada GNU/Linux :

man : manual, perintah untuk membuka manual guide lengkap dari suatu command
--help : opsi untuk melihat manual singkat dari suatu command

Contoh penggunaan :

- **cp --help** (melihat manual guide singkat dari perintah cp)
- **man cp** (melihat manual guide lengkap dari perintah cp)
- **nano --help** (melihat manual guide singkat dari perintah nano)
- **man nano** (melihat manual guide lengkap dari perintah nano)

Informasi yang disediakan oleh manual guide tersebut biasanya sudah sangat mewakili dan sangat bisa dipraktekkan. Namun jika masih mengalami kesulitan, jangan lupa bahwa kita selalu bisa mencarinya di google. Cukup mengetikkan “cp command practice” atau “nano command linux practice” biasanya akan muncul banyak artikel yang membahasnya.

3.5.8. Exercise

Teori

1. Kenapa kita perlu mempelajari perintah dasar command line?

Praktek

Waktu pengerjaan : 15 menit

Masing-masing peserta praktekkan soal berikut ini di kolom terminal Linux masing-masing sambil melakukan sharing screen agar instruktur dapat melihat hasilnya.

1. Pertama-tama login sebagai user masing-masing.
2. Pindahlah ke direktori /tmp.
3. Buatlah 1 buah file bernama test1.txt yang berisi “Sekolah DevOps Cilsy” didalam folder anda berada sekarang.

4. Buatlah 1 folder bernama latihan didalam folder Anda berada sekarang.
5. Kopikan file test1.txt tersebut ke dalam folder latihan yang barusan anda buat dengan nama test2.txt.
6. Masih dalam posisi Anda sekarang berada (jangan pindah folder), lakukan cut-paste file test2.txt yang ada didalam folder latihan ke folder /home/ dengan nama baru test3.txt.
7. Sekarang pindahlah ke home direktori anda lagi.
8. Hapuslah file test1.txt yang ada di /tmp tanpa anda pindah folder ke /tmp terlebih dahulu.
9. Hapuslah folder latihan yang ada di /tmp tanpa anda pindah folder ke /tmp terlebih dahulu.
10. Kumpulkan hasilnya ke instruktur.

3.6. Belajar manipulasi Teks & File

3.6.1. Apa itu manipulasi teks & file?

Kunci utama skill DevOps yang lain adalah ilmu scripting. Yaitu ilmu untuk merangkai berbagai macam command-command di Linux untuk mencapai tujuan tertentu.

Ilmu scripting ini sangat mirip dengan programming, karena menggunakan alur dan pola berpikir yang sama. Namun bedanya scripting ini versinya jauh lebih kecil dibandingkan dengan programming. Pada bab berikutnya akan kita bahas lebih dalam terkait scripting ini.

Salah satu ilmu yang akan sangat menunjang skill scripting adalah ilmu manipulasi teks dan file. Dimana kita dapat memodifikasi dan memanipulasi isi dari suatu file yang acak, untuk di ekstrak informasi pentingnya saja untuk mencapai tujuan tertentu.

Misalnya kita adalah DevOps engineer di sebuah perusahaan. Kemudian bos kita ingin mengetahui berapa kali Web Server perusahaan tersebut down selama ini.

Nah untuk mencapai tujuan tersebut, kita bisa coba menggunakan salah satu informasi dari sebuah file log yang mencatat aktifitas web server perusahaan. Kira-kira isinya seperti ini :

```
[2018-06-05] Website has been accessed from IP 18.39.9.9
[2018-07-05] Website has been down
[2018-09-12] User has been blacklisted from accessing website with IP
9.29.29.100
[2018-03-30] Service restarted
[2018-12-17] Website has been down
[2018-11-18] Website has been down
```

Dari isi file tersebut idenya adalah bagaimana caranya kita menghitung berapa kali tulisan “Website has been down” muncul lalu kita tampilkan berupa angka misalnya : 5. Bagaimana caranya?

Misalnya pertama-tama kita eliminasi dulu semua baris yang tidak mengandung kata-kata Website has been down. Sehingga isi file yang baru menjadi seperti ini :

```
[2018-07-05] Website has been down
[2018-12-17] Website has been down
[2018-11-18] Website has been down
```

Kemudian berikutnya kita hilangkan seluruh bagian tanggalnya :

```
Website has been down
Website has been down
Website has been down
```

Kemudian kita hitung berapa kali line Website has been down muncul. Sehingga pada akhirnya isi file hanya menjadi seperti ini :

```
3
```

Semua step by step diatas menggunakan teknik manipulasi file dan teks yang akan kita pelajari di bab ini. Kita akan coba bahas lebih dalam terkait teknik-teknik manipulasi teks dan file yang sering digunakan.

3.6.2. Belajar redirection & Pipeline

3.6.2.1. Apa itu Redirection & Pipeline?

Redirection dan Pipeline merupakan salah satu teknik yang dapat memperkaya kombinasi saat melakukan manipulasi teks/file. Dimana **Redirection** adalah teknik menghubungkan sebuah perintah command line dengan file, sedangkan **pipeline** adalah teknik untuk menghubungkan perintah command line dengan perintah command line lainnya.

Format dasar redirection dan piping :

```
# command > file (format redirection)
# command1 | command2 (format pipeline)
```

3.6.2.2. Contoh Penggunaan Redirection

Misalnya kita ingin memasukkan seluruh isi dari file test.txt ke dalam file baru bernama testcopy.txt.

```
# cat test.txt > testcopy.txt
```

Cobalah lihat hasil dari file testcopy.txt, maka isinya akan sama persis dengan isi dari file test.txt.

Atau misalnya kita ingin melihat apa saja isi folder dari /, lalu kita masukkan hasilnya ke dalam file bernama direktory.txt.

```
# ls / > direktory.txt
```

Coba lihat isi dari file direktory.txt, maka akan menampilkan keseluruhan isi folder /.

nb : Karena redirection sifatnya benar-benar menghapus seluruh isi file dengan yang baru, maka harus digunakan dengan sangat hati-hati. Salah-salah, isi file yang sangat penting bisa terhapus karena tidak sengaja kita redirection dengan isi command lain.

3.6.2.3. Contoh Pipeline

Misalnya kita ingin melihat isi folder /home/rizal lalu mencari file bernama devops.png. Jika kita menggunakan perintah ls saja, maka hasilnya akan sangat semerawut seperti ini :

```
rizal@rizal-Inspiron-5468 ~ $ ls /home/rizal/ | grep png
aws training scheme for company white.png
aws training scheme for trainer.png
bandwidthkesedot1.png
bandwidthkesedot2.png
devops 1.png
devops 2.png
devops 3.png
hamster2.png
header-bg.png
jalankosong.png
okr1.png
pisahjalu1.png
pisahjalu2.png
profile1.png
profile2.png
proxmox 1.png
proxmox 2.png
```

Nah jika kita hubungkan perintah ls tersebut dengan perintah untuk memfilter kata kunci tertentu seperti grep, maka hasilnya menjadi seperti ini :

```
rizal@rizal-Inspiron-5468 ~ $ ls /home/rizal/ | grep devops
devops 1.png
devops 1.svg
devops 2.png
devops 2.svg
devops 3.png
devops 3.svg
```

Hasil diatas sudah lumayan betul, tetapi masih ada yang kurang tepat. Masih ada file dengan tipe svg disana. Oleh karena itu kita bisa lakukan piping yang lebih panjang seperti ini :

```
rizal@rizal-Inspiron-5468 ~ $ ls /home/rizal/ | grep devops | grep png
devops 1.png
devops 2.png
devops 3.png
```

3.6.3. Belajar grep

3.6.3.1. Apa itu Grep?

Sebelumnya kita sudah sempat coba contoh penggunaan command grep pada saat belajar redirection dan piping. Disini kita akan bahas lebih banyak lagi contohnya. Grep ini pada intinya adalah **command untuk memfilter teks/file sesuai kata kunci tertentu.** Berikut adalah beberapa contoh-contoh penggunaannya.

3.6.3.2. Menampilkan Beberapa baris setelah atau sebelum pola yang dicari

Misal berikut ada perintah untuk menampilkan informasi jaringan pada server kita :

```
enp3s0  Link encap:Ethernet  HWaddr a4:4c:c8:47:75:42
        UP BROADCAST MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
```



```
RX packets:1319 errors:0 dropped:0 overruns:0 frame:0
TX packets:1319 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:141519 (141.5 KB) TX bytes:141519 (141.5 KB)
```

Nah kita ingin hanya menampilkan 4 baris setelah kata kunci `lo` maupun 4 baris sebelum kata kunci `lo`. Kita bisa gunakan opsi `-A` untuk 4 baris setelah, dan opsi `-B` untuk 4 baris sebelum.

```
rizal@rizal-Inspiron-5468 ~ $ ifconfig | grep -A 4 lo
lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:65536  Metric:1
            RX packets:1325 errors:0 dropped:0 overruns:0 frame:0
rizal@rizal-Inspiron-5468 ~ $ ifconfig | grep -B 4 lo
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo          Link encap:Local Loopback
```

3.6.3.3. Menampilkan teks dengan pola kata kunci tertentu

Misalnya kita tidak tahu opsi apa saja yang bisa digunakan pada command `grep`. Sedangkan kita ingin menggunakan `grep` untuk memfilter kata kunci yang hanya match secara keseluruhan kata saja. Jika kita menggunakan fasilitas bantuan dari `grep` seperti ini, hasilnya terlalu banyak dan kita bingung mana yang harus digunakan :

```
rizal@rizal-Inspiron-5468 ~ $ grep --help
Usage: grep [OPTION]... PATTERN [FILE]...
Search for PATTERN in each FILE or standard input.
PATTERN is, by default, a basic regular expression (BRE).
Example: grep -i 'hello world' menu.h main.c

Regex selection and interpretation:
```

```
-E, --extended-regexp    PATTERN is an extended regular expression (ERE)
-F, --fixed-strings      PATTERN is a set of newline-separated strings
-G, --basic-regexp       PATTERN is a basic regular expression (BRE)
-P, --perl-regexp        PATTERN is a Perl regular expression
-e, --regexp=PATTERN     use PATTERN for matching
-f, --file=FILE          obtain PATTERN from FILE
-i, --ignore-case        ignore case distinctions
-w, --word-regexp        force PATTERN to match only whole words
```

Nah kita bisa filter ini dengan grep misalnya kita cari yang hanya berkaitan dengan kata kunci “words” seperti ini :

```
rizal@rizal-Inspiron-5468 ~ $ grep --help | grep words
-w, --word-regexp        force PATTERN to match only whole words
```

Dari situ terfilter outputnya sesuai yang kita inginkan saja. Sehingga kita tahu bahwa dengan menggunakan opsi -w maka grep bisa melakukan matching secara keseluruhan kata.

3.6.3.4. Membalik Hasil yang muncul untuk memfilter

Biasanya setiap file konfigurasi di Linux mengandung banyak komentar berawalan tanda pagar “#”. Contohnya seperti ini :

```
rizal@rizal-Inspiron-5468 ~ $ sudo cat /etc/hosts
# Localhost line
127.0.0.1    localhost
127.0.1.1    rizal-Inspiron-5468

# The following lines are desirable for IPv6 capable hosts
::1         ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
```

Nah bagaimana kalau kita ingin hanya menampilkan baris yang tidak ada tanda pagarnya saja? Maka kita bisa gunakan opsi `-v`. Opsi ini membuat yang ditampilkan justru yang tidak cocok dengan apa yang kita cari.

```
rizal@rizal-Inspiron-5468 ~ $ sudo grep -v "#" /etc/hosts
127.0.0.1    localhost
127.0.1.1    rizal-Inspiron-5468

::1         ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
```

3.6.4. Exercise

Teori

1. Kenapa kita perlu mempelajari manipulasi teks dan file?

Praktek

Waktu pengerjaan : 10 menit

Masing-masing peserta praktekkan soal berikut ini di kolom terminal Linux masing-masing sambil melakukan sharing screen agar instruktur dapat melihat hasilnya.

1. Pertama-tama buatlah sebuah file bernama latihan1.txt yang berisi :

```
cilsy.mp3
cilsy.mp4
devops cilsy.png
devops.mp3
sekolah devops cilsy.mp3
#cilsy yahud.mp3
cilsy keren.mp3
```

```
#keren kok cilsy.mp3
sekolah devops.cilsy.flv
```

2. Anda diminta untuk menyaring isi file tersebut dengan ketentuan sebagai berikut :

- Hanya tampilkan baris yang mengandung kata “cilsy” dan berekstensi “.mp3” saja.
- Tidak boleh ada baris yang mengandung “#” walaupun baris tersebut sudah betul mengandung kata “cilsy” dan berekstensi “.mp3”.
- Hasil saringan file dimasukkan ke file baru bernama hasil1.txt.
- Anda hanya boleh menggunakan teknik pipeline, grep, dan redirection untuk mencapai hal tersebut.

3.6.5. Belajar Regular Expression

3.6.5.1. Apa itu Regular Expression?

Pada materi ini kita akan belajar untuk lebih memperkaya lagi kombinasi yang bisa kita lakukan dalam melakukan seleksi, manipulasi, dan modifikasi file dengan Regular Expression.

Regular Expression ini adalah simbol yang merepresentasikan pola karakter. Gunanya adalah untuk lebih mempermudah kita dalam melakukan filtering output dari sebuah command, teks, maupun file.

Regular Expression terdiri dari :

- (.) → untuk menandai minimal ada 1 karakter apapun,
- (*) → untuk menandai minimal ada 0 atau lebih karakter.
- (+) → untuk menandai minimal ada 1 atau lebih karakter .
- (?) → untuk menandai minimal ada 0 atau 1 karakter.

- [character] → untuk mencocokkan karakter yang ditentukan didalam tanda [], bisa juga menggunakan tanda hubung seperti [-] untuk menandai range dari karakter : [a-z], [1-9]
- ^ → untuk mencocokkan awal dari sebuah baris pada file.
- \$ → untuk mencocokkan akhir dari sebuah baris pada file.
- \ → karakter escape untuk menghindari sebuah karakter biasa dianggap menjadi karakter meta.
- | → sebagai fungsi OR (atau).

Kita akan coba bahas konsep dari Regular Expression ini dengan menggunakan sebuah file bernama filetest.txt dengan isi seperti ini :

```
rizal@rizal-Inspiron-5468 ~ $ cat filetest.txt
rizal
Rizal
RIZAL
manrah
rahman
rahmaN
1rizal
2rizal
3rizal
4rizal
5rizal
6Rizal
```

Isi file tersebut akan kita coba ubah dan manipulasi menggunakan pola-pola regular expression.

3.6.5.2. Penggunaan (.)

Contoh jika kita mengetikkan command berikut :

```
grep -E '.r'
```

Artinya “Cari baris yang memiliki 1 karakter apapun sebelum huruf r”. Maka akan muncul seperti ini :

```
rizal@rizal-Inspiron-5468 ~ $ grep -E '.r' filetest.txt
manrah
1rizal
2rizal
3rizal
4rizal
5rizal
```

Baris manrah masuk kriteria, karena sebelum huruf r ada 1 huruf, yaitu n. Begitupun dengan baris 1rizal, 2rizal, 3rizal, dst, karena sebelum huruf r ada angka 1,2,3,dll. Sedangkan untuk baris rizal saja, itu tidak termasuk karena sebelum huruf r tidak ada 1 huruf apapun.

Contoh lain penggunaan (.) misalnya :

```
grep -E '1.i'
```

Artinya “Cari baris yang memiliki minimal 1 karakter apapun diantara angka 1 dan huruf i”. Maka hasilnya sebagai berikut :

```
rizal@rizal-Inspiron-5468 ~ $ grep -E '1.i' filetest.txt
1rizal
```

Yang memenuhi syarat hanyalah baris 1rizal. Karena memiliki 1 karakter r diantara angka 1 dan huruf i.

3.6.5.3. Penggunaan (*)

Contoh jika kita mengetikkan perintah ini :

```
grep -E 'riz*' filetest.txt
```

Maka artinya “Cari baris yang memiliki “riz” lalu boleh diikuti dengan karakter apapun dan dengan jumlah berapapun (termasuk 0)”. Maka hasilnya :

```
rizal@rizal-Inspiron-5468 ~ $ grep -E 'riz*' filetest.txt
rizal
1rizal
```

```
2rizal
3rizal
4rizal
5rizal
```

Semua baris tersebut kebetulan memiliki karakter yang sama, yaitu setelah riz terdapat karakter al. Namun sebenarnya jika terdapat karakter seperti rizaaaaaaal maupun riz saja, itu tetap akan masuk kriteria, karena tetap memenuhi syarat bahwa setelah kata riz minimal ada 0 karakter atau lebih.

3.6.5.4. *Penggunaan +*

Untuk mencari baris yang memiliki minimal 1 huruf r sebelum huruf z.

```
rizal@rizal-Inspiron-5468 ~ $ grep -E 'r+z' filetest.txt
```

Tidak ada hasil yang muncul, karena tidak ada satu baris pun yang memiliki 1 huruf r sebelum huruf z. Misalnya ada baris seperti 1rzl atau rzal atau rzaaaaal, itu baru akan muncul.

3.6.5.5. *Penggunaan ?*

Untuk mencari baris yang memiliki minimal 0 (tidak ada) maupun 1 huruf r sebelum huruf z.

```
rizal@rizal-Inspiron-5468 ~ $ grep -E 'r?z' filetest.txt
rizal
Rizal
1rizal
2rizal
3rizal
4rizal
5rizal
6Rizal
```

Walaupun tidak ada satupun baris yang memiliki huruf r sebelum huruf z, tapi semua baris diatas tetap muncul karena masih dianggap memenuhi syarat. Yaitu boleh tidak muncul karakter r sama sekali sebelum huruf z.

3.6.5.6. Penggunaan ^ dan |

Untuk mencari baris yang berawalan r ATAU berawalan 1.

```
rizal@rizal-Inspiron-5468 ~ $ grep -E '^r|^1' filetest.txt
rizal
rahman
rahmaN
1rizal
```

Baris yang tidak berawalan r atau 1 tidak akan muncul seperti baris 2rizal, 3rizal, maupun manrah.

3.6.5.7. Penggunaan \$ dan [karakter]

Untuk mencari baris yang berakhiran l maupun berakhiran n atau N.

```
rizal@rizal-Inspiron-5468 ~ $ grep -E 'l$|[nN]$' filetest.txt
rizal
Rizal
rahman
rahmaN
1rizal
2rizal
3rizal
4rizal
5rizal
6Rizal
```

Untuk mencari baris yang berawalan angka 1-9

```
rizal@rizal-Inspiron-5468 ~ $ grep -E '^[1-9]' filetest.txt
1rizal
2rizal
3rizal
4rizal
5rizal
6Rizal
```

Untuk mencari baris yang berawalan 1-9 DAN memiliki 1 huruf R kapital.


```
rizal@rizal-Inspiron-5468 ~ $ grep -E '^[1-9]' filetest.txt | grep -E 'R'
6Rizal
```

3.6.6. Exercise

Teori

1. Kenapa kita perlu mempelajari manipulasi teks dan file?

Praktek

Waktu pengerjaan : 15 menit

Masing-masing peserta praktekkan soal berikut ini di kolom terminal Linux masing-masing sambil melakukan sharing screen agar instruktur dapat melihat hasilnya.

1. Pertama-tama buatlah sebuah file bernama latihan2.txt yang berisi :

```
1cilsy@
aDV0PScilsy
Acilsy
bcilsy#
Bcilsy
99cilsy
100cilsy
08cilsy
76cilsy
zZcilsy
%cilsy
@cilsy
#
##cilsy
#devopscilsy
#sekolahcilsy
```

```
.cilsy
.cilsy@
.cilsy1
2sekolahdevopscilsy
3cilsy@
7cilsy
9cilsy
```

2. Anda diminta untuk menyaring isi file tersebut dengan ketentuan sebagai berikut :

- Tampilkan baris yang harus berawalan angka apapun 1 digit (tidak boleh lebih) kemudian diikuti oleh tulisan cilsy. Masukkan hasilnya ke file baru bernama hasil2-1.txt.
- Tampilkan baris yang harus berawalan angka apapun 1 digit atau lebih yang diikuti oleh tulisan cilsy. Masukkan hasilnya ke file baru bernama hasil2-2.txt.
- Tampilkan baris yang harus berawalan angka apapun 1 digit (tidak boleh lebih) yang diikuti oleh tulisan cilsy dan harus diakhiri oleh tanda "@". Masukkan hasilnya ke file baru bernama hasil2-3.txt.
- Anda hanya boleh menggunakan teknik regular expression, grep, dan redirection untuk mencapai hal tersebut.

3.6.7. Belajar sed

3.6.7.1. Apa itu Sed?

Tools lain yang juga sering digunakan untuk manipulasi file dan teks adalah Sed. Sed biasa digunakan untuk mengganti karakter-karakter dari input yang diberikan. Misalnya merubah semua kalimat yang berawalan saya menjadi kamu. Merubah semua yang huruf kecil menjadi huruf kapital, dll.

3.6.7.2. Contoh penggunaan Sed

Sintaks dasar :

```
# sed 's/yangingindiubah/diubahmenjadiapa/flag'
```

Flag yang umum digunakan adalah g (untuk mengubah) dan d (untuk menghapus).

Misalnya kita ingin merubah semua kata rizal menjadi cilsy pada filetest.txt :

```
rizal@rizal-Inspiron-5468 ~ $ sed 's/rizal/cilsy/g' filetest.txt
cilsy
Rizal
RIZAL
manrah
rahman
rahmaN
1cilsy
2cilsy
3cilsy
4cilsy
5cilsy
6Rizal
```

Disana terlihat masih ada kata rizal yang tidak terganti akibat huruf kapital yang digunakan. Kita bisa coba gabungkan ilmu Regular Expression disini.

```
rizal@rizal-Inspiron-5468 ~ $ sed 's/[rR][iI][zZ][aA][lL]/cilsy/g' filetest.txt
cilsy
cilsy
cilsy
manrah
rahman
rahmaN
1cilsy
2cilsy
3cilsy
```

```
4cilsy
5cilsy
6cilsy
```

Terlihat sekarang semua karakter rizal sudah berubah menjadi cilsy baik itu yang kapital maupun tidak.

Contoh lain misal kita ingin menghapus baris yang berawalan angka :

```
rizal@rizal-Inspiron-5468 ~ $ sed '/^[1-9]/d' filetest.txt
rizal
Rizal
RIZAL
manrah
rahman
rahmaN
```

Semua baris 1rizal,2rizal,3rizal sampai 6Rizal sudah terhapus karena masuk kriteria baris yang memiliki awalan angka 1-9.

3.6.8. Belajar cut

3.6.8.1. Apa itu Cut?

Cut adalah command yang dapat memotong bagian-bagian tertentu dari sebuah file/teks. Cocok juga untuk melakukan manipulasi dan filtering.

3.6.8.2. Contoh penggunaan Cut

Misalnya kita bisa memotong karakter ke 1-3 saja dari isi filetest.txt

```
rizal@rizal-Inspiron-5468 ~ $ cut -b 1-3 filetest.txt
riz
Riz
RIZ
man
rah
rah
1ri
2ri
```

```
3ri
4ri
5ri
6Ri
```

Terlihat bahwa selain 3 karakter awal, semua karakter yang lain menjadi hilang.

Contoh lain misal kita bisa memotong karakter berdasarkan batas tertentu.

Misalnya kita punya file bernama file2.txt dengan isi seperti ini :

```
rizal@rizal-Inspiron-5468 ~ $ cat file2.txt
Sekolah,Smith,34,London
Cilsy,Evans,21,Newport
Devops,Jones,32,Truro
```

Disana kita dapat ambil bagian sebagian teks berdasarkan batas tanda koma menggunakan opsi -d (delimiter)

```
rizal@rizal-Inspiron-5468 ~ $ cut -d\, -f1 file2.txt
Sekolah
Cilsy
Devops
rizal@rizal-Inspiron-5468 ~ $ cut -d\, -f2 file2.txt
Smith
Evans
Jones
rizal@rizal-Inspiron-5468 ~ $ cut -d\, -f3 file2.txt
34
21
32
```

-f disana berarti kita ingin ambil potongan yang bagian mana. Bagian 1 artinya bagian paling kiri dari koma pertama, bagian f2 artinya sebelah kanan dari koma pertama, dst. Sedangkan -d untuk menentukan batasnya, yaitu tanda koma (disana juga digunakan tanda escape \ agar tanda koma tetap dianggap karakter biasa *baca kembali bagian Regular Expression).

3.6.9. Belajar Sort & Uniq

3.6.9.1. Apa itu Sort & Uniq?

Sort dan Uniq adalah command yang biasanya selalu digunakan berpasangan. Sort adalah command untuk mengurutkan teks, sedangkan Uniq untuk mengeliminasi baris-baris yang sama. Sangat cocok digunakan untuk menghitung jumlah baris-baris yang sama.

3.6.9.2. Contoh penggunaan Sort dan Uniq

Misalnya kita punya 1 buah file bernama file3.txt yang berisi :

```
rizal@rizal-Inspiron-5468 ~ $ cat file3.txt
website down
website aktif
website aktif
website down
website aktif
website down
website aktif
website aktif
website down
```

Kemudian kita ingin mengetahui ada berapa kali baris website down muncul dan berapa kali website aktif muncul dengan perintah sort dan uniq :

```
rizal@rizal-Inspiron-5468 ~ $ sort file3.txt | uniq -c
      5 website aktif
      4 website down
```

Perintah sort pertama akan mengurutkan seluruh baris berdasarkan alphabet, kemudian perintah uniq -c menampilkan berapa kali baris yang sama muncul.

Materi sort dan uniq ini adalah materi terakhir untuk manipulasi teks dan file. Sebenarnya masih sangat banyak kombinas-kombinasi manipulasi teks yang bisa dilakukan. Juga masih ada ada tools-tools yang lain seperti tr dan awk

yang bisa kita eksplor lebih jauh. Kunci agar bisa menguasai teknik ini hanyalah sering-sering berlatih

3.6.10. Exercise

Praktek

Waktu pengerjaan : 10 menit

Masing-masing peserta praktekkan soal berikut ini di kolom terminal Linux masing-masing sambil melakukan sharing screen agar instruktur dapat melihat hasilnya.

1. Pertama-tama buatlah sebuah file bernama latihan3.txt yang berisi :

```
lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:65536  Metric:1
            RX packets:4654 errors:0 dropped:0 overruns:0 frame:0
            TX packets:4654 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:519016 (519.0 KB)  TX bytes:519016 (519.0 KB)

wlp2s0     Link encap:Ethernet  HWaddr d4:6a:6a:9a:56:c5
            inet addr:192.168.88.254  Bcast:192.168.88.255  Mask:255.255.255.0
            inet6 addr: fe80::3d7d:f7c8:d7c7:7661/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:805339 errors:0 dropped:0 overruns:0 frame:0
            TX packets:312298 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:997268528 (997.2 MB)  TX bytes:52292659 (52.2 MB)

eth0       Link encap:Ethernet  HWaddr d4:6a:6a:9a:56:c5
```

```

inet addr:11.10.100.29 Bcast: 11.10.100.255 Mask:255.255.255.0
inet6 addr: fe80::3d7d:f7c8:d7c7:7661/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:805339 errors:0 dropped:0 overruns:0 frame:0
TX packets:312298 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:997268528 (997.2 MB) TX bytes:52292659 (52.2 MB)

```

2. Anda diminta untuk menyaring isi file sehingga hasil akhirnya menjadi seperti ini :

```

11-10-100-29
127-0-0-1
192-168-88-254

```

Hasil akhir tidak perlu dimasukkan ke suatu file. Cukup tampilkan di command line saja. Anda bebas menggunakan seluruh teknik yang sudah dipelajari di materi manipulasi teks dan file ini.

3.7. Belajar archiving

3.7.1. Apa itu Archiving?

Archiving ini sebenarnya sudah sangat sering kita lakukan sehari-hari menggunakan komputer Windows/Mac OS. Yaitu ketika kita meng-compress file menjadi .zip atau .rar maupun sebaliknya, mengekstrak. Nah disini kita akan mempelajari bagaimana teknik tersebut dilakukan di Linux.

Ilmu Archiving ini juga sangat penting untuk dipelajari, karena kegiatan untuk meng-compress dan meng-ekstrak file itu akan cukup sering dilakukan. Fungsinya untuk memperkecil ukuran-ukuran file yang nantinya disimpan di Server, biasa digunakan untuk kebutuhan backup dan restore data.

3.7.2. Jenis dan Opsi kompresi

Dalam melakukan kompresi di Linux, tools yang paling terkenal digunakan bernama tar. Jenis kompresi yang dihasilkan oleh tar ada 3, yaitu Gzip dengan ekstensi gz, Bzip2 dengan ekstensi bz2, dan Xz dengan ekstensi xz. Xz memiliki

kemampuan kompresi yang paling baik (nilai file yang terkompres paling kecil), namun memakan waktu yang paling banyak saat melakukan kompresi. Sedangkan Gz yang nilai kompresinya paling biasa saja, namun waktu untuk melakukan kompresinya sangat cepat. Bzip2 berada ditengah-tengah.

Sintaks dasar tar adalah sebagai berikut :

```
# tar [ opsi ] [ fileyang ingin dikompresi ]
```

Opsi-opsi yang dapat digunakan pun cukup banyak :

Opsi panjang	Opsi pendek	Deskripsi
--create	c	Creates a tar archive
--concatenate	A	Appends tar files to an archive
--append	r	Appends files to the end of an archive
--update	u	Appends files newer than copy in archive
--diff or --compare	d	Find differences between archive and file system
--file archive	f	Use archive file or device ARCHIVE
--list	t	Lists the contents of a tarball
--extract or --get	x	Extracts files from an archive

Opsi panjang	Opsi	deskripsi
--------------	------	-----------

	pendek	
--directory dir	C	Changes to directory dir before performing operations
--same-permissions	p	Preserves original permissions
--verbose	v	Lists all files read or extracted. When this flag is used along with -list, the file sizes, ownership, and time stamps are displayed.
--verify	W	Verifies the archive after writing it
--exclude file	—	Excludes file from the archive
--exclude=pattern	X	Exclude files, given as a PATTERN
--gzip or --gunzip	z	Processes an archive through gzip
--bzip2	j	Processes an archive through bzip2
--xz	J	Processes an archive through xz

Kita akan coba praktekkan bagaimana melakukan berbagai macam teknik archiving menggunakan opsi-opsi tersebut.

3.7.3. Melakukan Kompresi

Misalnya saja kita mempunyai 9 buah file bernama file0, file1, file2 hingga file9 yang ingin kita kompres. Maka perintahnya :

```
# touch file{0..9} (perintah untuk membuat filenya terlebih dahulu)
# tar czf myfiles.tar.gz file[0-9] (untuk kompresi menggunakan gunzip)
# tar cjf myfiles.tar.bz2 file[0-9] (untuk kompresi menggunakan bzip2)
```

```
# tar cJf myfile.tar.xz file[0-9] (untuk kompresi menggunakan xz)
```

3.7.4. Melihat isi file yang sudah di kompres

Untuk melihat isi file dari sebuah file arsip adalah seperti ini :

```
# tar tvf myfiles.tar.gz
# tar tvf myfiles.tar.bzip2
# tar tvf myfiles.tar.xz
```

3.7.5. Mengupdate dan menghapus file didalam Arsip

Bagaimana jika suatu ketika kita ingin menambahkan atau menghapus file tertentu dari dalam file yang sudah terarsip? Pertama-tama jalankan perintah command berikut (pilih salah satu saja) untuk mengekstrak file arsip :

```
# gzip -d myfiles.tar.gz      [#1]
# bzip2 -d myfiles.tar.bz2    [#2]
# xz -d myfiles.tar.xz        [#3]
```

Kemudian :

```
# tar --delete --file myfiles.tar file4 (menghapus file4 dari dalam file arsip)
# tar --update --file myfiles.tar file10 (menambahkan file10 ke dalam file arsip)
```

Lalu eksekusi juga perintah ini untuk mengkompres kembali :

```
# gzip myfiles.tar           [ jika memilih #1]
# bzip2 myfiles.tar          [ Jika memilih #2 ]
# xz myfiles.tar              [ Jika memilih #3 ]
```

Terakhir coba cek ulang isi dari file tersebut dan lihat perubahannya. Seharusnya kini file4 sudah tidak ada tergantikan oleh file10.

3.7.6. Mengekstrak File

Perintah untuk mengekstrak file arsip secara keseluruhan adalah sebagai berikut :

```
# tar xjf myfiles.tar.bz2 --directory /home/
```

Perintah diatas juga menggunakan opsi kemana file tersebut akan diletakkan setelah diekstrak, yaitu ke direktori /home.

3.7.7. Exercise

Teori

1. Kenapa kita perlu mempelajari Archiving file?

Praktek

Waktu pengerjaan : 10 menit

Masing-masing peserta praktekkan soal berikut ini di kolom terminal Linux masing-masing sambil melakukan sharing screen agar instruktur dapat melihat hasilnya.

1. Pertama-tama buatlah 5 buah file bernama test1, test2, test3, test4, dan test5.
2. Buatlah 1 archive bzip2 bernama latihan4.tar.bz2 yang berisi file test1 sampai test4 saja. Test5 tidak.
3. Setelah itu hapus file test4 yang ada didalam file archive yang sudah dibuat dan gantikan dengan file test5.
4. Terakhir ekstraklah seluruh isi archive latihan4.tar.bz2 ke folder /home.

3.8. Belajar finding files

3.8.1. Apa itu Finding Files?

Finding files ini juga selalu kita gunakan sehari-hari di komputer Windows. Yaitu ketika kita ingin mencari suatu file di File Explorer. Nah di Linux kita akan mempelajari bagaimana caranya dengan berbagai teknik pencarian file yang lebih efektif, seperti mencari file berdasarkan kata kunci, berdasarkan kapan terakhir di edit, berdasarkan ukuran filenya, dll. Ilmu ini akan sangat berguna untuk kegiatan kita sehari-hari di Server Linux dan juga untuk kebutuhan

scripting. Misalnya ketika kita lupa letak file yang kita edit 2 hari yang lalu, atau kita ingin menghapus file-file yang berukuran terlalu besar, atau ingin menghapus file-file backup yang sudah kadaluarsa.

Untuk pencarian file kita dapat menggunakan command bernama find. Sintaks dasar dari find adalah sebagai berikut :

```
# find [dimanafolderkitamencari] [opsi]
```

3.8.2. Mencari file berdasarkan nama dan ekstensi

Misalnya kita ingin mencari file-file dengan nama apapun yang berekstensi .conf di folder /etc, maka perintahnya adalah sebagai berikut :

```
# find /etc -iname "*.conf" -print
```

3.8.3. Mencari file berdasarkan tipe dan ukuran

Kita juga bisa mencari file berdasarkan tipe filenya maupun ukurannya. Pada perintah dibawah kita menggunakan opsi -maxdepth 3 yang berarti paling jauh hanya mencari 3 subfolder kebawah saja, opsi -type f yang berarti kita hanya mencari tipe file biasa saja (ada juga tipe direktori, symbolic link, dll), serta hanya yang berukuran lebih dari 2MB saja.

```
# find . -maxdepth 3 -type f -size +2M -print
```

3.8.4. Mencari file berdasarkan waktu dimodifikasi

Berikut contoh untuk mencari file yang pernah diedit/diubah dalam kurun waktu kurang dari 180 hari yang lalu.

```
# find /etc -iname "*.conf" -mtime -180 -print
```

Bisa juga menggunakan opsi -atime (pernah di akses).

3.8.5. Mencari file berdasarkan permission

Kita juga bisa cari file-file berdasarkan nilai permissionnya (kita akan bahas permission lebih dalam berikutnya). Misalnya disini kita mencari file yang memiliki permission berbahaya, yaitu 777 dimana file yang memiliki permission 777 akan dapat di edit/dihapus oleh siapapun.

```
# find /home/user -perm 777
```

3.8.6. Exercise

Teori

1. Kenapa kita perlu mempelajari Finding file?

Praktek

Waktu pengerjaan : 5 menit

Masing-masing peserta praktekkan soal berikut ini di kolom terminal Linux masing-masing sambil melakukan sharing screen agar instruktur dapat melihat hasilnya.

1. Carilah didalam folder /etc, file-file dengan ekstensi .conf yang memiliki permission 600 yang terakhir diakses 100 hari yang lalu. Simpan hasilnya dengan teknik redirection ke file hasil5.txt

3.9. Belajar Manajemen User & Group

3.9.1. Apa itu manajemen user & group?

Kita sudah terbiasa menggunakan konsep 1 user dalam keseharian kita di laptop Windows. Dimana kita cukup login 1 buah user, dan user itulah yang kita pakai. Biasanya user yang kita pakai adalah user yang juga berfungsi sebagai user Administrator. Semuanya kita lakukan dengan menggunakan 1 user ini, apapun kegiatannya.

Jika dianalogikan, yang kita lakukan tersebut seperti ada sebuah rumah yang ditinggali hanya oleh 1 orang. 1 orang ini adalah satu-satunya yang kita perbolehkan untuk masuk ke seluruh ruangan di rumah tersebut termasuk menggunakan segala fasilitasnya seperti mencuci, memasak, tidur dll. Tanpa ada batasan apapun.

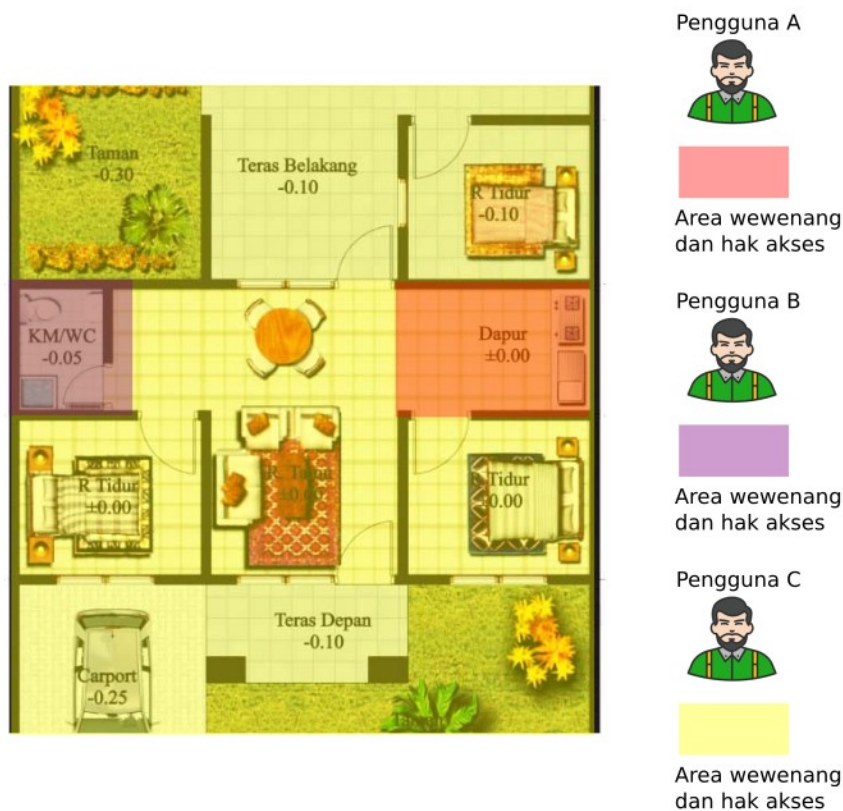


Ilustrasi Single User

Ketika kita berbicara sebuah Server Linux, maka kita sudah berbicara konsep Multi User. Dimana dalam sebuah server itu terdapat banyak user yang memiliki tugas dan wewenangnya masing-masing. Ada user yang khusus untuk menjalankan layanan web saja, ada user yang khusus untuk menjalankan layanan database, ada kelompok user yang berhak mengakses file-file departemen keuangan saja, hingga ada user yang khusus berfungsi sebagai Administrator. Kenapa perlu seperti ini? Karena inilah salah satu letak keamanan Linux. Dimana kita bisa mengatur hak akses segala macam layanan, folder, file, hingga command-command yang bisa dieksekusi berdasarkan user atau grup tertentu saja.

Ibarat analogi rumah tadi, jika konsep Multi User maka di rumah tersebut misalnya terdapat 4 orang bernama A,B,C,D. Si A hanya boleh masuk ke dapur untuk melakukan segala kegiatan di dapur tersebut, ke ruangan lain sama sekali tidak boleh masuk. Si B misalnya boleh masuk ke dapur, tapi hanya

boleh memasak saja. Memblender, mencuci piring, dll tidak boleh. Lalu si C hanya boleh mengakses kamar mandi saja, tidak boleh ke ruangan lain. Sedangkan si D ini boleh mengakses seluruh ruangan dan menggunakan segala fasilitasnya. Semua sudah diatur sesuai wewenang dan hak aksesnya masing-masing agar lebih aman.



Ilustrasi Multi User

Oleh karena itu disini kita akan mempelajari terlebih dahulu bagaimana cara melakukan manajemen user dan grup seperti membuat user, menghapus user, memasukkan user ke dalam sebuah grup, mengeluarkannya, dst. Untuk kemudian user dan grup yang sudah kita atur ini bisa kita konfigurasi lagi hak akses dan wewenangnya.

Dalam konsep Multiuser, manajemen user dan group sangat vital peranannya. Hal ini harus dikuasai dengan baik untuk menerapkan kebijakan-kebijakan

seputar ownership permission di lingkungan GNU/Linux. Terkait ownership dan permission sendiri akan kita bahas lebih dalam di materi berikutnya.

3.9.2. Perintah-perintah manajemen User

3.9.2.1. Membuat User Baru

Perintah untuk menambahkan user baru bernama limbad.

```
# adduser limbad
```

3.9.2.2. Memberi/mengganti password user

Perintah untuk memberi password baru maupun mengganti password dari sebuah user.

```
# passwd namauser
```

3.9.2.3. Mengganti nama user

Perintah untuk mengganti nama user ke nama yang baru.

```
# usermod -l namabarunya namauserlama
```

3.9.2.4. Menghapus User

Berikut adalah perintah untuk menghapus user beserta home direktorinya (seluruh data-datanya). Hilangkan opsi -r jika hanya ingin menghapus usernya saja tanpa menghapus data.

```
# userdel -r limbad
```

3.9.2.5. Disable/lock user

Kita juga bisa hanya mengunci user dari pemakaian dibanding harus menghapusnya. Perintahnya adalah sebagai berikut untuk mendisable user root dan user cilsy :

```
# passwd -l root
```

```
# passwd -l cilsy
```

3.9.2.6. *Unlock user*

Jika suatu saat sudah ingin digunakan kembali, maka kita tinggal meng-enable kembali user yang sebelumnya kita kunci.

```
# passwd -u cilsy
```

3.9.2.7. *Perintah-perintah manajemen user lainnya*

```
su → berganti user  
id → mengecek user ID  
w → mengetahui siapa user yang sedang login  
whoami → siapa saya?  
last → melihat daftar login terakhir
```

3.9.3. Perintah-perintah manajemen Group

3.9.3.1. *Membuat Group Baru*

Membuat group baru bernama staff

```
# groupadd staff
```

3.9.3.2. *Merubah nama Group*

Perintah untuk merubah nama grup dari staff menjadi office.

```
# groupmod -n office staff
```

3.9.3.3. *Menghapus Group*

Menghapus group office yang sudah kita buat sebelumnya :

```
# groupdel office
```

3.9.3.4. *Menampilkan group suatu user*

Kita dapat mencari tahu suatu user itu tergabung kedalam grup apa saja dengan perintah berikut :

```
# groups rizal
```

3.9.3.5. Menambahkan user ke suatu grup

User limbad yang telah dibuat sebelumnya akan dimasukkan ke dalam grup root. Perintahnya :

```
# usermod -a -G root limbad
```

3.9.3.6. Mengeluarkan user dari suatu grup

User limbad yang telah dimasukkan ke grup root akan dikeluarkan lagi. Perintahnya adalah sebagai berikut :

```
# gpasswd -d limbad root
```

3.9.4. Exercise

Praktek

Waktu pengerjaan : 5 menit

Masing-masing peserta praktekkan soal berikut ini di kolom terminal Linux masing-masing sambil melakukan sharing screen agar instruktur dapat melihat hasilnya.

1. Buatlah user dan grup dengan skema sebagai berikut dan tunjukkan hasilnya :

Grup	Anggota
keuangan	1. cilsy 2. devops 3. jono
penjualan	1. tono 2. devops 3. jono
hrd	1. siska 2. ramon 3. jono

Teori

1. Kenapa kita perlu mempelajari Manajemen User & Grup?
2. Pada kasus praktek diatas, apa efek ketika 1 user menjadi anggota 2 grup atau lebih (jono)?

3.10. Belajar Permission

3.10.1. Apa itu Permission?

Permission ini sangat erat kaitannya dengan materi manajemen user dan grup sebelumnya. Dimana materi manajemen user & grup adalah materi untuk membuat dan mendesain skema user dan grup yang diinginkan, sedangkan pada materi Permission kita akan mengatur hak akses apa saja yang kita berikan kepada user-user dan grup tersebut.

3.10.2. Penjelasan Konsep Permission

3.10.2.1. Aturan Umum Permission

Permission memiliki aturan-aturan umum sebagai konsep dasar

1. Setiap file pasti dimiliki oleh satu user pemilik dan satu group pemilik

Didalam sistem GNU/Linux setiap file, direktori ataupun proses pasti dimiliki oleh salah satu user dan group. Pada gambar dibawah ditunjukkan bahwa folder rizal dimiliki oleh user rizal dan dimiliki oleh grup rizal. Sedangkan folder lost+found dimiliki oleh user root dan grup root. Angka 1 menunjukkan user pemilik, dan angka 2 menunjukkan grup pemilik.

```
rizal@rizal-Inspiron-5458 /home $ ls -l
total 28
drwx----- 2 1 root 2 root 16384 Jan  4  2018 lost+found
drwxr-xr-x 57 1 rizal 2 rizal 12288 Aug 12 13:12 rizal
```

Selain yang tertera sebagai user pemilik dan grup pemilik disebut other.

2. Setiap file hak aksesnya dimiliki oleh user pemilik, group pemilik dan other. Dan setiap user pemilik, group pemilik, dan other memiliki hak akses nya sendiri-sendiri.

```
rizal@rizal-Inspiron-5468 /home $ ls -l
total 28
drwx----- 2 root root 16384 Jan 4 2018 lost+found
drwxr-xr-x 57 rizal rizal 12288 Aug 12 13:12 rizal
```

Pada gambar diatas tertulis kombinasi huruf-huruf seperti `rwxr-xr-x` dan `rwx-----`. Itu disebut sebagai permission atau hak akses. Dimana `r` berarti hak akses untuk read (membaca, membuka), `w` berarti hak akses untuk write (menghapus, mengedit, membuat), `x` berarti hak akses untuk execute (menjalankan script, membuka folder), dan `-` berarti hak akses none (tidak punya akses apa-apa).

Perhatikan pola dibawah. Secara keseluruhan permission memiliki 9 kombinasi `r,w,x,-` yang dibagi menjadi 3 blok (masing-masing 3 kombinasi per bloknya). Yaitu blok permission untuk masing-masing user, group, dan others yang sudah ditandai dengan warna merah, hijau dan biru. Masing-masing user, group, dan others memiliki kombinasi permissionnya masing-masing.

`rwxrwxrwx`

			Other
			Group
			User

Jika kita sambungkan lagi dengan screenshot gambar diatas, maka artinya folder `rizal` memiliki permission `rwxr-xr-x`. Dimana user `rizal` sebagai user pemilik memiliki hak akses untuk `rwx` (full akses) , grup `rizal` memiliki hak akses untuk `r-x` (tidak bisa write, hanya baca dan eksekusi saja), dan others memiliki hak akses untuk `r-x` (tidak bisa write, hanya baca dan eksekusi saja).

	Owner	Group	World
Read	✓	✓	✓
Write	✓	✗	✗
Execute	✓	✓	✓

Ilustrasi Permission

3. Group mewakili user-user anggotanya.

Misalnya ada sebuah grup bernama keuangan. Anggota grup tersebut adalah cilsy, egi, adi. Jika ada sebuah file yang dimiliki oleh grup keuangan, maka user cilsy, irfan, dan adi akan mewakili hak akses grup tersebut. Misalnya grup keuangan memiliki hak akses rwx pada folder invoice, maka user cilsy, irfan dan adi bisa melakukan apapun didalam folder tersebut seperti membuat file, mengedit file, membuat folder baru, dll.

4. Other adalah user/grup selain user pemilik maupun group pemilik

Misalnya ada sebuah file yang dimiliki oleh user egi dan grup marketing. Grup marketing beranggotakan tresna dan saputra. Maka jika ada user/grup lain bernama gudang atau jono itu dianggap sebagai other, karena gudang jono bukanlah user egi maupun anggota grup marketing.

5. User pemilik file memiliki hak mengatur permission file bersangkutan

Pada kasus sebelumnya terlihat bahwa folder rizal dimiliki oleh user rizal dan grup rizal serta memiliki permission rwxr-xr-x. Nah yang berhak untuk merubah permission rwxr-xr-x menjadi misalnya rwx-----, itu hanyalah user rizal sebagai user pemilik. Jika ada user lain misalnya cilsy, maka itu tidak bisa mengubahnya. Selain user pemilik ada 1 user yang juga bisa mengubah permission, yaitu user root (administrator).

3.10.2.2. Cara Menghitung Nilai Oktal Permission

Perhatikan kembali pola berikut :

rw**x****rw****x****rw****x**

| | | _____ Other
 | | _____ Group
 | _____ User

Pada materi ini, kita akan belajar untuk bisa mengubah seluruh 9 kombinasi rwx pada permission diatas menjadi bentuk angka. Kenapa? Karena saat kita melakukan pengaturan permission, kita akan memasukkan sintaks commandnya dalam bentuk angka. Misalnya 777 untuk mewakili kombinasi rwxrwxrwx, 755 untuk mewakili kombinasi rwxr-xr-x, 600 untuk mewakili kombinasi rw-----, dst.

Lalu bagaimana cara menghitungnya? Pertama-tama kita perlu tahu dulu berapa nilai satuan dari masing-masing rwx-. Yaitu sebagai berikut :

r = 4 w = 2 x = 1 - = 0

Setelah mengetahui nilai satuannya, kita coba gabungkan kombinasi tersebut kedalam 1 blok terlebih dahulu. Caranya adalah dengan menjumlahkan nilainya. Misalnya begini :

rwX = (r) 4 + (w) 2 + (x) 1 = 7

r-x = (r) 4 + (-) 0 + (x) 1 = 5

r-- = (r) 4 + (-) 0 + (-) 0 = 4

Mudah bukan? Nah berikut adalah contoh kombinasi-kombinasi lainnya.

rwX = 7 r-- = 4 --x = 1

rw- = 6 -wx = 3 --- = 0

r-x = 5 -w- = 2

Setelah sudah memahami bagaimana menghitung nilainya dalam 1 blok, maka berikutnya kita tinggal menghitungnya secara keseluruhan blok. Tekniknya kita

tinggal menghitungnya secara per blok, baru dijabarkan secara keseluruhan. Contohnya begini :

```
rw-rw-rw- = [4+2+1] [4+2+1] [4+2+1] = 777
```

```
rw-r--r-x = [4+2+1] [4+0+0] [4+0+1] = 745
```

Ingat kuncinya adalah dihitung dulu perblok, lalu dijabarkan (tidak dijumlah lagi).

Berikut adalah contoh-contoh dari kombinasi permission secara keseluruhan :

```
rw-rw-rw- = 777
```

```
rw-r-xr-x = 755
```

```
rw----- = 700
```

```
-wx--xrw- = 316
```

3.10.3. Praktek Permission

Memperbaiki Ownership pada suatu file saja tidak cukup. Terkadang kepemilikan file sudah diberikan pada user dan grup yang tepat, tetapi nyatanya user tersebut misalnya tidak bisa mengedit atau membuat file baru. Nah disinilah peran Permission. Yaitu untuk mengatur hak akses yang tepat kepada folder/file sesuai dengan Ownership yang sudah diberikan.

Format untuk mengubah permission adalah sebagai berikut :

```
# chmod [opsi] nilai_oktal namafilenya
```

Misalnya kita mempunyai folder dengan nama latihan dengan permission sebagai berikut :

```
# ls -l
```

```
drwxr-xr-x 2 rizal rizal 4096 Jul 26 17:38 latihan
```

Kemudian kita ingin mengubah hak akses user pemilik menjadi read dan write saja (tanpa execute), sedangkan group dan other tidak punya hak akses apa-apa. Secara perhitungan, kebutuhan diatas nilainya akan menjadi seperti ini :

```
user = rw- = 6
```



```
grup = --- = 0
others = --- = 0
```

Sehingga sintaks yang bisa kita lakukan adalah :

```
# chmod 600 latihan
# ls -l
drw----- 2 rizal    rizal      4096 Jul 26 17:38 latihan
```

Contoh lain misal kita ingin mengubah direktori latihan dengan hak akses user pemilik full akses, untuk hak akses group diberikan read dan write saja tanpa execute, serta untuk others hanya read saja.

Secara perhitungan seharusnya menjadi seperti ini :

```
user = rwx = 7
grup = rw- = 6
others = r-- = 4
```

Sehingga sintaksnya menjadi seperti ini :

```
# chmod 764 latihan
# ls -l
drwxrw-r-- 2 rizal    rizal      4096 Jul 26 17:38 latihan
```

Gunakan opsi -R untuk merubah hak akses ownership keseluruhan dari isi direktori

```
# chmod -R 775 latihan
# ls -l
drwxrwxr-x 2 rizal    rizal      4096 Jul 26 17:38 latihan
```

3.11. Belajar Manajemen Paket

3.11.1. Apa itu Manajemen Paket?

Jika di Windows kita sering melakukan aktifitas terkait instalasi software, meng-uninstall software, mencari tahu apakah software tersebut sudah diinstall atau belum, versi berapakah yang sudah kita install, dll, ini adalah ilmu tersebut.

Seluruh kegiatan terkait manajemen aplikasi/software di Linux disebutnya sebagai ilmu manajemen paket.

Kita perlu memahami hal ini agar kita bisa melakukan berbagai manajemen terkait aplikasi/layanan yang ada di Server Linux kita. Misalnya ketika programmer di perusahaan kita membutuhkan untuk menginstall layanan database Mysql versi 5.5 di server Linux, maka kita harus tahu caranya. Dan juga kasus-kasus lain.

3.11.2. Melakukan Update Repository

Sistem instalasi aplikasi di Linux yang paling sering dipakai adalah dengan cara menginstall melalui repository. Konsepnya sama persis seperti kita menginstall aplikasi di Android atau di IOS. Dimana kita tidak perlu mendownload file-file installernya secara manual, tetapi kita tinggal buka Play Store untuk mendownload dan menginstall aplikasi yang tersedia disana. Play Store yang berisi banyak aplikasi ini disebut sebagai Repository jika di Linux.

Sebelum repository dapat digunakan, kita perlu mengupdatenya terlebih dahulu dengan perintah ini :

Ubuntu

```
# apt-get update
```

Centos

```
# yum update
```

3.11.3. Melakukan Instalasi Aplikasi/Service

Ini adalah kegiatan yang sama persis seperti kita menginstall aplikasi di Android atau IOS. Kita tinggal install aplikasi dari Repository, sistem akan secara otomatis melakukan pendownloadan dan instalasi sampai selesai. Namun jika kita ingin menginstall melalui installer offline juga bisa.

Perintahnya adalah seperti ini :

Ubuntu

```
# apt-get install namaaplikasi
# dpkg -i namaaplikasi.deb [Instalasi offline]
```

Centos

```
# yum install namaaplikasi
# rpm -ivh namaaplikasi.rpm [instalasi offline]
```

3.11.4. Mengecek apakah aplikasi sudah terinstall atau belum

Jika ingin mengetahui apakah sebuah aplikasi sudah berhasil terinstall atau belum bisa gunakan perintah ini :

Ubuntu

```
# dpkg -l | grep namaaplikasi
```

Centos

```
# rpm -qa | grep namaaplikasi
```

3.11.5. Menghapus Aplikasi/Service

Untuk menghapus aplikasi gunakan perintah berikut :

Ubuntu

```
# apt-get remove namaaplikasi
```

Centos

```
# yum remove namaaplikasi
```

3.11.6. Mencari Aplikasi dengan kata kunci tertentu

Bagaimana jika kita tidak mengetahui nama aplikasinya? Kita bisa cari aplikasi yang kita inginkan di repository dengan kata kunci tertentu dengan perintah berikut :

Ubuntu

```
# apt-cache search "kata kunci"
```

Centos

```
# yum search "kata kunci"
```

3.11.7. Upgrade aplikasi

Jika ingin mengupgrade versi aplikasi ke versi yang lebih baru, gunakan perintah ini :

Ubuntu

```
# apt-get upgrade
```

Centos

```
# yum update
```

3.11.8. Exercise

Teori

1. Kenapa kita perlu mempelajari Manajemen Paket?

Praktek

Waktu pengerjaan : 5 menit

Masing-masing peserta praktekkan soal berikut ini di kolom terminal Linux masing-masing sambil melakukan sharing screen agar instruktur dapat melihat hasilnya.

1. Carilah aplikasi dengan kata kunci “network mapper”.
2. Install aplikasi tersebut lalu cek versi berapa aplikasi yang diinstall tersebut.
3. Setelah itu hapuslah aplikasi tersebut.

3.12. Belajar Manajemen Layanan

3.12.1. Apa itu Manajemen Layanan?

Di Server Linux kita selalu berurusan dengan yang namanya layanan. Disini kita mempelajari bagaimana dasar-dasar cara untuk me-restart layanan,

menghentikan layanan, melihat status layanan tersebut aktif atau tidak, hingga cara menjalankan suatu layanan.

3.12.2. Mengecek Status Layanan

Ini adalah perintah untuk mengetahui apakah layanan yang ingin kita cek sedang berjalan atau berhenti.

Ubuntu

```
service namalayanan status
systemctl status namalayanan
```

Centos

```
service namalayanan status
systemctl status namalayanan
```

Misalnya :

```
rizal-Inspiron-5468 ~ # service ssh status
ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset:
enabled)
   Active: active (running) since Sat 2018-08-11 18:32:20 WIB; 5h 18min ago
   Process: 1839 ExecReload=/bin/kill -HUP $MAINPID (code=exited,
status=0/SUCCESS)
   Main PID: 1192 (sshd)
   CGroup: /system.slice/ssh.service
           └─1192 /usr/sbin/sshd -D
```

Disana tertulis bahwa service SSH sedang active (running). Artinya service ssh pada server Linux tersebut sedang berjalan.

3.12.3. Merestart Layanan

Perintah untuk memati-hidupkan layanan yang kita inginkan. Biasa digunakan sehabis kita melakukan suatu konfigurasi (biasanya konfigurasi baru berjalan setelah layanan direstart), atau ketika mengalami problem.

Ubuntu

```
service namalayanan restart
systemctl restart namalayanan
```

Centos

```
service namalayanan restart
systemctl restart namalayanan
```

3.12.4. Menghentikan Layanan

Perintah untuk mematikan layanan yang kita inginkan. Biasa digunakan jika ingin mematikan layanan yang sudah tidak dibutuhkan lagi, atau ketika ada salah satu prosedur yang mengharuskan mematikan layanan terkait.

Ubuntu

```
service namalayanan stop
systemctl stop namalayanan
```

Centos

```
service namalayanan stop
systemctl stop namalayanan
```

3.12.5. Menjalankan Layanan

Perintah untuk menghidupkan layanan yang kita inginkan. Layanan yang sudah diinstall di Linux Server tidak akan bisa dinikmati jika belum di jalankan.

Ubuntu

```
service namalayanan start
systemctl start namalayanan
```

Centos

```
service namalayanan start
systemctl start namalayanan
```

3.12.6. Mengecek & Membuat layanan aktif/tidak saat booting

Setiap layanan yang sudah diinstall di server Linux, bisa diatur apakah layanan tersebut auto start atau tidak setiap server booting.

Sebelum mengaktifkan atau menonaktifkan autostart, perintah untuk mengecek status autostartnya adalah ini :

Ubuntu

```
systemctl is-enabled namalayanan
```

Centos

```
systemctl is-enabled namalayanan
```

Jika hasilnya enabled, maka layanan tersebut sudah auto start.

Lalu untuk mengaktifkannya adalah :

Ubuntu

```
systemctl enable namalayanan
```

Centos

```
systemctl enable namalayanan
```

Dan untuk menonaktifkannya :

Ubuntu

```
systemctl disable namalayanan
```

Centos

```
systemctl disable namalayanan
```

3.12.7. Exercise

Teori

1. Kenapa kita perlu mempelajari Manajemen Layanan?

Praktek

Waktu pengerjaan : 5 menit

Masing-masing peserta praktekkan soal berikut ini di kolom terminal Linux masing-masing sambil melakukan sharing screen agar instruktur dapat melihat hasilnya.

1. Pertama-tama lakukan instalasi layanan webserver apache.
2. Jika sudah, buatlah layanan tersebut agar auto-start setiap booting.
3. Restart layanan webserver tersebut dan pastikan bahwa statusnya sudah running.

3.13. Roadmap Topologi yang akan dibangun

Pada bab ini kita sudah mulai masuk kedalam tahap pembangunan infrastructure awal, dimana kita akan coba untuk membangun sebuah webserver yang berisikan aplikasi web didalamnya. Kita akan lihat roadmap proses dimana kita sudah melangkah saat ini, dan topologi yang akan kita kerjakan pada awal infrastructure ini.

3.13.1. Roadmap Pembelajaran

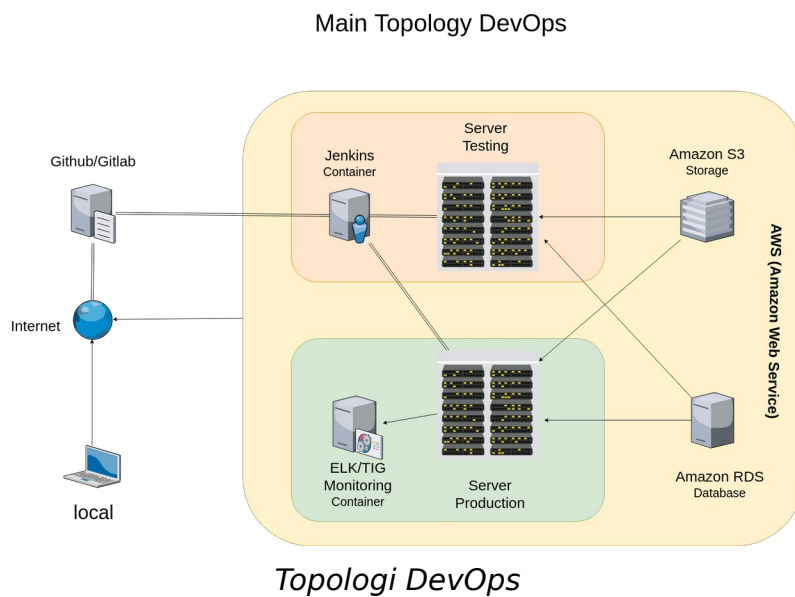


Roadmap Pembelajaran

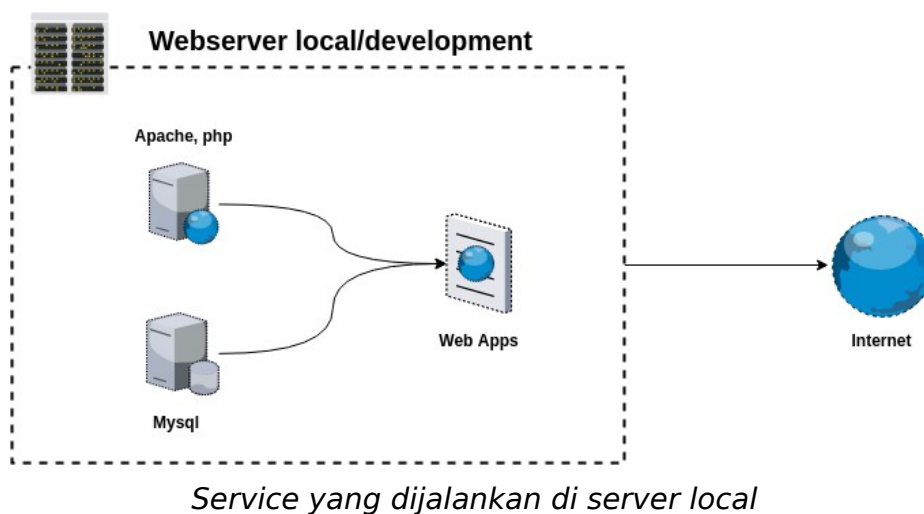
Jika kita lihat pada roadmap yang sudah kita bahas sebelumnya, kita kemarin sudah melewati tahap pertama yaitu bab 1 mengenal roadmap dan pembelajaran devops. Sekarang kita masuk tahap kedua yaitu belajar dasar jaringan dan membuat web server.

Untuk Dasar jaringan sudah kita pelajari di bab 2 dan untuk bab 3 ini kita membahas mengenai linux dan juga pembuatan web server. Ini adalah tahap awal dari sekian banyak rangkaian membangun infrastructure devops.

3.13.2. Topologi yang akan dibangun



Jika dilihat dari topologi diatas, kita sekarang akan mencoba untuk membangun bagian local. Atau bisa juga disebut sebagai server development. Dan berikut beberapa service yang akan kita installkan dengan topologi berikut.



3.14. Setup Webserver dan Aplikasi

3.14.1. Install Webserver dan Database Server

Webserver merupakan sebuah layanan yang ada dalam sebuah server yang berfungsi untuk membaca program berbasis web yang diantaranya adalah html, css, js, php, dll. Banyak aplikasi webserver yang bisa kita gunakan salah satunya adalah apache dan NginX.

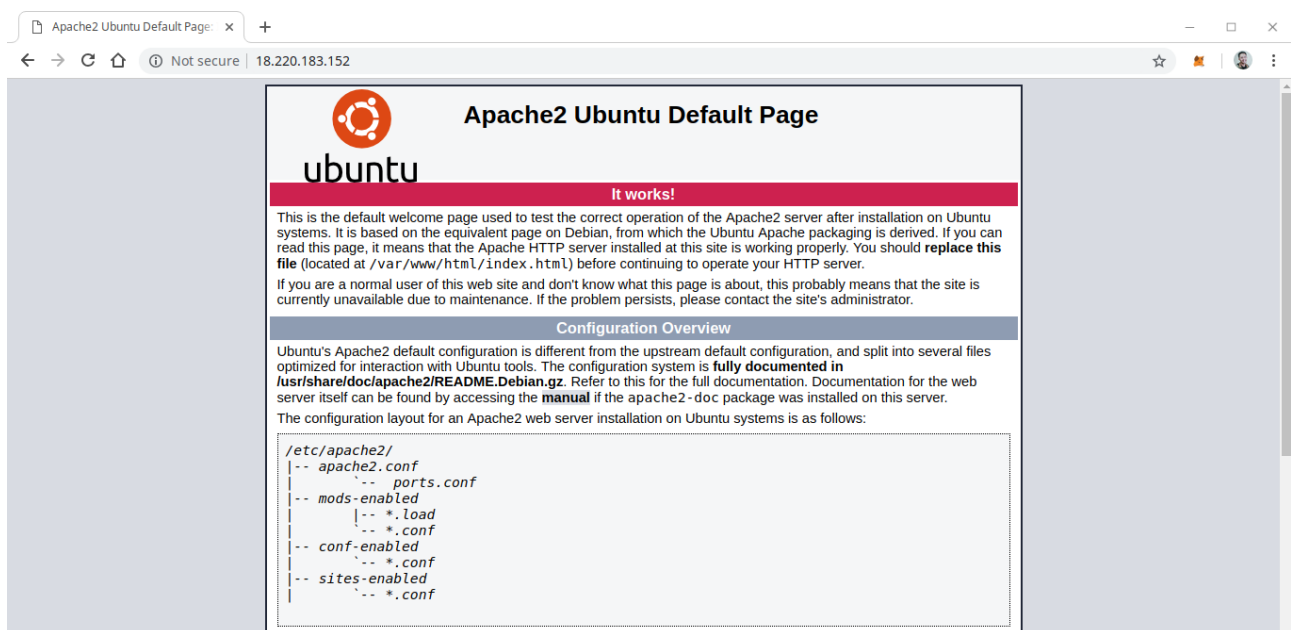
Pada bagian kali ini kita akan coba menginstallkan apache sebagai webservernya, serta menginstallkan php didalamnya menggunakan perintah berikut.

```
sudo apt-get install apache2 php php-mysql
```

Setelah webserver berhasil terinstall, maka selanjutnya kita install database mysql

```
sudo apt-get install mysql-server
```

Setelah selesai melakukan instalasi webserver dan juga database mysql, sekarang kita ujicoba apakah webserver sudah berjalan atau belum. Kita dapat memanggil alamat server tersebut di browser untuk melihat hasil dari instalasi webserver yang sudah kita lakukan.



Tampilan hasil webserver

Untuk mengecek keberhasilan dari database server, kita akan coba untuk masuk kedalam console mysql tersebut dengan menggunakan perintah berikut.

```
sudo mysql -u root -p
```

Masukan password mysql lalu tekan enter. Apabila belum membuat password, maka langsung enter saja dan akan muncul seperti berikut.

```
ubuntu@ip-172-31-12-225:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.25-0ubuntu0.18.04.2 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

Tampilan console mysql

Dengan begini persiapan webserver dan database server sudah selesai kita setup, selanjutnya kita akan coba melakukan setup pada web aplikasi yang akan kita integrasikan dengan webserver dan database server tersebut.

3.14.2. Setup Web Aplikasi Sosial Media

Setelah melakukan installasi webserver dan database server, sekarang kita akan melakukan setup pada Web Aplikasi di server local tersebut. Pertama kita harus menyiapkan terlebih dahulu database yang akan kita gunakan, masuk ke console mysql dan tambahkan user.

```
Sudo mysql -u root -p
create user 'devopscilasy'@'localhost' identified by '1234567890';
```

Perintah diatas menerangkan bahwa nama user yang akan kita buat adalah devopscilsy dan password usernya adalah 1234567890. Jika berhasil maka akan muncul seperti dibawah ini.

```
mysql> create user 'devopscilsy'@'localhost' identified by '1234567890';
Query OK, 0 rows affected (0.00 sec)
```

Selanjutnya kita berikan hak akses pada user yang sudah kita buat tersebut.

```
grant all privileges on *.* to 'devopscilsy'@'localhost';
```

Setelah selesai memberikan hak akses pada user, kita dapat keluar dari console dengan menggunakan perintah **lq**. Lalu coba kembali masuk kedalam konsole mysql menggunakan user baru tersebut, dan buat database dengan nama dbsosmed didalamnya.

```
sudo mysql -u devopscilsy -p
create database dbsosmed;
```

Cek keberadaan database yang sudah kita buat dengan menggunakan perintah berikut.

```
show databases;
```

```
mysql> create database dbsosmed;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| dbsosmed |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)
```

Selanjutnya keluar dari database dengan menggunakan perintah **lq**.

Setelah kita melakukan persiapan database, maka selanjutnya kita download web aplikasi untuk dipasangkan pada server lokal kita. Pertama kita pindah direktori dulu ke direktori utama, lalu download aplikasi seperti berikut.

```
cd
wget https://github.com/sdcilsy/sosial-media/archive/master.zip
```

Selanjutnya kita ekstrak file aplikasi tersebut, tetapi sebelumnya kita harus menginstallkan paket unzip terlebih dahulu.

```
sudo apt-get install unzip
unzip master.zip
```

Jika berhasil akan muncul seperti dibawah ini.

```
ubuntu@ip-172-31-12-225:~$ unzip master.zip
Archive: master.zip
ef65f8054e6639b48ee569a23bad2429c8d8be54
  creating: sosial-media-master/
  inflating: sosial-media-master/.htaccess
  extracting: sosial-media-master/auth.php
  inflating: sosial-media-master/config.php
  creating: sosial-media-master/css/
  inflating: sosial-media-master/css/bootstrap-grid.css
  inflating: sosial-media-master/css/bootstrap-grid.css.map
  inflating: sosial-media-master/css/bootstrap-grid.min.css
  inflating: sosial-media-master/css/bootstrap-grid.min.css.map
  inflating: sosial-media-master/css/bootstrap-reboot.css
  inflating: sosial-media-master/css/bootstrap-reboot.css.map
  inflating: sosial-media-master/css/bootstrap-reboot.min.css
  inflating: sosial-media-master/css/bootstrap-reboot.min.css.map
```

Selanjutnya kita akan pindahkan seluruh isi konten yang ada pada direktori *sosial-media-master* yang sudah kita ekstrak tadi ke */var/www/html/* agar konten aplikasi tersebut dapat kita akses. Sebelumnya hapus dulu file *index.html* yang ada pada direktori */var/www/html/*.

```
sudo rm /var/www/html/index.html
sudo mv sosial-media-master/* /var/www/html
```

Kita bisa check apakah semua konten sudah dipindahkan ke direktori */var/www/html/* dengan perintah *ls* seperti berikut.

```
ubuntu@ip-172-31-12-225:~$ ls /var/www/html/
auth.php  config.php  css  dump.sql  img  index.php  js  login.php  logout.php  register.php  timeline.php
ubuntu@ip-172-31-12-225:~$ ls sosial-media-master/
ubuntu@ip-172-31-12-225:~$
```

Dalam seluruh isi konten bisa kita lihat ada file *dump.sql*, file tersebut merupakan data export dari database konten web sosmed ini. Sekarang kita akan coba import file tersebut ke database yang sudah kita buat sebelumnya yaitu *dbsosmed*.

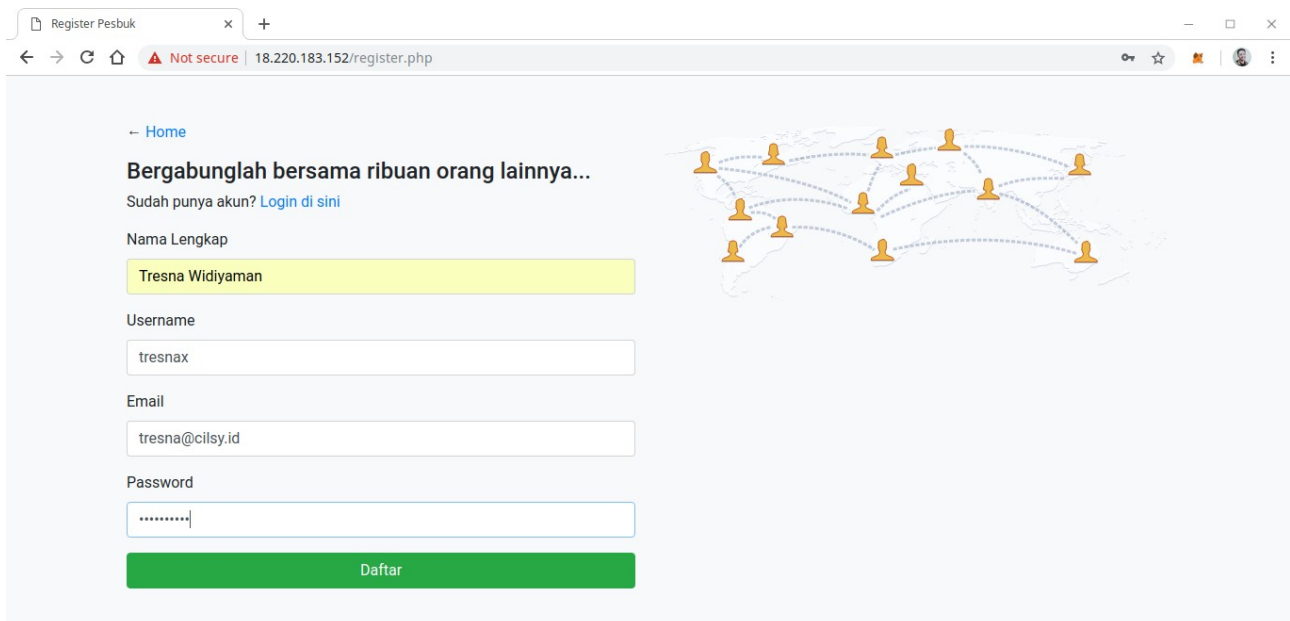
```
cd /var/www/html/
sudo mysql -u devopscilsy -p dbsosmed < dump.sql
```

Setelah selesai, sekarang kita bisa ujicoba web aplikasi tersebut dengan memanggil kembali alamat server local yang kita miliki.



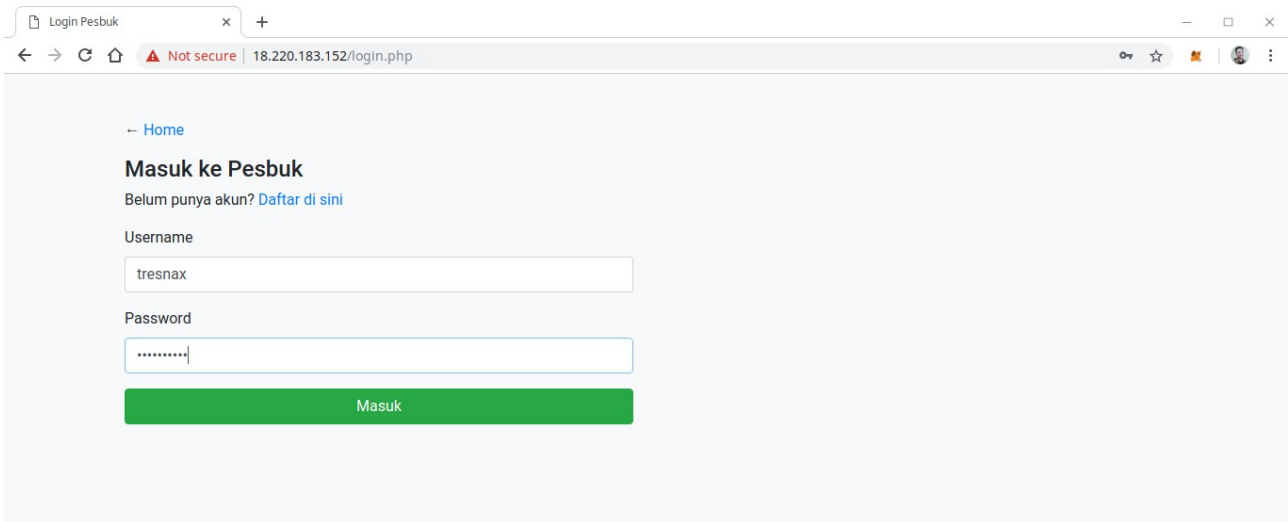
Halaman utama Aplikasi Web

Kita ujicoba aplikasi tersebut dengan memilih tombol daftar untuk mendaftarkan akun baru pada aplikasi sosmed pesbuk.



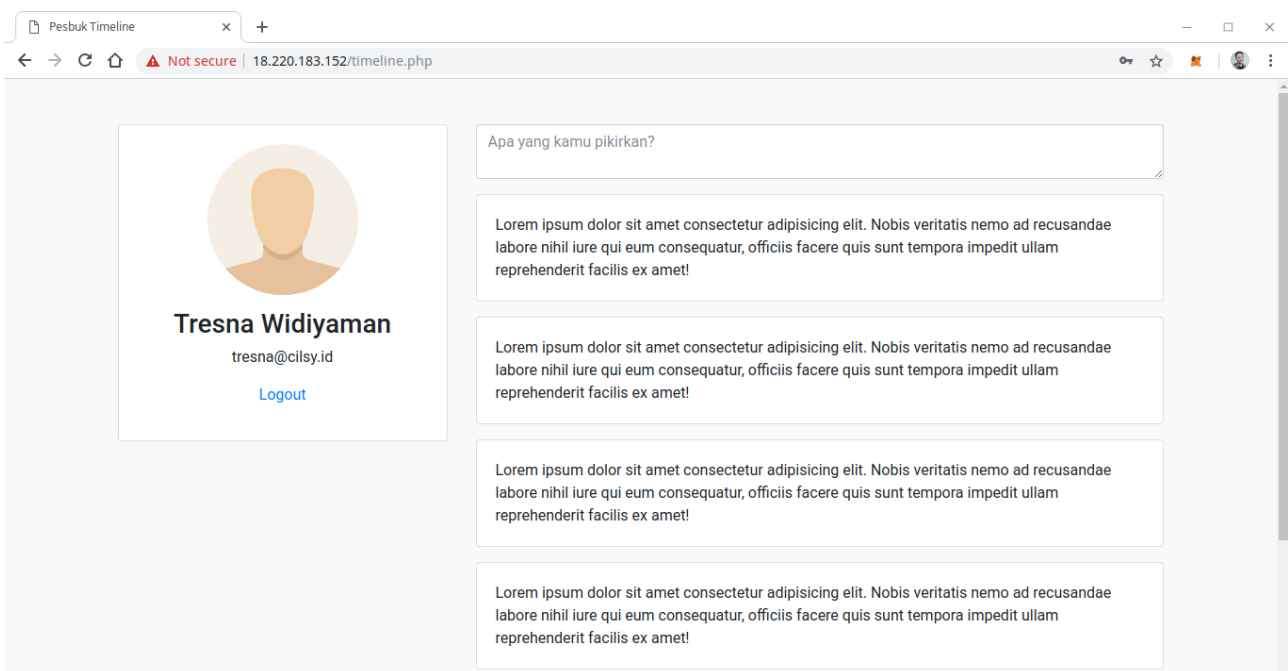
Halaman Pendaftaran Akun

Setelah berhasil mendaftar, kalian akan diarahkan langsung ke form login aplikasi. Lakukan login dengan akun tersebut hingga kalian diarahkan pada halaman dashboard aplikasi.



The screenshot shows a web browser window with the address bar displaying "18.220.183.152/login.php". The page title is "Login Pesbuk". Below the title, there is a link "Home" and a heading "Masuk ke Pesbuk". A sub-link "Belum punya akun? [Daftar di sini](#)" is present. The login form consists of two input fields: "Username" with the value "tresnax" and "Password" with masked characters "*****". A green button labeled "Masuk" is at the bottom of the form.

Halaman Login Akun



The screenshot shows a web browser window with the address bar displaying "18.220.183.152/timeline.php". The page title is "Pesbuk Timeline". On the left, there is a user profile card for "Tresna Widiyaman" with the email "tresna@cilsy.id" and a "Logout" link. On the right, there is a text input field with the placeholder "Apa yang kamu pikirkan?". Below this, there are four identical text boxes, each containing a placeholder Lorem Ipsum text.

Halaman Dashboard Aplikasi

Selamat kamu sudah menyelesaikan sebuah infrastructure untuk sebuah web aplikasi sosial media pesbuk dengan menyertakan layanan webserver dan juga database server.

3.15. Pemrograman Dasar Shell

3.15.1. Apa itu Shell ?

Shell adalah sebuah program tempat dimana pengguna dapat mengetikkan perintah-perintah yang diinginkan. Shell juga memungkinkan pengguna untuk menyusun sekumpulan perintah ini pada sebuah atau beberapa file untuk dieksekusi sebagai program.

```

    /bin/bash
( Today is National Existential Ennui )
( Awareness Day.)
-----
0
0
      .--.
     |o o|
     |: / |
   (( ))
  (/ \)
 (=)
metalx1000@taka ~ $ man pv
metalx1000@taka ~ $ echo "This is some text"
This is some text
metalx1000@taka ~ $ echo "This is some text"
This is some text
metalx1000@taka ~ $ echo "This is some text"|pv -qL 10
This is some text
metalx1000@taka ~ $ echo "This is some text"|pv -qL 10
This is some text
metalx1000@taka ~ $ echo "This is some text"|pv -qL 20
This is some text
metalx1000@taka ~ $ echo "This is some text"█iv -qL 100
This is some text
metalx1000@taka ~ $ █

```

Ilustrasi Shell

Shell memiliki berbagai macam jenis, yang diantaranya adalah Bourne shell(sh), C shell(csh), Korn shell(ksh), Bourne again shell(bash), Dsb.

Masing - masing shell mempunyai kelebihan dan kekurangan yang mungkin lebih didasarkan pada kebutuhan pemakai yang makin hari makin meningkat. Untuk modul yang digunakan di Sekolah DevOps Cilsy, shell yang digunakan adalah bash shell, yang merupakan pengembangan dari Bourne shell(sh) dan mengambil beberapa feature (keistimewaan) dari C shell serta Korn shell. Bash shell merupakan shell yang cukup banyak digunakan pemakai Linux karena kemudahan serta banyaknya fasilitas perintah yang disediakan.

3.15.2. Pemrograman Shell

Pemrograman Shell itu sendiri adalah teknik untuk menyusun atau mengelompokkan beberapa perintah shell menjadi kumpulan perintah yang melakukan tugas tertentu sesuai tujuan kita sebagai penyusunnya. Kelebihan shell di Linux dibanding sistem operasi lain adalah bahwa shell di linux memungkinkan kita untuk menyusun serangkaian perintah ini seperti halnya bahasa pemrograman. Yaitu melakukan proses I/O, menyeleksi, logika-logika if, looping, membuat fungsi, dsb seperti proses - proses yang umumnya dilakukan oleh suatu bahasa pemrograman. Dengan kata lain Shell di Linux membuat kita dapat membuat sebuah program seperti halnya bahasa pemrograman. Untuk pemrograman shell ini sendiri kita sering menyebutnya sebagai Shell Scripting.

3.15.3. Perintah Dasar Shell

Sebelum mempelajari pemrograman Shell Scripting di Linux sebaiknya kita telah mengetahui dan terbiasa menggunakan berbagai perintah-perintah dasar Shell di Linux. Pada dasarnya perintah dasar Shell yang sering digunakan adalah adalah seluruh command-command dan konsep yang sudah kita pelajari pada bagian sebelumnya. Sehingga jika kita sudah memahami dan terbiasa dengan command-command Linux tersebut, kita akan jauh lebih mudah dalam melakukan pemrograman Shell Scripting ini.

3.15.4. Bash Scripting Sederhana.

Sekarang kita coba membuat script baru dengan mengetikan perintah dibawah ini pada prompt shell :

```
[ubuntu@linux$]echo "Script shell pertamaku di linux"
Script shell pertamaku di linux
```

String yang diapit tanda kutip ganda (double quoted) akan ditampilkan pada layar kita, echo adalah statement (perintah) built-in bash yang berfungsi menampilkan informasi ke standard output yang defaultnya adalah layar.

Jika kita ingin mengulangi perintah sebelumnya, kita tidak perlu mengerjakan perintahnya kembali, cukup dengan fasilitas history dengan menggunakan tombol panah (atas bawah) kita sudah dapat mengulangi perintah tersebut.

Bagaimana jika perintah yang kita masukan berupa kumpulan perintah yang cukup banyak? tentunya jika kita harus mengetikkan satu-persatu secara terus menerus pada prompt shell akan sangat merepotkan. Untuk itulah sebaiknya perintah-perintah yang banyak tersebut kita simpan ke sebuah file. Dimana file ini dapat kita panggil/eksekusi kapanpun diinginkan.

Kita dapat melakukannya dengan membuat sebuah file misalnya bernama tes dengan perintah nano/vim, lalu didalam file tersebut isi dengan script berikut :

```
#!/bin/bash
echo "Hello, apa khabar"
```

Simpan file tersebut, lalu ubah permission file test tersebut agar memiliki hak akses untuk execute (x) menggunakan perintah chmod.

```
[ubuntu@linux$]chmod 755 tes
```

Selanjutnya adalah jalankan file tadi dengan perintah berikut :

```
[ubuntu@linux$] ./tes
```

Tanda #! pada /bin/bash dalam script tes adalah perintah yang diterjemahkan ke kernel linux untuk mengeksekusi path yang disertakan dalam hal ini program bash pada direktory /bin, sebenarnya tanpa mengikutkan baris tersebut anda tetap dapat mengeksekusi script bash, dengan catatan bash adalah shell aktif. atau dengan mengetikkan bash pada prompt shell.

```
[ubuntu@linux$]bash tes
```

Selanjutnya kita coba membuat script shell yang menampilkan informasi berikut :

1. Waktu System
2. Info tentang komputer
3. Jumlah pemakai yang sedang login di system

Berikut merupakan contoh bash scriptnya :

```
#!/bin/bash
#myinfo

#membersihkan tampilan layar
clear

#menampilkan informasi
echo -n "Waktu system   :"; date
echo -n "Anda           :"; whoami
echo -n "Banyak pemakai :"; who | wc -l
```

Simpan file tersebut dengan nama “myinfo” selanjutnya jangan lupa untuk merubah permission file myinfo agar dapat diakses oleh kita :

```
[ubuntu@linux$]chmod 755 myinfo
[ubuntu@linux$]. ./myinfo
Waktu system   : Sat Nov 25  22:57:15 BORT 2001
Anda           : ubuntu
Banyak pemakai : 2
```

3.15.5. Pemakaian Variable

Secara sederhana variabel adalah ibarat sebuah kotak penyimpanan yang dapat menyimpan suatu isi atau nilai. Isi atau nilai dapat berupa bermacam-macam. Bisa berupa angka, berupa huruf, kalimat, command, dll. Variable pada shell scripting sifatnya juga sama dengan variable pada bahasa pemrograman pada umumnya. Dalam dokumentasi ini kita membagi variabel menjadi 2 kategori, yaitu Positional Parameter Variable dan User Defined Variable. Sebenarnya masih ada 1 kategori variable lain, yaitu Environment Variable, namun variable ini belum terlalu perlu untuk dipahami di awal-awal pembelajaran Shell Scripting ini.

3.15.5.1. Positional Parameter

Positional Parameter merupakan variabel yang digunakan shell untuk menampung argumen yang diberikan pada saat shell dijalankan. Lebih jelasnya dapat kita lihat pada bagian script berikut.

```
#!/bin/bash
#argumen1

echo $1 adalah salah satu $2 populer di $3
```

Hasilnya adalah sebagai berikut.

```
[ubuntu@linux$]./argumen1 bash shell linux
bash adalah salah satu shell populer di linux
```

Ada 3 argumen yang disertakan pada script argumen1 yaitu bash, shell, dan linux, masing2 argumen akan disimpan pada variabel 1,2,3 sesuai posisinya.

3.15.5.2. User Defined Variable

User Defined Variable adalah variabel yang didefinisikan sendiri oleh pembuat script sesuai dengan kebutuhannya. Beberapa hal yang perlu diperhatikan dalam mendefenisikan User Defined Variable adalah sebagai berikut.

1. Dimulai dengan huruf atau underscore.
2. Hindari pemakaian spesial karakter seperti *,\$, #, dll.
3. Bash bersifat case sensitive, yaitu antara huruf besar dan kecil dianggap berbeda, antara huruf a berbeda dengan A, nama berbeda dengan Nama, NaMa, dsb.

Untuk menentukan nilai variabel gunakan operator assignment (pemberi nilai) "=". Bisa menggunakan contoh script berikut ini :

```
myos="linux"           #double-quoted
nama='penguin'        #single-quoted
hasil=`ls -l`;        #back-quoted
angka=12
```

Kalau kita perhatikan ada 3 tanda kutip yang kita gunakan untuk memberikan nilai string ke suatu variabel, masing-masing tanda kutip itu memiliki sedikit perbedaan.

Dengan kutip ganda (double-quoted), bash mengizinkan kita untuk menyisipkan variabel di dalamnya. Contohnya :

```
#!/bin/bash
nama="penguin"
kata="Hi $nama, apa khabarmu" #menyisipkan variabel nama
echo $kata;
```

Hasilnya adalah sebagai berikut :

```
Hi penguin, apa khabarmu
```

Kita dapat melihat bahwa di dalam variable kata, terdapat juga variable nama yang disisipkan.

Dengan kutip tunggal (single-quoted), akan ditampilkan apa adanya. Sehingga jika kita memasukkan variable didalamnya, tidak akan dianggap sebagai variable. Contohnya adalah sebagai berikut :

```
#!/bin/bash
nama="penguin"
kata='Hi $nama, apa khabarmu' #menyisipkan variabel nama
echo $kata;
```

Hasilnya adalah sebagai berikut :

```
Hi $nama, apa khabarmu
```

Dengan kutip terbalik (double-quoted), bash menerjemahkan sebagai perintah/command yang akan dieksekusi, contohnya :

```
#!/bin/bash

hapus=`clear`;
isi=`ls -l`; #hasil perintah ls -l disimpan di var isi
#hapus layar
echo $hapus
```

```
#ls -l
echo $isi
```

Hasilnya bisa kita lihat sendiri dengan mencoba perintah tersebut. Yaitu ketika kita memanggil variable \$hapus, kita berarti memerintahkan agar shell mengeksekusi perintah/command clear. Begitupun dengan variable \$isi.

Agar lebih jelas lagi perbedaannya, kita coba untuk menggabungkan beberapa penggunaan quote serta penggunaan Positional Parameter Variable menjadi satu :

```
#!/bin/bash
#varuse

nama="ubuntu"
OS='linux'
distro="macam-macam, bisa slackware, redhat, mandrake, debian, suse, dll"
pc=1
hasilnya=`ls -l $0`

clear
echo -e "Hi $nama,\npake $OS\nDistribusi, $distro\nkomputernya, $pc buah"
echo "Hasil ls -l $0 adalah =$hasilnya"
```

Maka hasilnya akan menjadi seperti berikut ini.

```
[ubuntu@linux$]./varuse
Hi ubuntu,
pake linux Distribusi, macam-macam, bisa
slackware,redhat,mandrake,debian,suse,dll
komputernya, 1 buah
Hasil ls -l ./varuse adalah -rwxr-xr-x 1 ubuntu users 299 Nov 21 06:24 ./varuse
```

Untuk operasi matematika ada 3 cara yang dapat kita gunakan, dengan statement builtin *let* atau *expr* atau perintah substitusi seperti contoh berikut :

```
#!/bin/bash
#mat1
```

```
a=10
b=5
#memakai let
let jumlah=$a+$b
let kurang=$a-$b
let kali=$a*$b
#memakai expr
bagi=`expr $a / $b`
#memakai perintah substitusi $((ekspresi))
modul=$(( $a%$b )) #sisanya pembagian
echo "$a+$b=$jumlah"
echo "$a-$b=$kurang"
echo "$a*$b=$kali"
echo "$a/$b=$bagi"
echo "$a%$b=$mod"
```

Hasilnya adalah seperti berikut.

```
[ubuntu@linux$] ./mat1
10+5=15
10-5=5
10*5=50
10/5=2
10%5=0
```

Fungsi *expr* begitu berguna baik untuk operasi matematika ataupun string contohnya :

```
[ubuntu@linux$] mystr="linux"
[ubuntu@linux$] expr length $mystr
5
```

Mungkin kalian bertanya-tanya, apakah bisa variabel yang akan digunakan dideklarasikan secara eksplisit dengan tipe data tertentu? mungkin seperti C atau pascal, untuk hal ini oleh Bash disediakan statement *declare* dengan opsi *-i* hanya untuk data integer (bilangan bulat). Contohnya sebagai berikut.

```
#!/bin/bash
```

```
declare -i angka
angka=100;
echo $angka;
```

3.15.6. Exercise

1. Buatlah sebuah script yang dapat menampilkan status layanan dari masing-masing layanan berikut dengan menggunakan teknik Variable Position Parameter :
 1. sshd
 2. mariadb
 3. httpd

3.15.7. Belajar Logika Pemrograman pada Bash Script

Materi ini mengajarkan agar kita lebih terbiasa dengan pola pikir logika seorang programmer. Kita dapat menggunakan berbagai logika ini untuk mencapai tujuan-tujuan tertentu bukan hanya saat membuat file-file script saja, namun untuk berbagai kasus selama menjadi seorang DevOps nantinya.

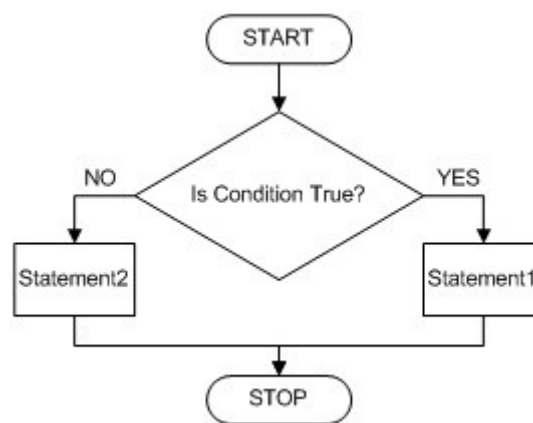
Disini kita akan mempelajari beberapa contoh kondisi logika seperti If Conditional dan Looping. Misalnya seperti : Jika kondisi A terpenuhi Maka lakukan command B, Ulangi kondisi A sebanyak X, Ulangi command A sampai kondisi B terpenuhi, dsb.

Contoh realnya misal kita dapat membuat script yang berisi kondisi : Jika Programmer sudah upload code mereka, Maka nyalakan server A. Atau Jika user yang akses Server sudah mencapai 1000 orang, maka aktifkan Server tambahan B.

Logika-logika ini akan cukup sering digunakan selama Anda menjadi DevOps. Karena seorang DevOps tidak hanya orang yang pandai jaringan, tapi juga memahami konsep-konsep programming.

3.15.7.1. If Conditional

Sederhananya If Conditional ini menggunakan pola berpikir : **Jika A Maka lakukan B.**



Flow chart If Condition

Contohnya :

- jika kondisi A terpenuhi, Maka lakukan B.
- Jika $1+1 = 2$ maka lakukan C
- Jika User A belum masuk group Admin, maka masukkan user A kedalam group Admin.

Berikut kita akan coba aplikasikan If Conditional pada bash script. Sintaks dasarnya adalah :

```

if [ CONDITION ]; then
COMMANDS;
else
OTHER-COMMANDS;

```

```
fi
```

Dimana condition ini terdapat cukup banyak kombinasi yang bisa digunakan diantaranya :

Operator	Description
! EXPRESSION	The EXPRESSION is false.
-n STRING	The length of STRING is greater than zero.
-z STRING	The length of STRING is zero (ie it is empty).
STRING1 = STRING2	STRING1 is equal to STRING2
STRING1 != STRING2	STRING1 is not equal to STRING2
INTEGER1 -eq INTEGER2	INTEGER1 is numerically equal to INTEGER2
INTEGER1 -ne INTEGER2	INTEGER1 is numerically not equal to INTEGER2
INTEGER1 -gt INTEGER2	INTEGER1 is numerically greater than INTEGER2
INTEGER1 -lt INTEGER2	INTEGER1 is numerically less than INTEGER2
-d FILE	FILE exists and is a directory.
-e FILE	FILE exists.
-r FILE	FILE exists and the read permission is granted.
-s FILE	FILE exists and its size is greater than zero (ie. it is not empty).
-w FILE	FILE exists and the write permission is granted.
-x FILE	FILE exists and the execute permission is granted.

Misal kita ingin membuat script sebagai berikut :

```
#!/bin/bash
Y=1
if [ $Y -eq 2 ]; then
    echo "Hebat"
else
    echo "Keren"
fi
```

Script itu artinya : Jika variabel Y sama dengan 2, maka tampilkan pesan Hebat. Jika selain 2 maka tampilkan Keren. Kita bisa coba ubah-ubah isi variable Y tersebut untuk memastikan apakah kondisi If sudah berjalan.

Kita bisa coba aplikasikan ilmu berbagai command, manipulasi teks dan penggunaan variable yang sudah kita pelajari sebelumnya dengan if conditional ini. Misal kita bisa buat script untuk mencari tau apakah seorang user bernama cilsy sudah masuk ke dalam grup sudo atau belum, jika belum maka masukkan user tersebut ke dalam grup sudo tersebut.

Nb : Pertama-tama buatlah terlebih dahulu user cilsy jika belum ada

Selanjutnya buatlah file bernama scriptbelajar2.sh yang berisi script berikut :

```
#!/bin/bash
grupsudo="sudo"
groups $1 | grep $grupsudo

if [ $? -ne 0 ]; then
    echo "User $1 belum masuk grup sudo, kita masukkan terlebih dahulu.."
    sudo usermod -aG $grupsudo $1
    echo "Berhasil!"
else
    echo "User $1 sudah masuk grup sudo"
fi
```

Jangan lupa setelah itu kita berikan hak akses execute :

```
chmod 755 scriptbelajar2.sh
```

Selanjutnya coba jalankan script tersebut sambil kita cek perbedaannya antara sebelum masuk grup sudo dan sesudah :

```
rizal@rizal-Inspiron-5468 ~ $ groups cilsy
cilsy : cilsy
rizal@rizal-Inspiron-5468 ~ $ ./scriptbelajar2.sh cilsy
User cilsy belum masuk grup sudo, kita masukkan terlebih dahulu..
Berhasil!
rizal@rizal-Inspiron-5468 ~ $ groups cilsy
cilsy : cilsy sudo
rizal@rizal-Inspiron-5468 ~ $ ./scriptbelajar2.sh cilsy
cilsy : cilsy sudo
```

User cilsy sudah masuk grup sudo

Pada script tersebut kita menggunakan beberapa teknik yang sudah kita pelajari sebelumnya seperti User Defined Variable, dan Positional Parameter Variable. Kita juga menggunakan manipulasi teks menggunakan grep.

Yang perlu diperhatikan sedikit adalah pada bagian Condition disana kita menggunakan kondisi \$? . \$? ini adalah kondisi yang bisa menangkap status dari ada atau tidaknya output dari command yang sudah kita lakukan. Jika command tersebut ada outputnya dan berhasil, maka akan menampilkan angka 0. Jika tidak ada outputnya atau gagal, maka menampilkan angka 1. Sehingga pada kondisi diatas, bisa diartikan kita menangkap status dari command :

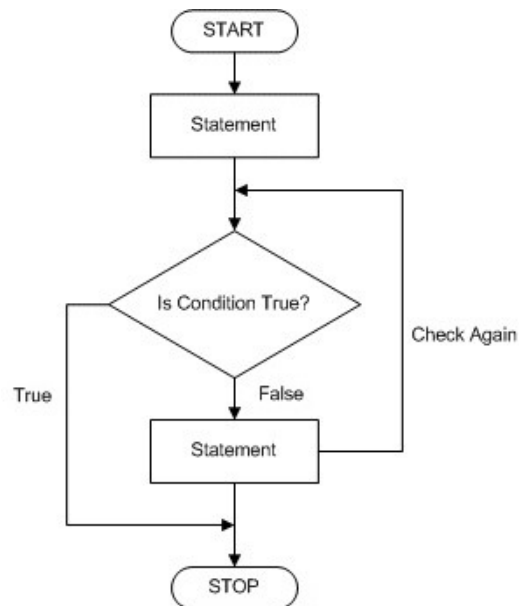
`groups $1 | grep $grupsudo`

Jika Command tersebut menampilkan output (bahwa user Cilsy ada di grup sudo), maka hasil \$? menjadi 0. Jika tidak, maka menjadi 1.

Sehingga kesimpulan dari kondisi yang kita gunakan adalah : Jika status dari command “groups \$1 | grep \$grupsudo” TIDAK SAMA (-ne) dengan 0, maka

3.15.7.2. While Looping

Pola pikir While Looping adalah seperti ini : **Lakukan X terus menerus selama kondisi Y belum terpenuhi.**



Flow Chart While Looping

Contohnya :

- Kurangi angka 100 selama hasilnya belum 1.
- Lakukan Command Y Selama command X belum dilaksanakan.
- Jangan restart server selama user yang akses belum 1000 orang.

Sintaks dasar dari While Looping adalah :

```
while [ CONDITION ]
do
COMMAND
done
```

Berikut contoh sederhana dari penggunaan While Looping :

```
#!/bin/bash
# Basic while loop
counter=1
while [ $counter -le 10 ]
do
    echo "Angka : $counter"
```

```
((counter++))
done
echo All done
```

Script diatas Melakukan penambahan variable counter sebanyak 1 terus menerus hingga mencapai 10. Dan pada setiap perulangannya kita menampilkan pesan Angka : counter.

```
rizal@rizal-Inspiron-5468 ~ $ ./scriptbelajar4.sh
Angka : 1
Angka : 2
Angka : 3
Angka : 4
Angka : 5
Angka : 6
Angka : 7
Angka : 8
Angka : 9
Angka : 10
All done
```

3.15.7.3. For Looping

Pola pikir For Looping agak sedikit berbeda dengan looping While. Dimana untuk For cara berpikirnya adalah seperti ini : **Untuk setiap X pada List yang diberikan, lakukan serangkaian Command Y. Dimana X nilainya akan sama dengan setiap List yang diberikan.**

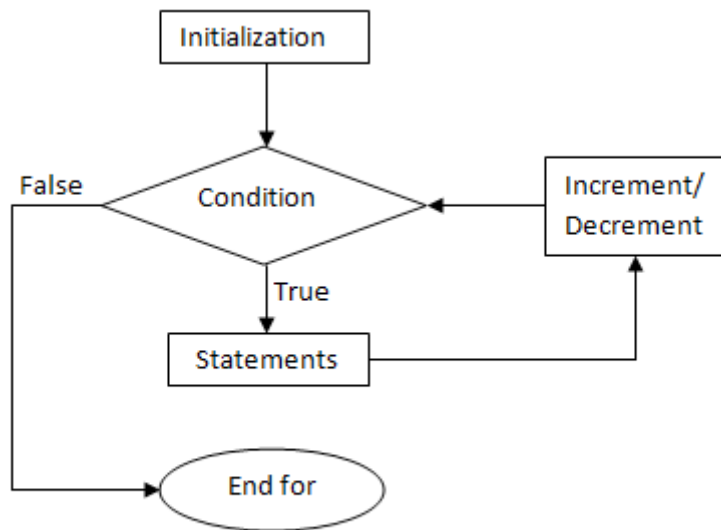


fig: Flowchart for for loop

Flow Chart For Looping

Contohnya :

- Masukkan user A kedalam setiap grup yang ada di server.
- Cari file bernama X sebanyak 1000 file yang ada di folder Z.
- Hapus angka 1 pada setiap nama file yang ada di folder Z.

Sintaks dasar dari For Looping adalah :

```

for VARIABLE in LISTS
do
COMMANDS
done
  
```

Berikut contoh sederhana dari penggunaan For Looping :

```

#!/bin/bash
# Basic range in for loop
for value in {1..5}
do
    touch /tmp/cilsy$value
done
  
```

```
echo All done
```

Pada script tersebut, kita membuat teks bernama cilsy+angka sebanyak 5x di folder /tmp. Sehingga terbuatlah file cilsy1, cilsy2, cilsy3 hingga cilsy5.

Contoh kedua misalnya kita akan membuat List dari masing-masing nama file yang ada di suatu folder kemudian melakukan sesuatu terhadap file tersebut pada setiap perulangannya.

Misal buatlah 3 buah file bernama seperti ini di folder /tmp/latihanscript :

```
mkdir /tmp/latihanscript
touch /tmp/latihanscript/forloop1.txt
touch /tmp/latihanscript/forloop2.txt
touch /tmp/latihanscript/forloop3.txt
```

Lalu buat file script bernama scriptbelajar6.sh dengan isi seperti ini :

```
#!/bin/bash
lihatdirektori=`ls /tmp/latihanscript`
counter=1

for value in $lihatdirektori
do
    mv /tmp/latihanscript/$value /tmp/latihanscript/hasilloop$counter.txt
    ((counter++))
done
echo All done
```

Cobalah eksekusi script tersebut dan lihat perbedaan antara sebelum dan sesudah :

```
rizal@rizal-Inspiron-5468 ~ $ chmod 755 scriptbelajar6.sh
rizal@rizal-Inspiron-5468 ~ $ ls /tmp/latihanscript/
forloop1.txt forloop2.txt forloop3.txt
rizal@rizal-Inspiron-5468 ~ $ ./scriptbelajar6.sh
All done
rizal@rizal-Inspiron-5468 ~ $ ls /tmp/latihanscript/
hasilloop1.txt hasilloop2.txt hasilloop3.txt
```


Script tersebut membuat variable value akan menangkap nama masing-masing nama file pada setiap perulangannya. Cara kerjanya kira-kira seperti ini :

```
mv /tmp/latihanscript/forloop1.txt /tmp/latihanscript/hasilloop1.txt
mv /tmp/latihanscript/forloop2.txt /tmp/latihanscript/hasilloop2.txt
mv /tmp/latihanscript/forloop3.txt /tmp/latihanscript/hasilloop3.txt
```

3.15.8. Exercise

1. Buatlah sebuah script yang dapat menghitung pada baris seberapa kata “shadow” muncul pada file /etc/group. Jika nilai barisnya lebih dari 10 maka tampilkan pesan “Lebih dari 10”, jika dibawah 10 maka tampilkan pesan “kurang dari 10”.

3.16. Membuat Automasi Setup Webserver dengan Bash

3.16.1. Automasi instalasi webserver

Pada beberapa bagian sebelumnya kita sudah mempelajari bagaimana kita melakukan instalasi webserver dan juga setup web aplikasi, begitupun dengan pemrograman dasar menggunakan bahasa pemrograman bash yang ada pada linux.

Pada bagian ini kita akan menyatukan kedua hal yang sudah kita pelajari, jika sebelumnya kita sudah melakukan instalasi webserver secara manual, sekarang kita akan coba lakukan instalasi webserver secara otomatis menggunakan bash script.

Tujuan dari dibuatnya script automasi ini adalah untuk mempermudah kita pada saat membuat project baru yang menggunakan environment sistem yang masih sama. Kita tidak perlu lama mengetikan apa saja yang harus kita install,

kita hanya perlu menjalankan satu script sehingga environment yang kita butuhkan sudah otomatis terbentuk.

```
#!/bin/bash

jawaban="Y"

read -p "Apakah kamu yakin akan menginstall webserver ? (Y/n) " pilih;

if [ $pilih == $jawaban ];
then
    echo "Menyiapkan Installasi Web server"
    sudo apt-get update
    echo "Melakukan Installasi Webserver"
    sudo apt-get install -y apache2 php php-mysql
    echo "Melakukan Installasi Database Server"
    sudo apt-get install -y mysql-server
    echo "Installasi Selesai"
    exit 0
else
    echo "Installasi dibatalkan"
    exit 1
fi
```

Script diatas akan memberikan pilihan kepada kita untuk menjalankan instalasi atau tidak dengan memberikan perintah Y untuk ya dan n untuk tidak. Apabila kita memilih ya maka script tersebut akan menjalankan perintah untuk menginstall apache, php, dan mysql. Jika aplikasi sudah terinstall dikomputer kita, maka akan muncul seperti dibawah ini.

```
ubuntu@ip-172-31-12-225:~$ sudo ./install2.sh
Apakah kamu yakin akan menginstall webserver ? (Y/n) Y
Menyiapkan Instalasi Web server
Reading package lists... Done
Building dependency tree
Reading state information... Done
php is already the newest version (1:7.2+60ubuntu1).
php-mysql is already the newest version (1:7.2+60ubuntu1).
apache2 is already the newest version (2.4.29-1ubuntu4.5). Web server"
0 upgraded, 0 newly installed, 0 to remove and 146 not upgraded.
Instalasi Selesai
ubuntu@ip-172-31-12-225:~$ sudo apt-get update
```

Proses instalasi menggunakan script bash

3.16.2. Automasi Setup Web Aplikasi

Selain melakukan automasi untuk instalasi webserver, kita juga dapat melakukan automasi untuk setup web aplikasi menggunakan bash script. Berikut merupakan salah satu contoh script automasinya.

```
#!/bin/bash

jawaban="Y"

read -p "Apakah kamu yakin akan melakukan setup Aplikasi Web ? (Y/n) " pilih;

if [ $pilih == $jawaban ];
then
    echo "=====>"
    echo "Downloading Data"
    echo "=====>"
    cd
    wget https://github.com/sdcilisy/sosial-media/archive/master.zip
    echo "=====>"
    echo "Ekstrak File"
    echo "=====>"
    unzip master.zip
    echo "=====>"
    echo "Memindahkan data"
    echo "=====>"
    sudo rm /var/www/html/*
```

```

sudo rm -R /var/www/html/*
sudo mv sosial-media-master/* /var/www/html
echo "Setup selesai"
exit 0
else
    echo "Setup dibatalkan"
    exit 1
fi

```

Dengan begini semua data yang ada pada direktori /var/www/html akan di replace oleh aplikasi baru yang sudah kita download. Kita juga dapat menambahkan beberapa setup untuk membuat database mysql didalamnya.

Apabila pada saat program diatas dijalankan muncul log seperti dibawah ini maka biarkan, karena memang direktory sudah terhapus.

```

inflating: sosial-media-master/js/prism.js
inflating: sosial-media-master/js/twemoji.min.js
inflating: sosial-media-master/login.php
inflating: sosial-media-master/logout.php
inflating: sosial-media-master/register.php
inflating: sosial-media-master/timeline.php
=====
Memindahkan data
=====
rm: cannot remove '/var/www/html/css': Is a directory
rm: cannot remove '/var/www/html/img': Is a directory
rm: cannot remove '/var/www/html/js': Is a directory
Setup selesai

```

Hasil setup web aplikasi

Dengan begini kita sudah bisa menyiapkan environment server local dengan mudah, hanya dengan menggunakan satu script yang sudah dibuat kita bisa menjalankan beberapa layanan yang akan kita gunakan. Selain itu kita juga tidak perlu susah untuk melakukan setup web aplikasi karena kita bisa melakukannya secara otomatis.

3.16.3. Exercise

1. Buatlah sebuah script bash yang bisa menghapus semua layanan webserver secara sekaligus seperti layanan apache, php, dan database mysql.

3.17. Summary

Berikut adalah rangkuman poin-poin penting materi yang sudah dipelajari dari bab ini :

1. Server secara konsep hanyalah komputer biasa. Yang membedakan adalah server khusus ditujukan untuk memberikan layanan kepada pengguna.
2. Sistem Operasi Linux yang utuh sebenarnya disebut GNU/Linux. Yaitu gabungan kernel Linux dengan tools, software, dan komponen pelengkap dari GNU.
3. Linux adalah Sistem Operasi yang cocok untuk kebutuhan server.
4. Distro adalah sistem operasi GNU/Linux yang di modifikasi lagi sesuai kebutuhan tertentu. Distro yang terkenal dan banyak digunakan adalah CentOS dan Ubuntu.
5. Struktur direktori adalah ilmu untuk memahami fungsi dari masing-masing folder yang ada di Linux.
6. Command Line dasar merupakan perintah-perintah sederhana yang akan selalu digunakan selama mengoperasikan server Linux. Diantaranya command line navigasi, editing, manajemen file dan folder, dan bantuan.
7. Teknik Manipulasi teks dan file sangat penting untuk kebutuhan manajemen server linux dan scripting kedepannya. Tools yang bisa dipakai adalah sed, grep, cut, sort, dan uniq.
8. Teknik Archiving penting untuk keperluan menghemat space harddisk server. Biasa digunakan untuk keperluan backup.

9. Pencarian file dengan berbagai opsi membantu kita mendapatkan file yang dibutuhkan di tengah semrawutnya file dan folder yang ada di dalam server Linux.
10. Bagaimana membuat user, menghapus user, mengganti password user, meng-enable dan men-disable user, disebut ilmu Manajemen User.
11. Bagaimana membuat grup, menghapus grup, memasukkan user ke dalam grup dan mengeluarkannya, disebut ilmu Manajemen Grup.
12. Manajemen Paket berkaitan dengan cara instalasi aplikasi, cara uninstall aplikasi, cara mengecek aplikasi sudah terinstall atau belum, hingga cara upgrade versi aplikasi di Linux.
13. Tentang manajemen layanan : Manajemen Layanan berkaitan dengan cara mengecek status layanan apakah aktif atau tidak, cara menjalankan suatu layanan, menghentikannya, hingga mengatur apakah layanan tersebut diaktifkan saat booting atau tidak.
14. Pengetahuan pemrograman dasar sangat diperlukan untuk seorang devops, dimana pemrograman ini bisa diaplikasikan untuk sistem automasi dari environment yang akan kita buat nantinya