

---

# **Enterprise FOSS-GIS**

**Dany Laksono**

**Dec 09, 2020**



## CONTENTS

<b>1 Pengantar Linux Server, FOSS-GIS dan Web Server</b>	<b>3</b>
1.1 Pengantar Linux dan Server . . . . .	3
1.2 Pengaturan dan Administrasi Linux . . . . .	20
1.3 Linux sebagai Sistem Operasi Server . . . . .	35
1.4 Free and Open Source Software-GIS . . . . .	44
<b>2 Pengantar Basisdata Relational</b>	<b>55</b>
2.1 Basisdata Relational . . . . .	55
2.2 Administrasi Pengguna pada Basisdata . . . . .	76
2.3 Basisdata Spasial dengan PostGIS . . . . .	83
<b>3 Pengantar Interoperabilitas Data Spasial</b>	<b>97</b>
3.1 Interoperabilitas dan Standar OGC . . . . .	97
3.2 Publikasi Data Spasial dengan Geoserver . . . . .	109
3.3 Konsumsi Data dengan Standar OGC . . . . .	119



## Tentang Panduan ini

Panduan ini dibuat untuk memberikan pengantar dalam penggunaan perangkat lunak OpenSource untuk Enterprise GIS. Panduan ini memberikan dasar yang diperlukan untuk mengenal dan menggunakan beberapa aplikasi GIS yang bersifat bebas dan terbuka, dengan Linux sebagai sistem operasi utamanya.

Panduan ini dapat diberikan sebagai pegangan pada pelatihan maupun sebagai bahan untuk *self-paced learning*. Modul ini dilengkapi berbagai latihan yang diberikan untuk mendukung pemahaman tiap materi yang disampaikan.

## Jadwal Pelatihan

Apabila diberikan pada sesi pelatihan tentang Free and OpenSource GIS, pembagian jadwal dari materi ini dapat dilakukan sebagai berikut:

### Hari 1: Linux Server dan FOSS4G

Waktu	Materi Pelatihan
09.00 - 10.00	Pendahuluan, Instalasi Linux VM
10.00 - 10.15	<i>Break</i>
10.15 - 12.00	Arsitektur dan Administrasi Sistem Ubuntu Linux
12.00 - 13.00	<i>ISHOMA</i>
13.00 - 15.00	Linux untuk Server Web
15.00 - 15.15	<i>Break</i>
15.15 - 16.00	Pengantar Free and OpenSource GIS (FOSS-GIS)

### Hari 2: Relational Spatial Database

Waktu	Materi Pelatihan
09.00 - 10.00	Konsep Relational Database Pengaturan dan Instalasi Basisdata
10.00 - 10.15	<i>Break</i>
10.15 - 12.00	Adminsitrasni Enterprise GIS dengan PostGIS: User, Roles, Priviledges
12.00 - 13.00	<i>ISHOMA</i>
13.00 - 15.00	Bekerja dengan PostGIS
15.00 - 15.15	<i>Break</i>
15.15 - 16.00	SQL vs NoSQL Database

### Hari 3: Standar OGC untuk Interoperabilitas Data Spasial

Waktu	Materi Pelatihan
09.00 - 10.00	Interoperabilitas dan Standar OGC Instalasi dan Pengaturan Geoserver
10.00 - 10.15	<i>Break</i>
10.15 - 12.00	Publikasi data dengan standar OGC pada Geoserver
12.00 - 13.00	<i>ISHOMA</i>
13.00 - 15.00	Pengaturan dan konsumsi data melalui layanan OGC
15.00 - 15.15	<i>Break</i>
15.15 - 16.00	Arsitektur Enterprise GIS dengan FOSS4G

## Lisensi Dokumen

Dokumen ini dan latihan yang ada di dalamnya disusun di bawah panduan CC-BY-SA versi 4.0.

Dokumen ini ditulis oleh [Dany Laksono](#) untuk Pusat Data dan Teknologi Informasi (Pusdatin) Kementerian PUPR tahun 2020

**Lisensi Perangkat Lunak**

Seluruh perangkat lunak yang digunakan maupun diacu pada modul ini memiliki lisensi bebas dan terbuka (Free and Open Source). Lisensi masing-masing perangkat lunak dapat dilihat pada penjelasan lebih rinci mengenai software tersebut

## PENGANTAR LINUX SERVER, FOSS-GIS DAN WEB SERVER

Linux, atau GNU/Linux, mengacu pada sistem operasi komputer Unix-like yang menggunakan kernel Linux. Linux merupakan contoh paling menonjol dari pengembangan open source dan perangkat lunak bebas serta perangkat lunak yang dibuat pengguna; kode sumber yang mendasarinya tersedia bagi siapa saja untuk digunakan, dimodifikasi, dan didistribusikan kembali secara bebas.

Bagian ini merupakan bagian pertama dari panduan ini. Sesi ini membahas mengenai Linux dan cara kerjanya, serta bagaimana Linux digunakan sebagai sistem operasi untuk menunjang penggunaan *Free and Open Source Software (FOSS)*, khususnya untuk menunjang analisis dan manajemen data Geospasial.

Berikut adalah materi yang dibahas pada bagian ini:

*Pengantar mengenai Linux* : Sejarah dan perkembangan Linux, beberapa Distro Linux yang terkenal, serta instalasi dan penggunaan Ubuntu Linux, yang akan digunakan pada seluruh dokumen latihan ini.

*Arsitektur dan Administrasi sistem Linux* : Beberapa konsep tentang arsitektur Linux sebagai sebuah sistem operasi, serta bagaimana pengaturan Linux pada sebuah lingkungan server

*Linux untuk Webserver* : Latihan mengenai pengaturan Linux untuk publikasi halaman web dengan menggunakan Apache/httpd dan Tomcat. Akan disampaikan juga mengenai penggunaan Virtual Host dan konsep proxy pada Apache untuk keperluan publikasi website

*Pengantar FOSS-GIS* : Mengenal jenis-jenis lisensi perangkat lunak serta beberapa perangkat FOSS-GIS yang populer.

### 1.1 Pengantar Linux dan Server

Bagian ini akan mengulas mengenai sejarah perkembangan Linux dan bagaimana Linux menjadi salah satu sistem operasi yang paling banyak digunakan dan paling pesat perkembangannya di dunia: **Lisensi yang bebas dan terbuka**.

#### 1.1.1 Sejarah dan Perkembangan Sistem Operasi Linux

Pada tanggal 25 Agustus 1991, Linus Benedict Torvalds yang saat itu adalah seorang mahasiswa di University of Helsinki menulis sebuah email yang kemudian menjadi *ikon* di dunia pengembangan perangkat lunak:

Hello everybody out there using minix -

Email sederhana ini, dengan sedikit penjelasan mengenai sistem operasi baru yang sedang ia kembangkan, kemudian menjadi awal dari adopsi besar-besaran Linux sampai menjadi sistem operasi yang sangat populer saat ini.

Meskipun diakhiri dengan nada pesimis, Kernel Linux yang dikembangkannya menjadi salah satu *milestone* dalam sejarah komputer. Dari kernel sederhana yang menurutnya ‘hanya dapat digunakan pada mesin AT-386’, Linux kemudian berkembang menjadi sistem operasi yang dapat digunakan pada berbagai platform: super komputer, server, *embedded device*, dan berbagai platform lain.

```
From: torv...@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Keywords: 386, preferences
Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki
Lines: 20
```

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torv...@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-(.

Fig. 1.1: Email bersejarah Linus pada forum *comp.os.minix*

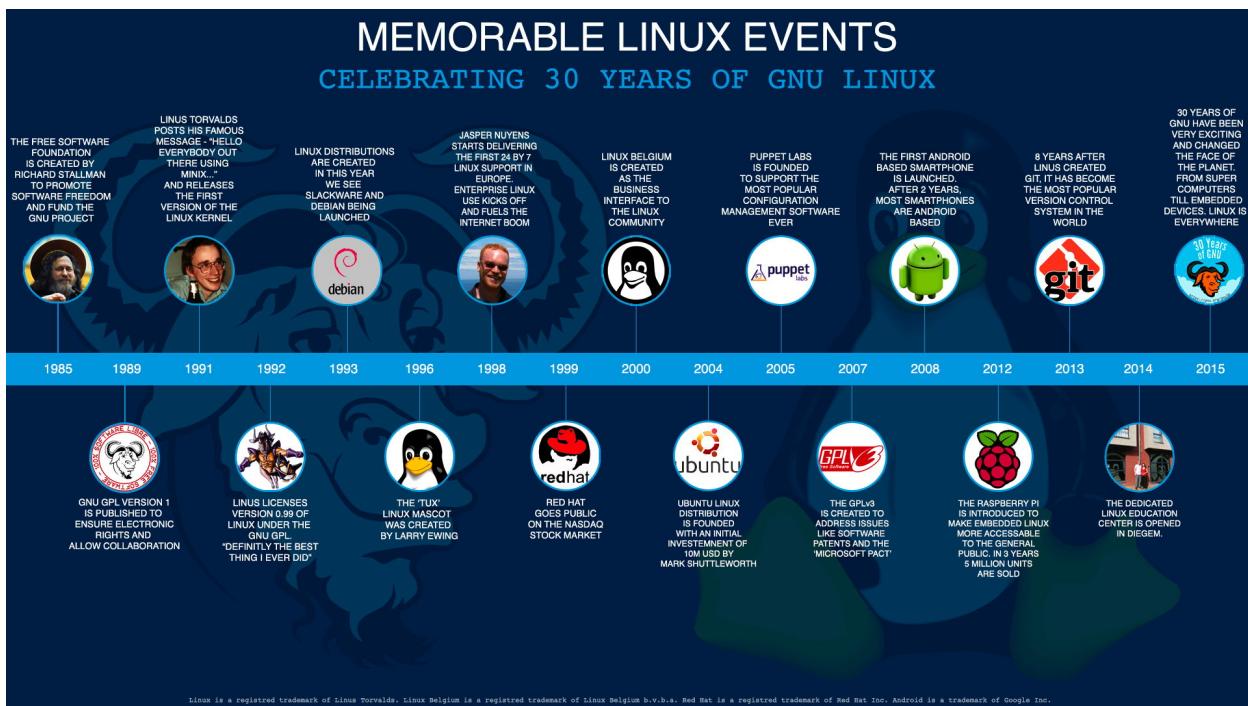


Fig. 1.2: Sejarah perkembangan Linux

Lisensi yang terbuka ([GNU General Public License](#) yang dikembangkan oleh [Richard Stallman](#)) memungkinkan Linux untuk dikembangkan secara bebas: tiap orang dapat membuat versi Linux mereka sendiri, membuat *copy*, melakukan modifikasi pada kode sumber, serta membagikan modifikasi tersebut dengan satu syarat, yaitu setiap perubahan tersebut dibuat dalam lisensi publik yang terbuka. Penggunaan lisensi ini penting dalam perkembangan Linux kemudian, sehingga bermunculan berbagai macam distribusi Linux yang menyesuaikan berbagai platform dan kebutuhan penggunanya.

### 1.1.2 Distribusi Linux

*Linux Distribution* (disingkat *Distro*) merupakan sistem operasi yang menggunakan Kernel Linux sebagai basisnya, dengan berbagai komponen lain yang berbeda-beda, seperti *Desktop Environment*, *package management system*, berbagai pustaka tambahan, dan seterusnya.

Meskipun menggunakan kernel Linux yang sama, dengan sifatnya yang bebas dan terbuka, berbagai distro ini kemudian memiliki komponen yang beragam sehingga muncul berbagai variasi [distro Linux yang tersedia saat ini](#). Saat dokumen ini ditulis, tidak kurang terdapat 600 distro Linux yang berbeda, dengan 500 diantaranya masih dalam tahap *development aktif*.

Beberapa distro Linux yang populer antara lain<sup>1</sup>:

<sup>1</sup> <https://www.techradar.com/best/best-linux-distros>

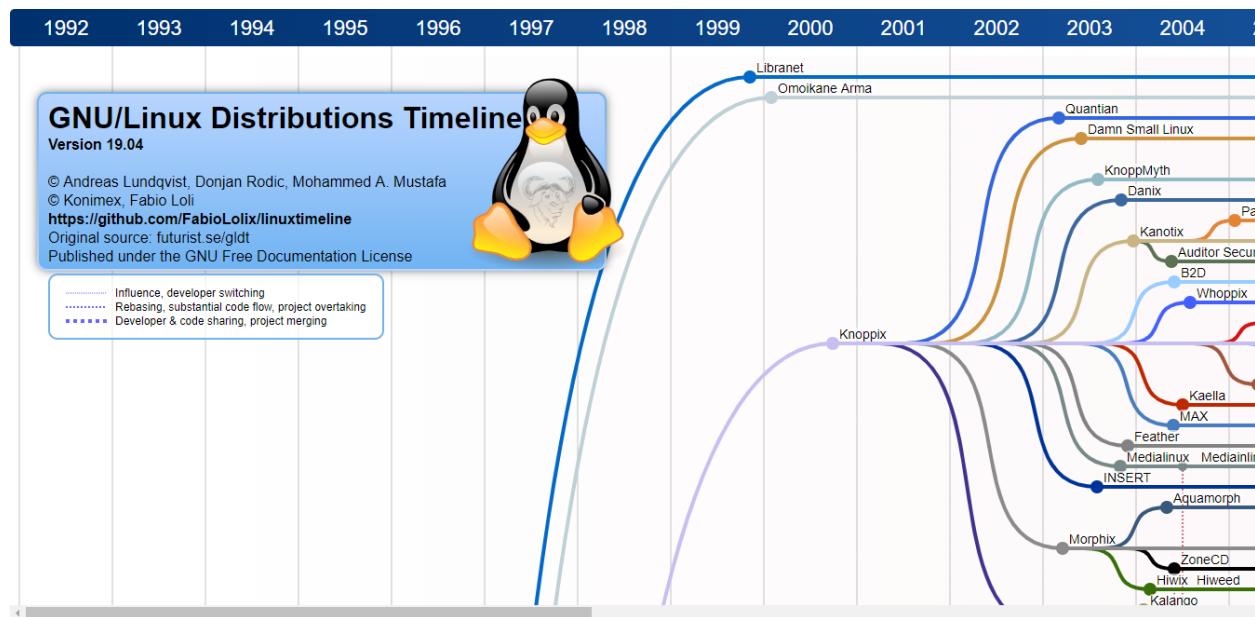
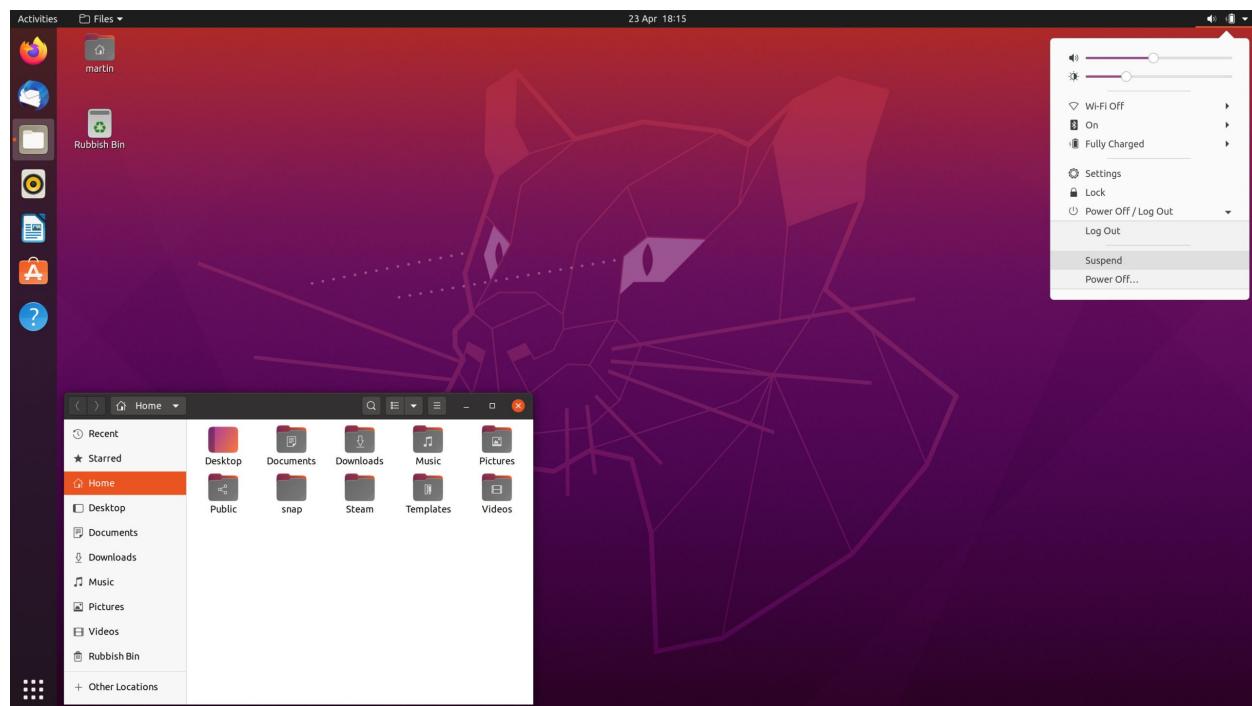


Fig. 1.3: Timeline Distribusi Linux. Klik [di sini](#) untuk melihat versi interaktifnya

### Ubuntu

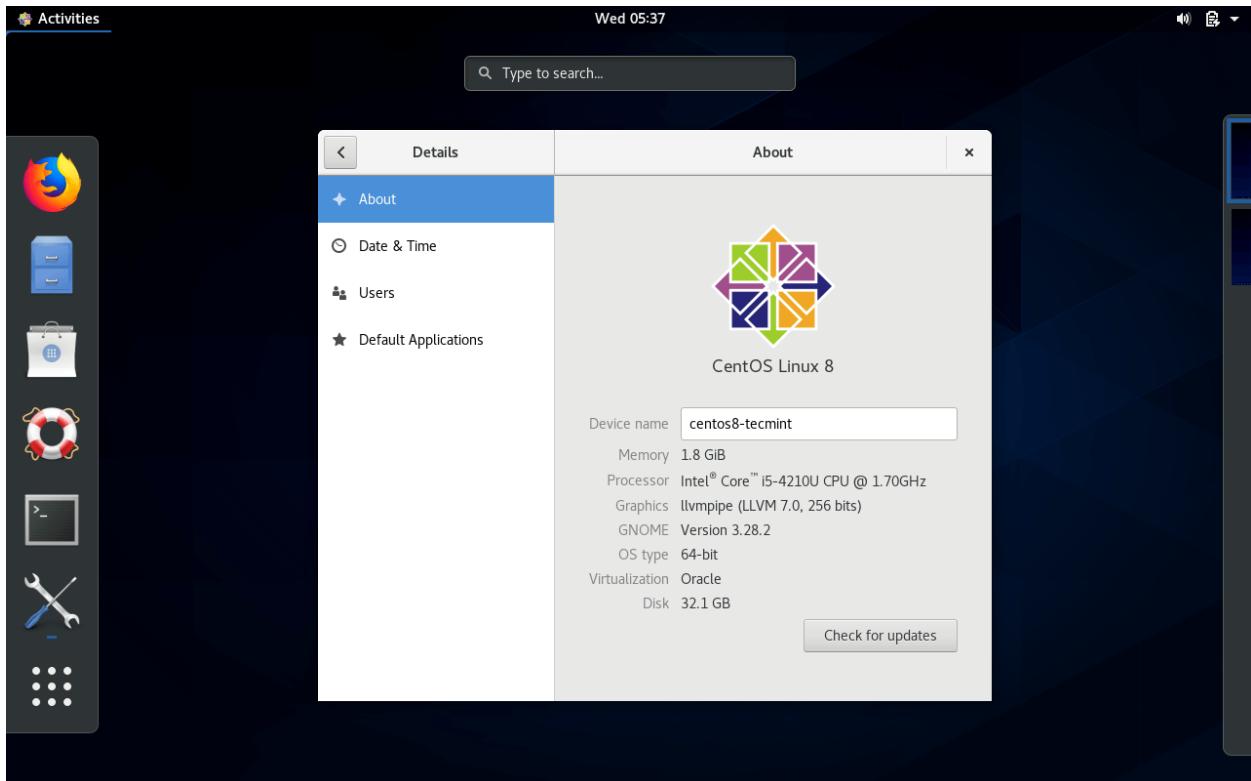
Ubuntu merupakan salah satu sistem operasi Linux yang paling populer saat ini. Ubuntu dilengkapi dengan package management (*Advanced Package Tool* atau *apt*) dan *desktop environment* berbasis GNOME yang mendukung kemudahan penggunaannya untuk pemula. Ubuntu didukung dengan stabilitas dan keamanan yang mumpuni dengan versi LTS (*Long Term Support*) yang dirilis secara berkala oleh Canonical sebagai developer sistem operasi ini. Atas alasan itulah Ubuntu juga menjadi salah satu sistem operasi yang banyak dijumpai pada server. Ini juga didukung oleh adanya *community support* yang besar dan aktif, serta dukungan berbagai perangkat lunak pihak ketiga pada sistemnya.



Saat ini, versi Long Term Support untuk Ubuntu adalah 20.04 (Focal Fossa). Selain versi ini, versi LTS sebelumnya adalah 18.04 (Bionic Beaver) dan 16.04 (Xenial Xerus). Untuk penggunaan pada modul ini, akan digunakan Ubuntu versi LTS 18.04, untuk mengantisipasi adanya beberapa package yang belum didukung pada versi LTS terbaru. Pada praktiknya, penggunaan versi LTS yang lebih baru akan lebih baik, karena adanya dukungan keamanan dan *patch* yang diberikan pada tiap versi terbarunya.

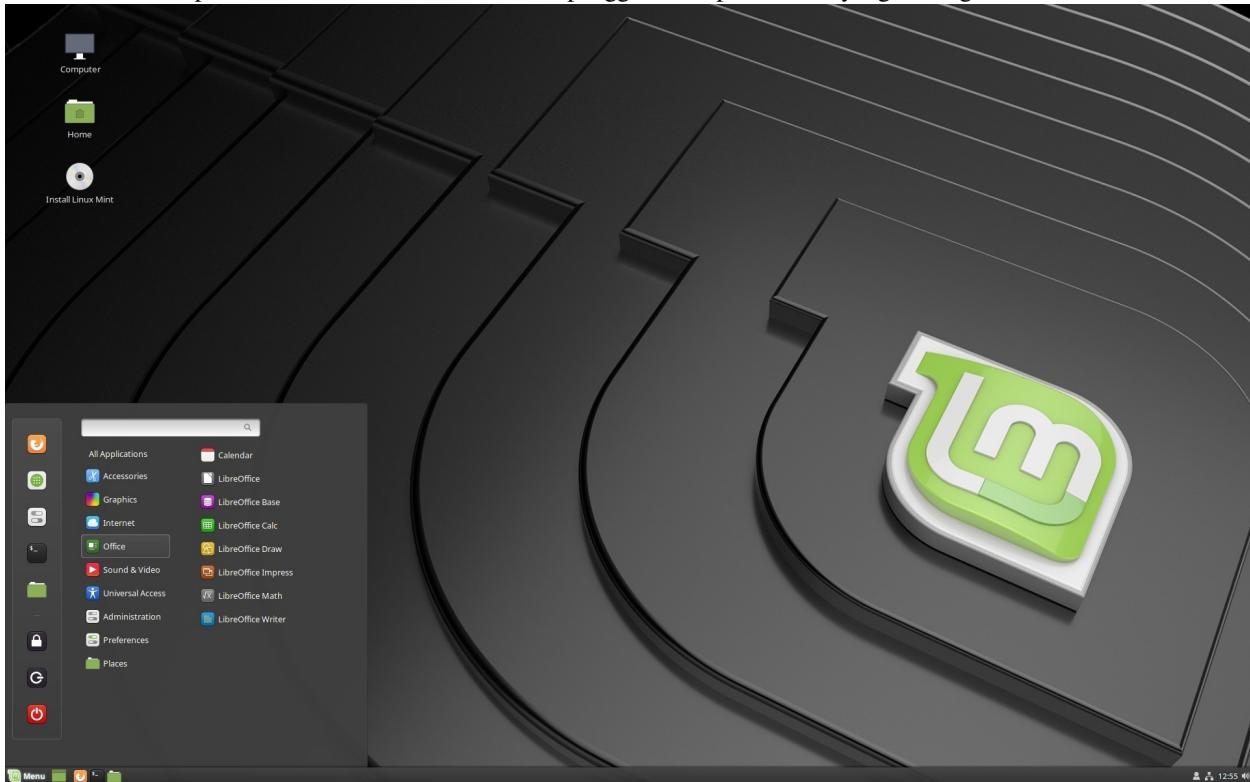
## CentOS

CentOS merupakan pengembangan dari RedHat Linux yang banyak digunakan untuk komputer pada server maupun komputasi awan. Sebagai sebuah sistem operasi desktop, CentOS juga dilengkapi dengan antarmuka berbasis GNOME, sama seperti Ubuntu. CentOS dikenal karena menyediakan fungsi Enterprise sebagai sebuah sistem operasi pada server tanpa perlu mengeluarkan biaya sepeser pun.



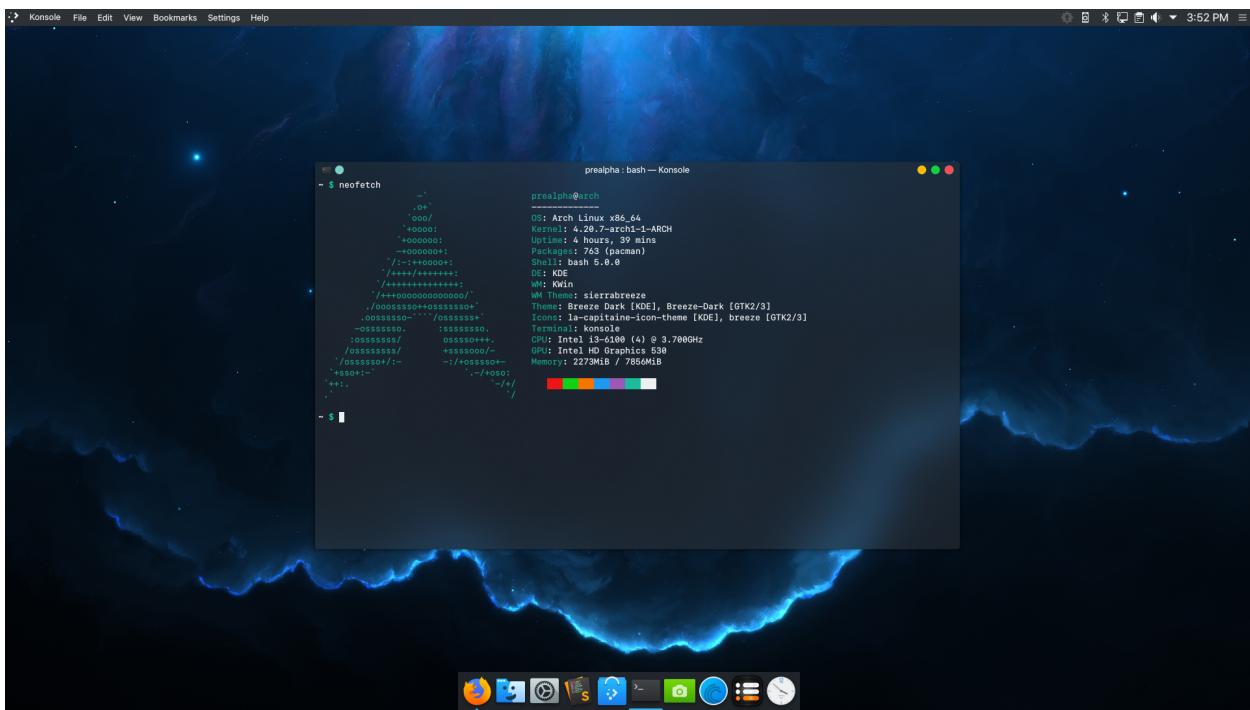
### Linuxmint

Bagi pengguna yang baru saja ingin migrasi dari Windows dan mencoba antarmuka Linux Desktop, [Linuxmint](#) merupakan pilihan yang sesuai. Desain antarmukanya dibuat sedemikian rupa untuk memudahkan pengguna pemula yang ingin mencoba Linux.



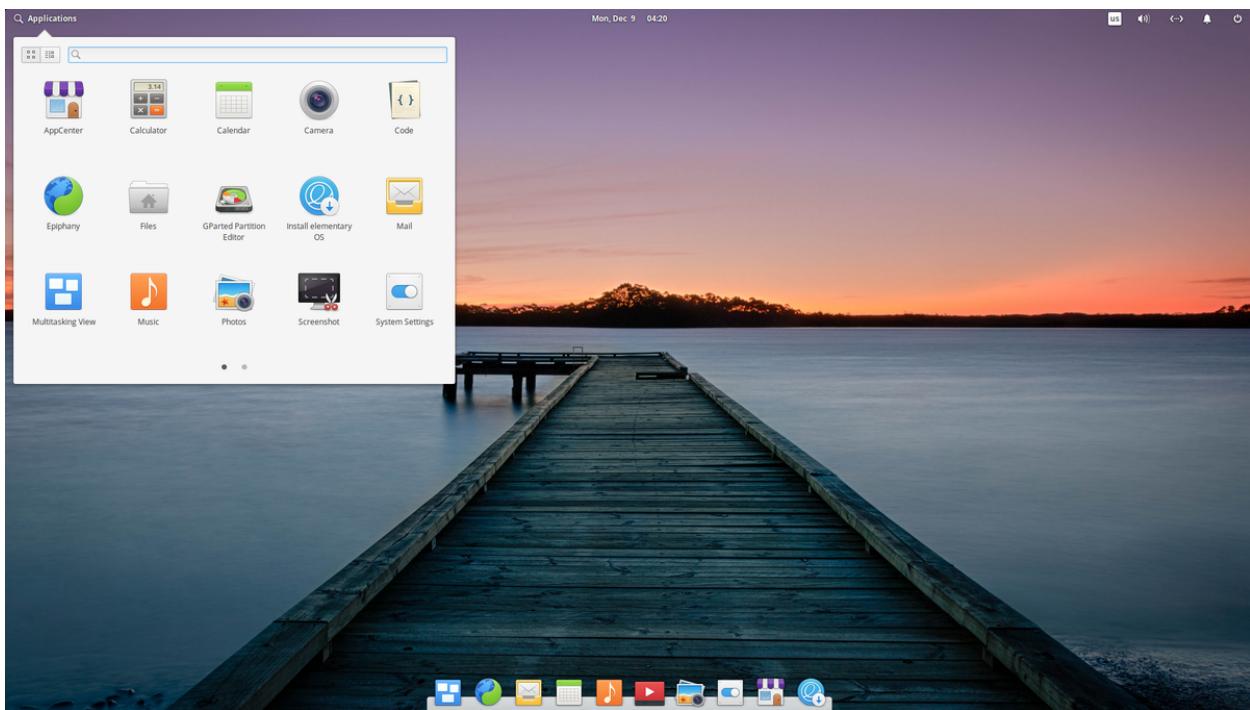
### ArchLinux

Untuk pengguna yang menginginkan kebebasan kustomisasi tanpa mengorbankan antarmuka berbasis desktop, [ArchLinux](#) dapat menjadi pilihan yang sesuai. ArchLinux memiliki konsep ringan (*lightweight*), dengan komponen desktop environment yang kaya pilihan: Cinnamon, Enlightenment, GNOME, KDE, LXDE, MATE, Xfce.



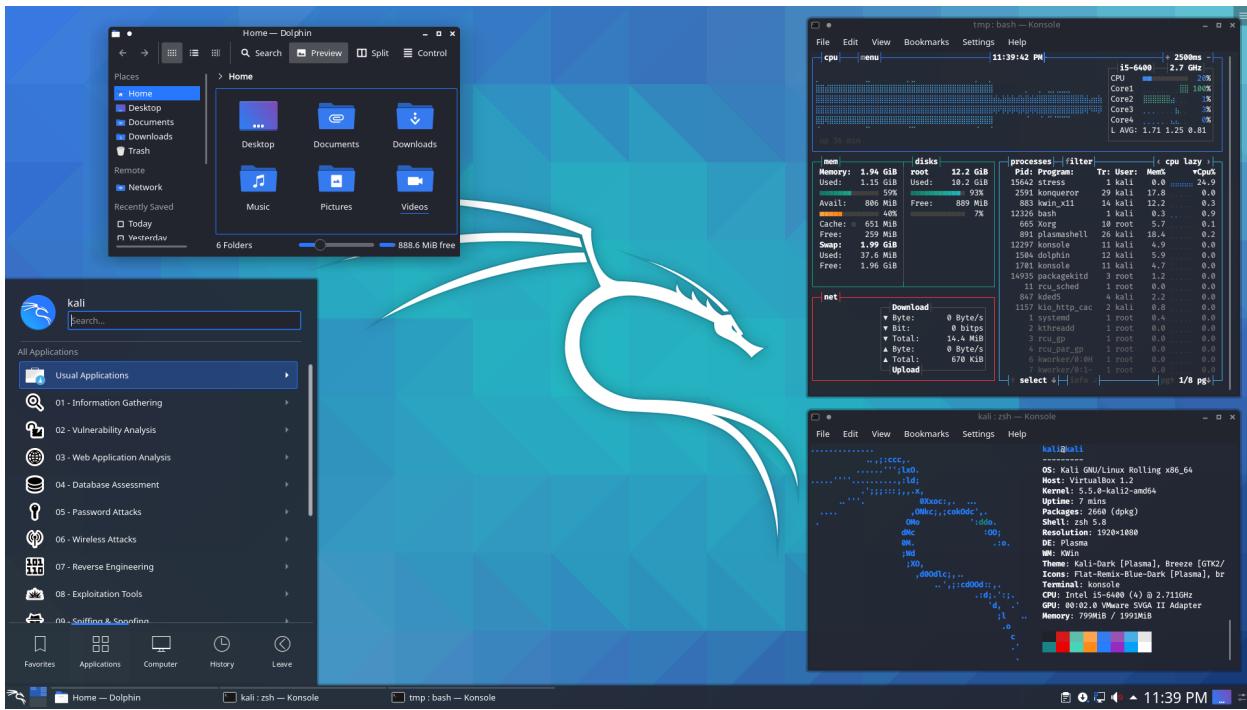
## Elementary OS

Selain LinuxMint, ElementaryOS dikenal sebagai sistem operasi dengan tampilan yang paling memukau. Sistem Operasi ini didesain dengan tampilan yang, bagi pengguna Mac maupun Windows, akan terasa familiar. ElementaryOS dikembangkan dari sistem operasi Ubuntu LTS, dan membanggakan kebijakan penjagaan privasi yang kuat untuk penggunanya.



### Kali Linux

Barangkali **Kali Linux** merupakan sistem operasi yang paling dikenal di kalangan Hacker. Kali Linux merupakan sistem operasi Linux yang dirancang khusus untuk keperluan *penetration testing* atau pengujian keamanan suatu sistem atau perangkat lunak. Instalasi standar Kali Linux dilengkapi dengan berbagai perangkat yang dapat digunakan untuk melakukan *pentesting*, seperti *aircrack-ng* dan *Wireshark*. Kali Linux bahkan juga dilengkapi dengan PowerShell bagi pengguna yang menginginkan lingkungan scripting tersebut pada Linux.



Selain berbagai distro yang disebutkan di atas, terdapat ratusan distro lain yang digunakan untuk berbagai keperluan. Tiap distro memiliki keunggulan masing-masing, sehingga daftar ini terlalu pendek untuk sebuah ulasan lengkap. Situs ini menyediakan perbandingan untuk tiap distro yang terdaftar berikut ulasan singkat mengenai masing-masing distro.

### 1.1.3 Windows Subsystem for Linux

Dengan perkembangan pesat Linux pada berbagai platform dan dukungan komunitas yang sangat besar, Linux menjadi sistem operasi yang banyak digunakan oleh pengembang aplikasi, khususnya pada penggunaan server. Linux banyak digunakan karena selain mudah, keamanan dan stabilitas sistem yang diberikan juga menjadi poin penting dalam penggunaannya untuk keperluan server. Di sisi lain, Windows sebagai sistem operasi yang paling populer saat ini juga banyak digunakan untuk keperluan sehari-hari. Untuk itu, pengembangan perangkat lunak seringkali melibatkan kedua sistem operasi ini sekaligus (juga MacOS, bagi pengguna setianya). Pada bagian ini akan dibahas mengenai WSL versi 2 sebagai jembatan untuk Windows dan Linux yang akan sangat membantu pengembang dalam menggunakan Linux pada Sistem Operasi Windows.

## Tentang Windows Subsystem for Linux (WSL)

Pada tahun 2017, Microsoft sebagai pesaing terbesar Linux membuat pengumuman yang mengejutkan: Windows 10 sebagai sistem operasi utama Windows pada saat ini akan memiliki dukungan untuk menjalankan Linux Kernel secara *native*. Sebelumnya, bagi *developer* yang ingin menggunakan Windows dan Linux secara bersamaan harus menggunakan perangkat virtualisasi (misalnya Vagrant) atau melakukan instalasi dual-boot agar kedua sistem dapat berjalan. Penggunaan WSL memungkinkan Linux untuk dapat dijalankan dengan ringan pada sistem operasi Windows.

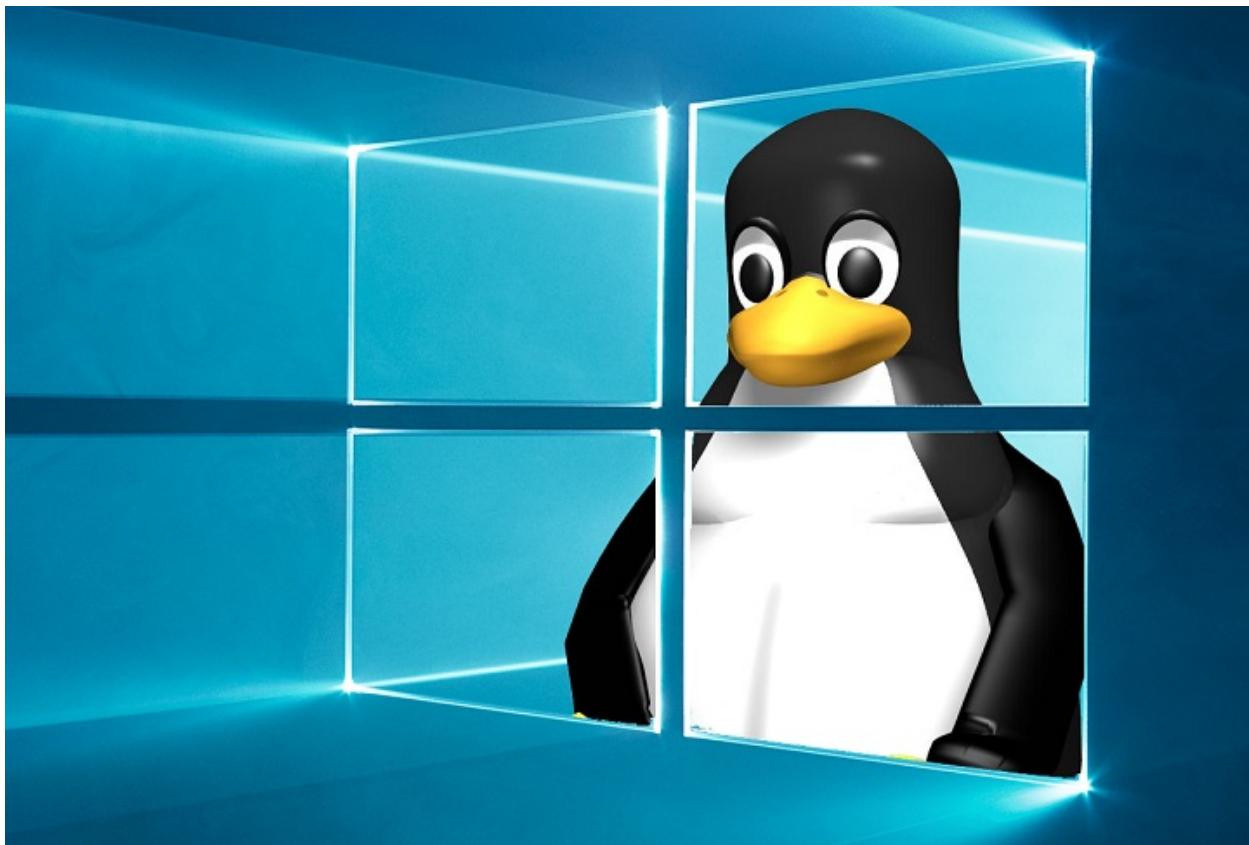


Fig. 1.4: WSL memungkinkan Linux dijalankan pada Windows

WSL2 (Windows Subsystem for Linux 2) merupakan integrasi mesin berbasis Linux pada Windows yang memungkinkan mesin Linux untuk dapat dijalankan pada Windows dengan performa maksimal. Indeks angka “2” menunjukkan bahwa virtualisasi ini adalah pengembangan lebih jauh dari mesin WSL yang pertama kali diusulkan oleh Microsoft pada tahun 2017. Perbedaan mencolok terlihat karena pada WSL2, mesin virtual yang dipasang menggunakan kernel linux yang sebenarnya. Artinya, performa yang dapat diperoleh dari mesin virtual ini juga sebaik mesin Linux dengan kekuatan penuh. Selain itu, dengan menggunakan [Distro yang didukung oleh WSL2](#), akses Windows pada mesin (dan sebaliknya) dapat dilakukan secara hampir *seamless*.

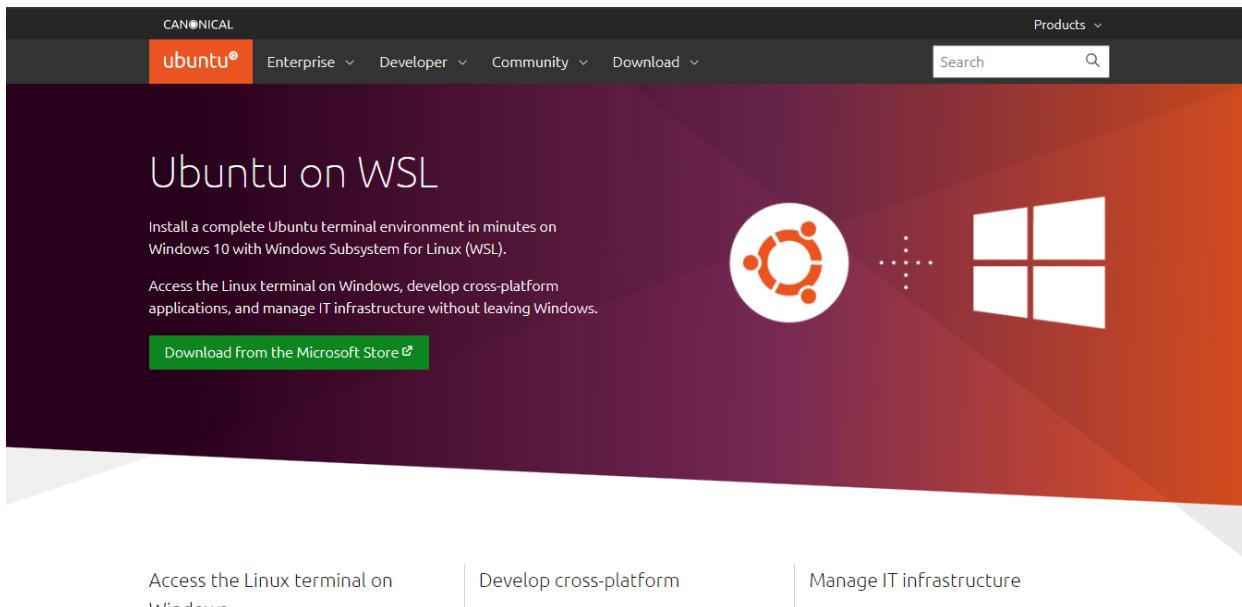


Fig. 1.5: Ubuntu yang khusus dirancang untuk WSL

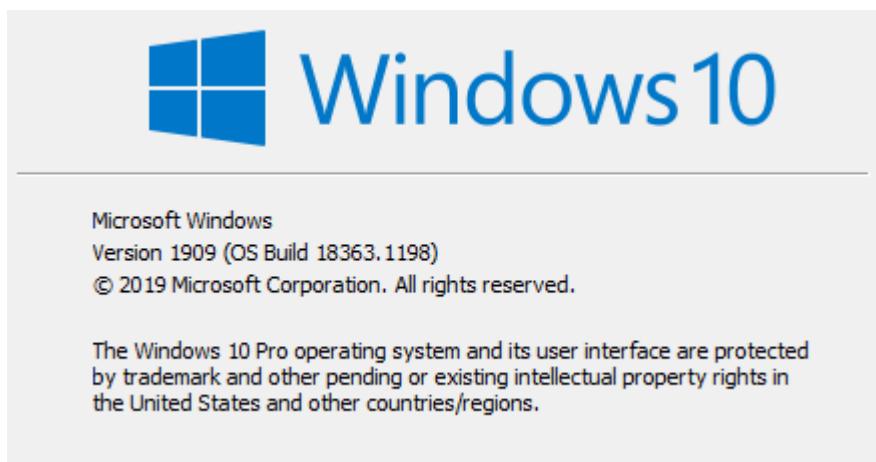
### Latihan: Instalasi WSL2 Pada Windows 10

Untuk instalasi WSL2 ini, berikut adalah persyaratan yang harus dipenuhi:

- Windows 10 dengan versi update terbaru: Windows 10 May 2020 (2004), Windows 10 May 2019 (1903), or Windows 10 November 2019 (1909). Windows dengan versi update lama (1903 atau 1909) tetap dapat menginstall WSL2 dengan terlebih dahulu melakukan instalasi Windows Update [berikut ini](#)
- Komputer atau laptop yang memiliki dukungan **Hyper-V Virtualization**. Periksa link berikut untuk mengetahui apakah Komputer atau laptop Anda memiliki dukungan Hyper-V.

Untuk melihat versi build Windows yang Anda gunakan, lakukan langkah berikut:

- Buka Windows Run (Win+R)
- Ketikkan 'winver'
- Pada jendela yang muncul, akan terlihat versi build Windows yang sedang berjalan



Apabila versi build Windows Anda di bawah 18362, maka terlebih dulu lakukan update Windows menggunakan [Update Assistant](#).

Jika Hyper-V belum diaktifkan pada firmware, maka opsi ini perlu diaktifkan terlebih dulu pada BIOS. Panduan ini menyediakan langkah yang diperlukan untuk mengaktifkan Hyper-V pada BIOS dan pada Windows 10.

#### Catatan:

Panduan ini dibuat dengan mengikuti panduan resmi di:

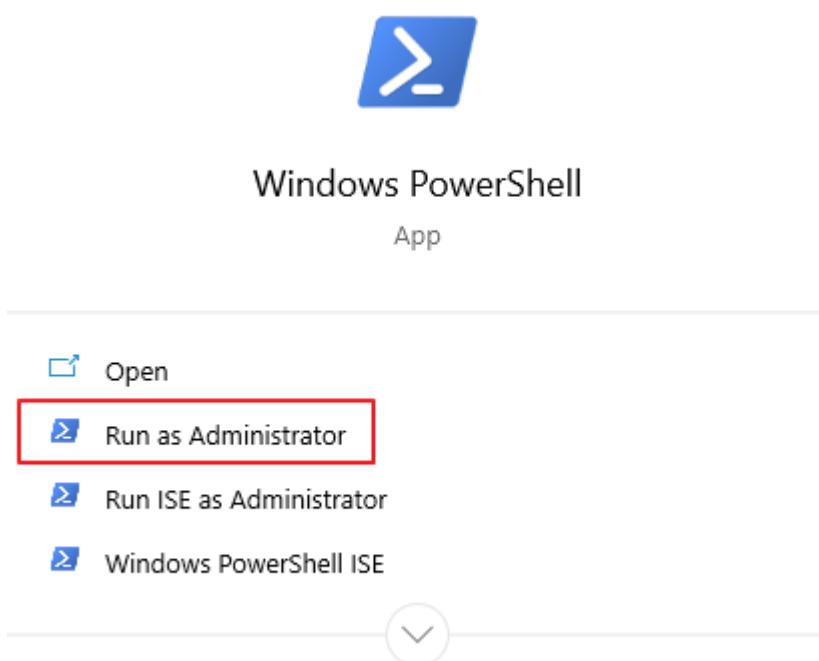
<https://docs.microsoft.com/en-us/windows/wsl/install-win10>

Periksa tautan tersebut untuk melihat perubahan terbaru terkait dengan instalasi WSL

Berikut adalah petunjuk untuk melakukan instalasi WSL2 pada Sistem Operasi Windows 10:

1. *Mengaktifkan WSL melalui PowerShell*

Buka Windows PowerShell sebagai Administrator.



Ketikkan perintah berikut pada command prompt yang tersedia:

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /  
-all /norestart
```

Akan muncul pemberitahuan apabila proses telah berhasil:

```
Deployment Image Servicing and Management tool  
Version: 10.0.18362.1139  
  
Image Version: 10.0.18363.1198  
  
Enabling feature(s)  
[=====100.0%=====  
The operation completed successfully.
```

### 2. Update ke WSL versi 2

Pastikan versi build Windows yang digunakan sesuai untuk persyaratan WSL2 seperti di atas. Periksa [link](#) berikut untuk melihat lebih detil mengenai bagian ini.

### 3. Aktifkan Virtual Machine Platform

Pada jendela PowerShell yang sama (sebagai Administrator), ketikkan perintah berikut:

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /  
-norestart
```

Setelah perintah selesai dijalankan, lakukan **restart pada Windows**. Ini akan mengupdate WSL pada system menjadi versi ke-2

**Caution:** Restart Windows penting dalam hal ini agar sistem WSL terupdate pada versi terbaru.

### 4. Unduh Linux kernel untuk WSL2

Terlebih dahulu unduh dan install Linux Kernel untuk WSL2 dari [Link Berikut](#).

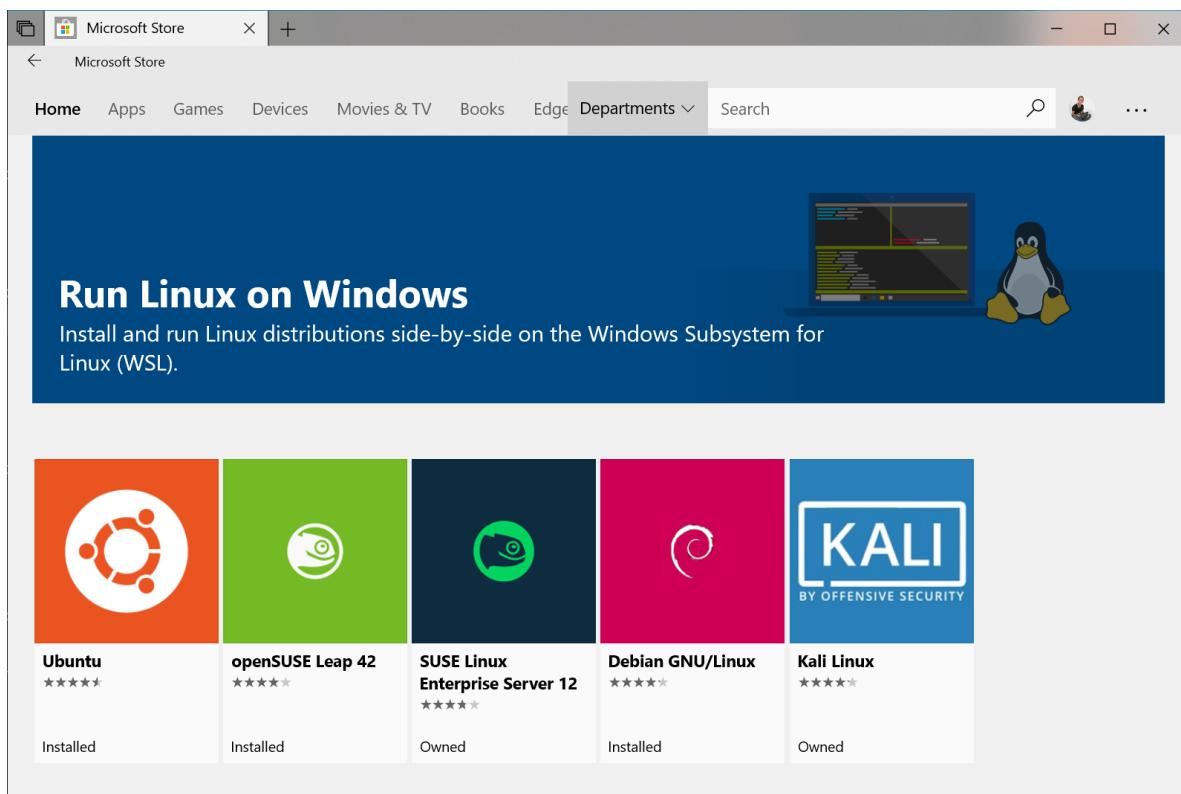
Setelah instalasi berhasil, masukkan perintah berikut ke dalam PowerShell yang sama seperti sebelumnya:

```
wsl --set-default-version 2
```

Ini akan membuat versi WSL yang telah terinstall menggunakan versi kedua.

### 5. Melakukan Instalasi Distro Linux

Pada [Microsoft Store](#) telah tersedia beberapa distro Linux yang dapat digunakan pada WSL. Buka Microsoft Store, kemudian lakukan instalasi untuk distro yang dipilih. Untuk latihan ini, gunakan Ubuntu, khususnya versi [Ubuntu 18.04](#).



Klik Get untuk melakukan instalasi Ubuntu pada WSL2

Store

Home Apps Games Music Movies & TV Books Microsoft

Search

Ubuntu  
Canonical Group Limited  
★★★★★ 22  
Free

Get Share

ESRB Everyone

Screenshots

Description

Ubuntu on Windows allows one to use Ubuntu Terminal and run Ubuntu command line utilities including bash, ssh, git, apt and many more.

To use this feature, one first needs to use "Turn Windows features on or off" and select "Windows Subsystem for Linux", click OK, reboot, and use this app.

The above step can also be performed using Administrator PowerShell prompt: Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux

More

Available on

PC

What's new in this version

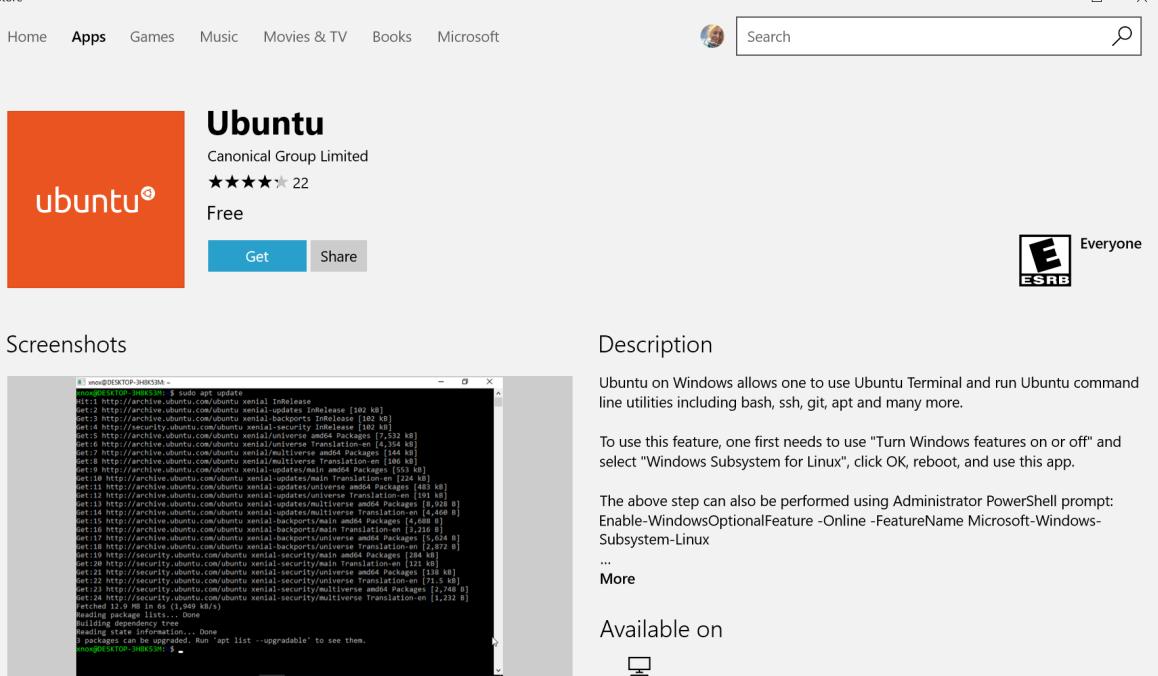
20170619.1 build of Ubuntu 16.04 LTS

Features

Ubuntu

bash

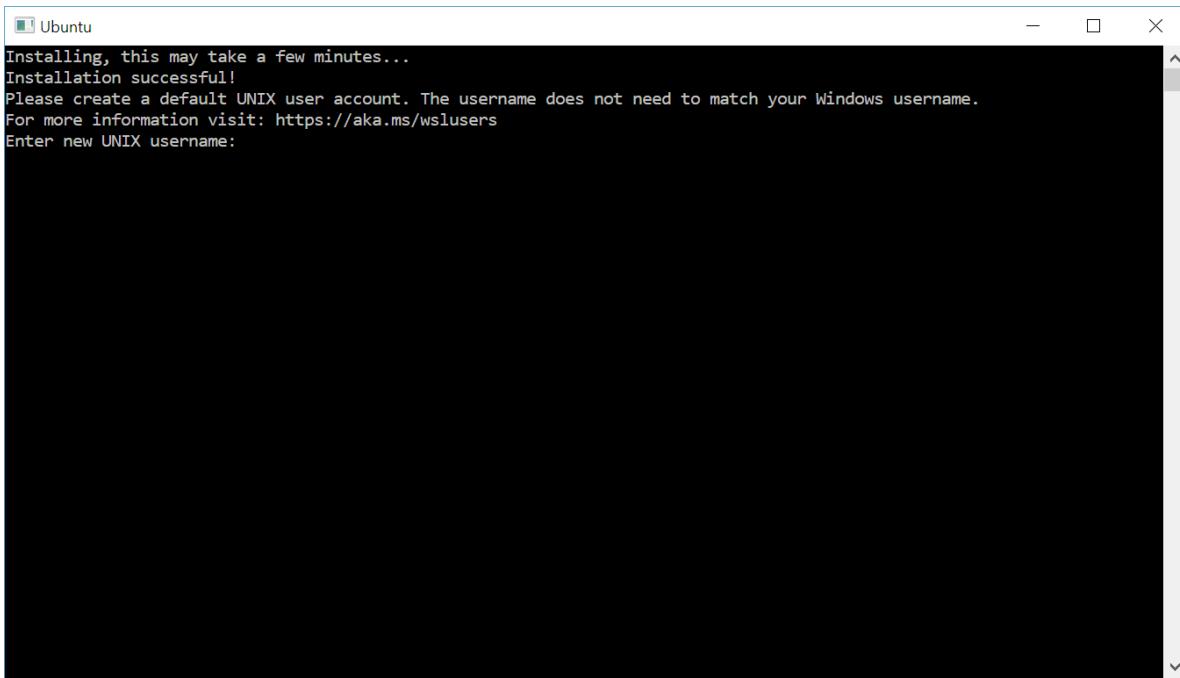
ssh



### 6. Memulai WSL

Setelah instalasi selesai, Ubuntu dapat langsung digunakan pada Windows. Untuk memulai WSL, buka icon menu bergambar logo Ubuntu pada Start Menu.

Pada saat pertama kali dibuka, mesin akan meminta **username** dan **password**. Catat username dan password yang dimasukkan karena ini akan menjadi akun *sudoer* pada mesin Linux yang dibuat.



Selanjutnya, Ubuntu siap digunakan

A screenshot of a Windows terminal window titled "danylaksono@DANYLAKSONO". The window shows the following text:

```
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 4.19.128-microsoft-standard x86_64)

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

System information as of Mon Nov 30 04:42:47 +07 2020

System load: 0.15      Processes:          8
Usage of /: 1.5% of 250.98GB  Users logged in:    0
Memory usage: 1%           IPv4 address for eth0: 192.168.244.140
Swap usage:  0%

6 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

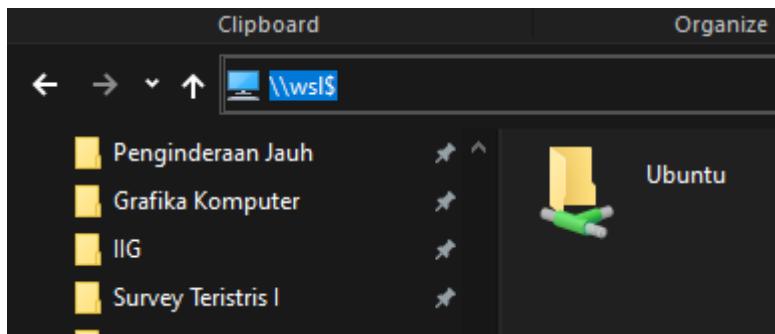
This message is shown once a day. To disable it please create the
/home/danylaksono/.hushlogin file.
danylaksono@DANYLAKSONO:~$
```

The window has a standard Windows title bar with minimize, maximize, and close buttons.

## 7. Integrasi WSL dan Windows

Telah disampaikan sebelumnya bahwa integrasi Linux dan Windows menggunakan WSL2 bersifat hampir seamless. Artinya, kedua sistem dapat saling berkomunikasi tanpa memerlukan media perantara yang rumit.

Untuk mengakses file WSL dari windows, buka Explorer, kemudian ketikkan pada bagian directory list: \\wsl\$.



Maka Windows akan membuka drive pada Linux sebagaimana sebuah folder biasa pada Explorer.

	Name	Date modified	Type	Size
	boot	8/5/2020 4:47 AM	File folder	
	dev	11/30/2020 4:42 AM	File folder	
	etc	11/30/2020 4:42 AM	File folder	
	home	9/29/2020 5:17 PM	File folder	
	lost+found	4/10/2019 11:35 PM	File folder	
	media	8/5/2020 4:39 AM	File folder	
	mnt	9/29/2020 5:17 PM	File folder	
	opt	8/5/2020 4:39 AM	File folder	
	proc	11/30/2020 4:42 AM	File folder	
	root	8/5/2020 4:42 AM	File folder	
	run	11/30/2020 4:42 AM	File folder	
	snap	7/10/2020 8:59 PM	File folder	
	srv	8/5/2020 4:39 AM	File folder	
	sys	11/30/2020 4:42 AM	File folder	
	tmp	11/28/2020 7:54 PM	File folder	
	usr	8/5/2020 4:40 AM	File folder	
	var	8/5/2020 4:42 AM	File folder	
	bin	8/5/2020 4:39 AM	File	1 KB
	init	9/29/2020 4:32 PM	File	618 KB
	lib	8/5/2020 4:39 AM	File	1 KB
	lib32	8/5/2020 4:39 AM	File	1 KB

Berkas yang disimpan pada ‘folder’ ini akan langsung terintegrasi dengan Linux tanpa perlu melakukan *rebooting*.

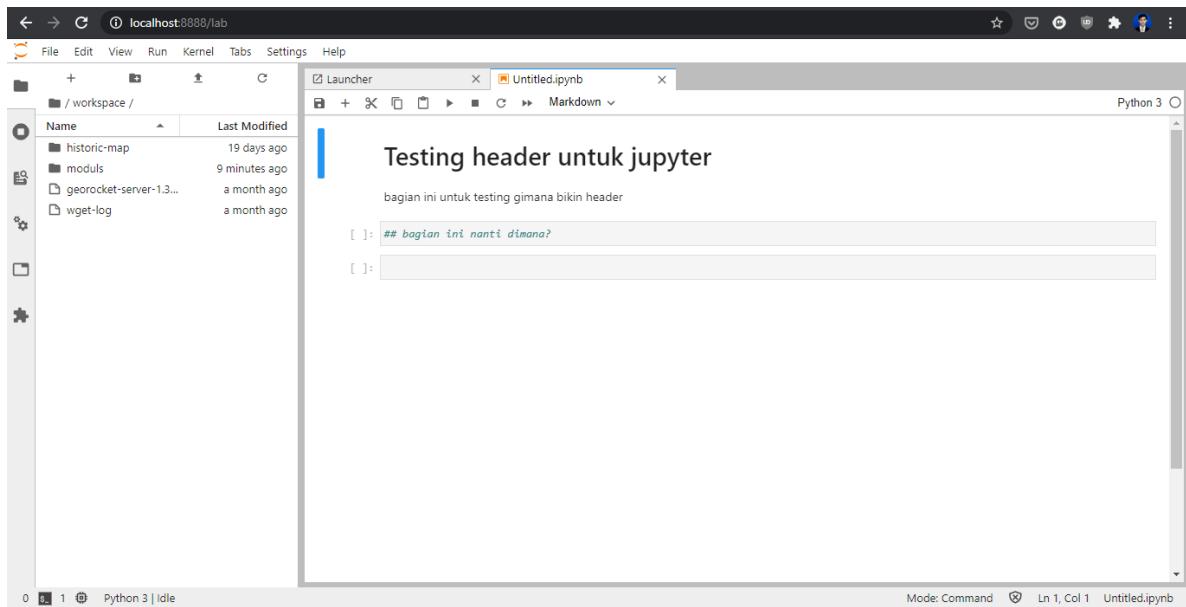
Demikian pula, pada saat Ubuntu memanggil port tertentu, maka WSL akan secara otomatis melakukan *port forwarding*, sehingga port tersebut dapat dibuka pada Windows tanpa perlu melakukan langkah tambahan apapun.

Sebagai contoh, berikut adalah pemanggilan Jupyter Lab dari konsol Ubuntu pada WSL2:

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
danylaksono@DANYLAKSONO:~$ conda activate python38
(python38) danylaksono@DANYLAKSONO:~$ jupyter lab
[I 05:07:31.938 LabApp] JupyterLab extension loaded from /home/danylaksono/miniconda3/envs/python38/lib/python3.8/site-packages/jupyterlab
[I 05:07:31.938 LabApp] JupyterLab application directory is /home/danylaksono/miniconda3/envs/python38/share/jupyter/lab
[I 05:07:32.070 LabApp] Serving notebooks from local directory: /home/danylaksono
[I 05:07:32.070 LabApp] Jupyter Notebook 6.1.4 is running at:
[I 05:07:32.070 LabApp] http://localhost:8888/?token=2683b4662986289bbec992638d8bbe0da0abdf24ff8806a
[I 05:07:32.070 LabApp] or http://127.0.0.1:8888/?token=2683b4662986289bbec992638d8bbe0da0abdf24ff8806a
[I 05:07:32.070 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 05:07:32.420 LabApp]

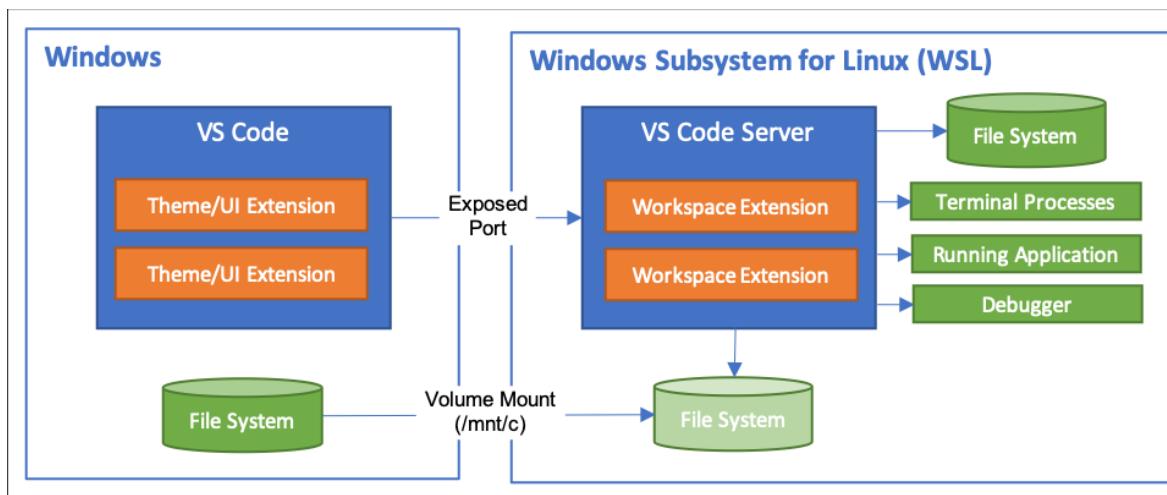
To access the notebook, open this file in a browser:
  file:///home/danylaksono/.local/share/jupyter/runtime/nbsviewer-582-open.html
Or copy and paste one of these URLs:
  http://localhost:8888/?token=2683b4662986289bbec992638d8bbe0da0abdf24ff8806a
  or http://127.0.0.1:8888/?token=2683b4662986289bbec992638d8bbe0da0abdf24ff8806a
```

Dan berikut adalah tampilan Jupyter Lab apabila dibuka pada browser di Windows:



#### 8. WSL2 untuk keperluan pembangunan aplikasi (development)

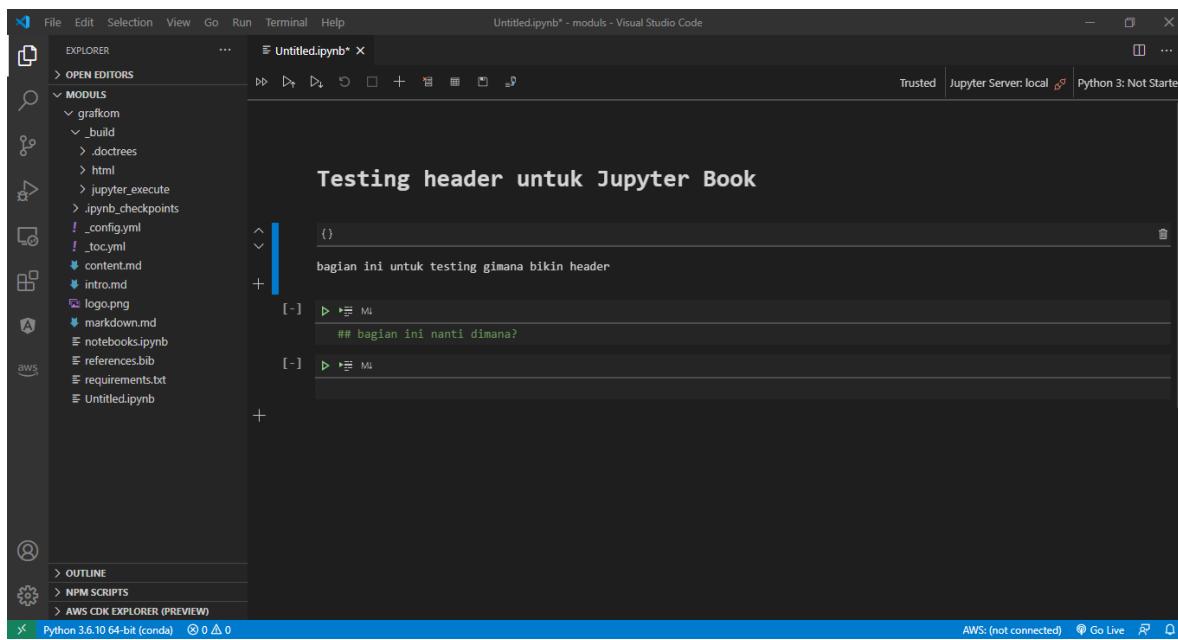
Untuk keperluan pembangunan aplikasi, WSL2 memungkinkan koneksi pada beberapa Code Editor. Sebagai contoh, VSCode memiliki dukungan untuk melakukan *editing* kode pada WSL2 melalui plugin. Klik [Link](#) berikut untuk detilnya.



Berikut adalah contoh pemanggilan VSCode dari mesin Ubuntu:

```
danylaksono@DANYLAKSONO:~/workspace/moduls$ code .
danylaksono@DANYLAKSONO:~/workspace/moduls$
```

dan berikut adalah tampilan dari VSCode yang langsung terbuka dan terkoneksi dengan WSL2 tersebut:



Karena pengaturan *resource* untuk komputasi (memory, space harddisk) dilakukan secara bersama-sama oleh Windows dan WSL2 melalui Linux Kernel, maka penggunaan WSL tidak akan membebani sistem Windows. Dengan demikian, penggunaan Linux pada WSL2 ini adalah metode yang lebih disarankan daripada menggunakan Virtual Machine seperti VirtualBox atau VMWare yang dapat membebani kinerja sistem.

Hasil dari latihan di atas adalah WSL2 dengan Ubuntu sebagai sistem operasi yang dapat digunakan pada Windows. Selain diinstall sebagai WSL yang dapat diakses pada sistem lokal, terdapat beberapa opsi lain yang dapat dilakukan untuk menggunakan Linux. [Link berikut](#) memberikan panduan ringkas mengenai bagaimana melakukan instalasi Linux baik pada Cloud maupun lokal.

---

## 1.2 Pengaturan dan Administrasi Linux

Setelah memiliki sebuah sistem operasi Linux yang dapat digunakan untuk latihan dengan menggunakan WSL, pada bagian ini akan dibahas mengenai beberapa perintah dasar pada Linux.

---

### Catatan

**Ubuntu Pocket Guide** merupakan panduan yang cukup komprehensif untuk ‘penggunaan Ubuntu sehari-hari’. Buku ini dapat diunduh secara gratis dari alamat: [http://www.ubuntupocketguide.com/download\\_main.html](http://www.ubuntupocketguide.com/download_main.html)

---

### 1.2.1 Arsitektur Linux

Sebelum terjun langsung dalam penggunaan Linux, sebaiknya terlebih dahulu dipelajari mengenai filosofi dan arsitektur sebuah sistem operasi Linux. Pemahaman mengenai hal ini akan memudahkan dalam bekerja dengan Linux serta menghindari permasalahan pada saat bekerja dengan Linux.

#### Kernel dan komponen-komponen Linux

Sebelumnya, *kernel linux* telah disebutkan beberapa kali pada panduan ini. Lisensi kernel Linux yang dikembangkan oleh Linus Torvalds sebagai rangkaian kode yang bersifat bebas dan terbuka memungkinkan pengembangan berbagai aplikasi lain di atasnya, yang kemudian melahirkan ratusan distro yang masing-masingnya adalah sebuah sistem operasi sendiri. Tetapi, apa sebenarnya yang dimaksud dengan *Linux Kernel*?

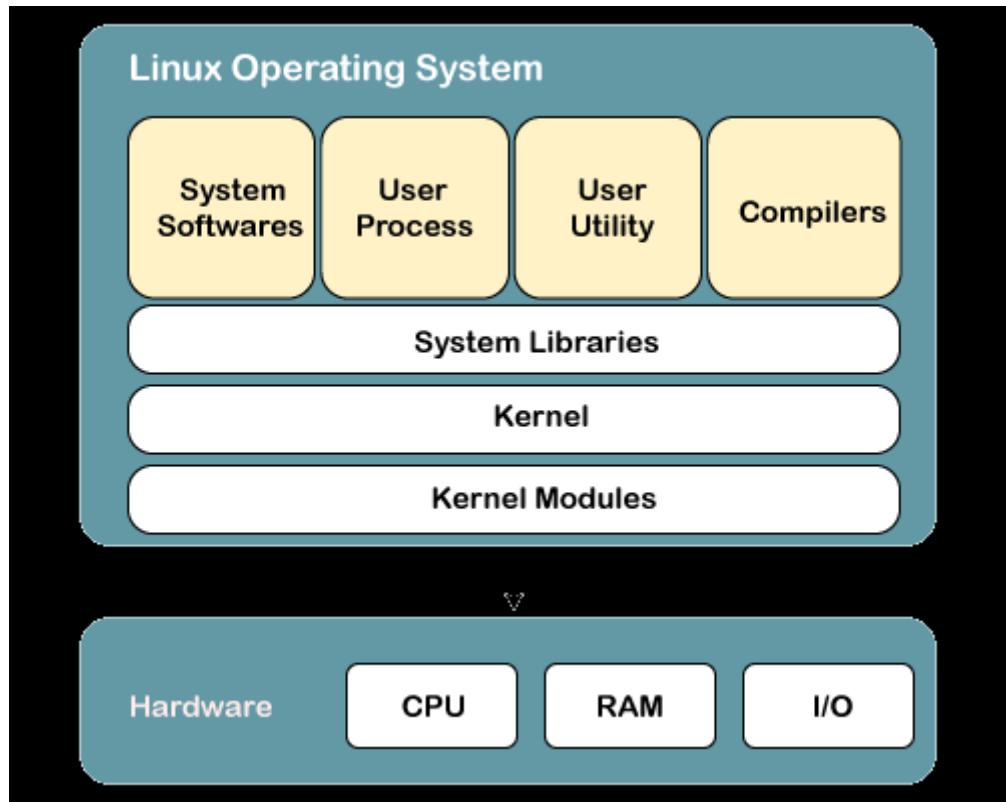


Fig. 1.6: Arsitektur Sistem Operasi Linux

Sebuah ‘kernel’ dapat diartikan sebagai potongan kode yang menjadi antarmuka antara perangkat keras (*hardware*) dan berbagai aplikasi yang menggunakan perangkat tersebut. Linux Kernel yang awalnya didesain oleh Linus Torvalds sebagai sebuah sistem yang tertutup dan tidak portabel ternyata berubah menjadi kode yang dapat digunakan pada berbagai platform dengan dukungan pada berbagai perangkat keras, sehingga muncul berbagai pengembangan versi distribusi Linux yang berbeda.

### Filosofi Linux

Di atas kernel inilah berbagai aplikasi berdiri sebagai perantara antara perangkat keras dan perangkat lunak pada sebuah sistem operasi. Filosofi Linux ini kemudian dikenal luas dengan istilah, “**Everything is a File**”: setiap transaksi, proses, dan apapun yang terjadi pada sebuah sistem Unix sejatinya adalah ‘stream of byte’ yang berdiri di atas kernel.

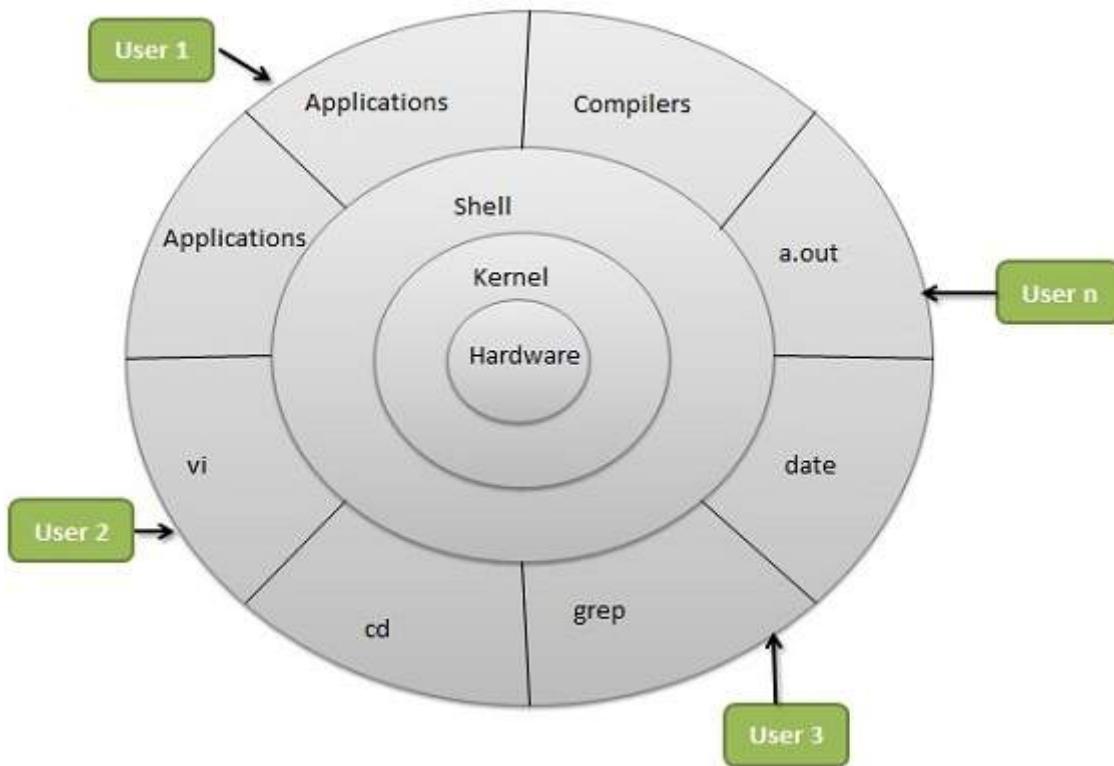


Fig. 1.7: Komponen-komponen Linux

Jika Kernel digunakan untuk berkomunikasi dengan perangkat keras, maka sebuah **Shell** pada Linux berfungsi sebagai penghubung antara pengguna dan kernel. Shell dapat menerima berbagai perintah pengguna dan menerjemahkannya menjadi perintah yang dapat dipahami oleh perangkat keras. Berbagai aplikasi juga menggunakan shell untuk melakukan operasi-operasi yang diinginkan pengguna, misalnya pada antarmuka desktop. Karena setiap operasi adalah ‘file’, maka pengguna yang berbeda dapat melakukan operasi yang berbeda pada sebuah berkas yang sama. Pada Linux, kondisi berikut berlaku:

- Tiap program dijalankan oleh *user/pengguna*
- Tiap berkas dimiliki oleh pengguna
- Tiap pengguna memiliki identitas unik yang membedakan pengguna satu dengan yang lain

Dari uraian di atas, dapat disimpulkan bahwa beberapa karakteristik dari Linux adalah sebagai berikut<sup>1</sup>:

- **Portabel** - Portabilitas berarti perangkat lunak dapat bekerja pada berbagai jenis perangkat keras dengan cara yang sama. Kernel Linux dan program aplikasi mendukung instalasi mereka pada semua jenis platform perangkat keras.
- **Open Source** - Kode sumber Linux tersedia secara gratis dan merupakan proyek pengembangan berbasis komunitas. Beberapa tim bekerja sama untuk meningkatkan kemampuan sistem operasi Linux dan pengembangannya terus berjalan.

<sup>1</sup> [https://www.tutorialspoint.com/operating\\_system/os\\_linux.htm](https://www.tutorialspoint.com/operating_system/os_linux.htm)

- **Multi-User** - Linux adalah sistem multiuser yang berarti banyak pengguna dapat mengakses sumber daya sistem seperti memori / ram / program aplikasi pada saat yang bersamaan.
- **Multiprogramming** - Linux adalah sistem multiprogramming yang berarti banyak aplikasi dapat berjalan pada waktu yang bersamaan.
- **Sistem File Hirarkis** - Linux menyediakan struktur file standar di mana file sistem / file pengguna disusun dengan struktur yang sama pada berbagai distribusi Linux.
- **Shell** - Linux menyediakan program penerjemah khusus yang dapat digunakan untuk menjalankan perintah sistem operasi. Dapat digunakan untuk melakukan berbagai jenis operasi, memanggil program aplikasi. dll.
- **Keamanan** - Linux menyediakan keamanan pengguna menggunakan fitur otentikasi seperti proteksi password dan akses terkontrol ke file/enkripsi data tertentu.

## 1.2.2 Menggunakan Shell

**Shell Command** merupakan antarmuka untuk melakukan berbagai operasi pada sistem operasi Linux. Meskipun saat ini berbagai Distro Linux telah dilengkapi dengan antarmuka yang sangat canggih, penggunaan Shell untuk operasi sehari-hari memiliki beberapa kelebihan:

- Perintah yang kompleks dapat dilakukan dengan mudah menggunakan Shell (coba: `cal`)
- Shell diperlukan pada saat menggunakan Linux sebagai mesin server: perintah shell jauh lebih cepat dan menggunakan bandwith lebih sedikit apabila koneksi dilakukan dengan shell
- Shell dapat diprogram (contoh: `for i in *.png do ...`)
- Penggunaan Shell memungkinkan kustomisasi, misalnya dengan menggunakan argumen untuk berbagai perintah (contoh: *pipe operation*)

Pada bagian ini akan dilakukan beberapa latihan untuk mengenal perintah-perintah shell pada Linux. Perintah-perintah pada panduan ini dilakukan pada Ubuntu, sehingga penggunaan distro yang lain mungkin memerlukan penyesuaian. Meskipun sebagian besar perintah antar distro Linux yang satu dengan yang lain sama, tetapi boleh jadi ada sedikit perbedaan (sebagai contoh, Ubuntu yang berbasis Debian menggunakan Package Manager *apt*, sedangkan CentOS yang berbasis RedHat menggunakan *yum*).

---

### Catatan

**Bash Shell** adalah aplikasi Command Shell yang saat ini merupakan aplikasi standar pada hampir semua distribusi Linux, sehingga perintah pada panduan ini mengacu pada Bash Shell. Perintah-perintah pada Bash Shell dapat dilihat pada [link berikut](#) atau [link berikut](#).

---

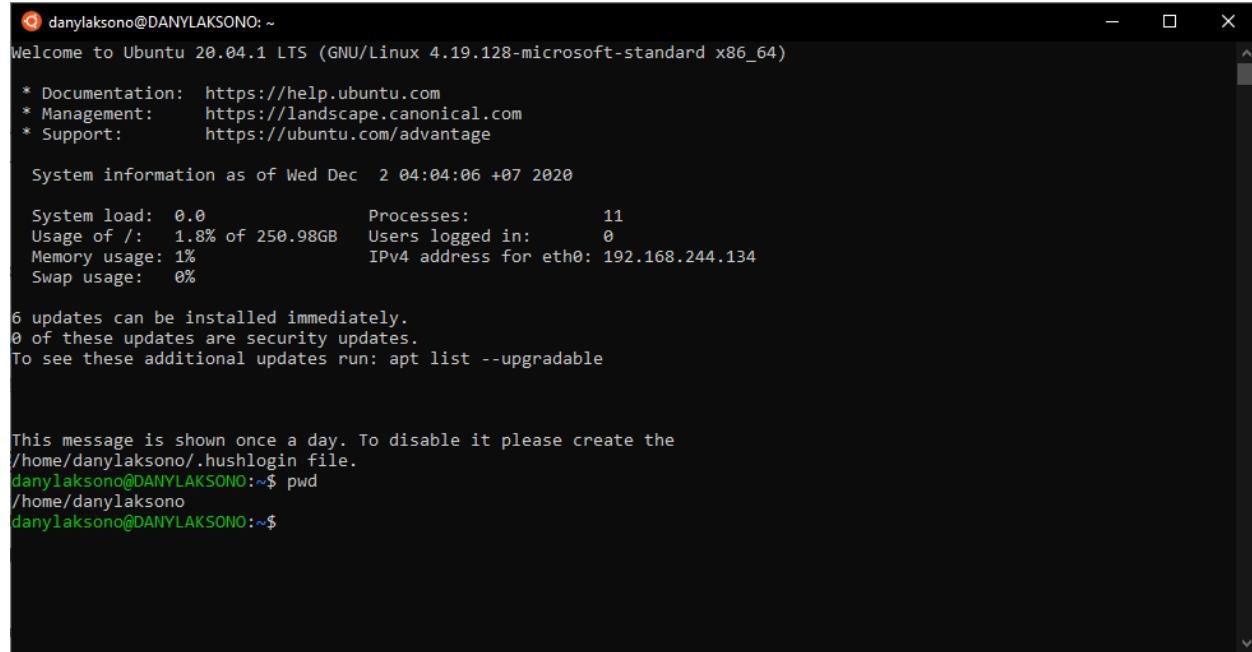
### Latihan: Perintah Dasar pada Shell

WSL2 untuk Windows 10 yang diinstall pada latihan sebelumnya memberikan antarmuka Shell pada pengguna (lihat [gambar sebelumnya](#)). Latihan-latihan berikut menggunakan antarmuka *shell command* yang diberikan oleh WSL2 ini.

Berikut adalah beberapa perintah yang dapat digunakan untuk berlatih manajemen file pada Linux:

Command	Deskripsi
man	bring up manual for a command
exit	log out of shell
clear	clears all output from console
date	output the system date
cal	output a text calendar
uname	print information about the current system

Untuk latihan, buka konsol Ubuntu WSL2, kemudian ketikkan salah satu perintah di atas:



```
danylaksono@DANYLAKSONO: ~
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 4.19.128-microsoft-standard x86_64)

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

 System information as of Wed Dec  2 04:04:06 +07 2020

 System load:  0.0          Processes:           11
 Usage of /:   1.8% of 250.98GB  Users logged in:     0
 Memory usage: 1%
 Swap usage:   0%
 IPv4 address for eth0: 192.168.244.134

6 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

This message is shown once a day. To disable it please create the
/home/danylaksono/.hushlogin file.
danylaksono@DANYLAKSONO:~$ pwd
/home/danylaksono
danylaksono@DANYLAKSONO:~$
```

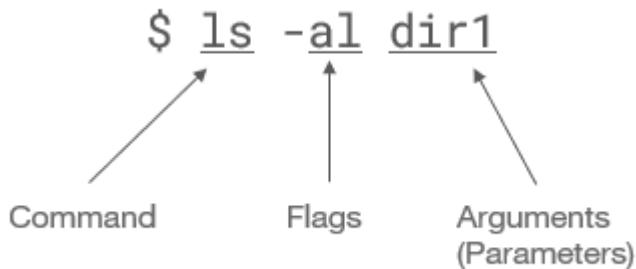
Perintah `pwd` pada gambar di atas digunakan untuk menampilkan direktori tempat shell saat ini berjalan. Keluaran dari perintah ini langsung ditampilkan pada konsol. Untuk latihan, jalankan perintah-perintah lain dan perhatikan apa yang terjadi pada konsol.

## Latihan: Operasi pada File dan Direktori

Sebagaimana pada Windows, berkas pada Linux disusun dalam beberapa folder yang masing-masing mengatur pekerjaan tertentu. Pada Linux, struktur direktorinya adalah sebagai berikut:

Location	Details
/bin	Essential software, typically needed to get the system running
/boot	Files related to the boot menu/loader
/dev	Virtual files representing hardware devices
/etc	System (global) configuration files
/home	Users' personal folders
/lib	Support (library) files required by software
/media	Contains subfolders where storage devices can be mounted
/proc	Virtual folder containing files representing stats and settings
/root	Personal folder of the root user
/sbin	Essential software for system maintenance, usually used only by the root user
/tmp	Temporary files/folders
/usr	Essentially, subdirectories containing most software used on the system, including system libraries and documentation
/var	Data that is vital to the running of the system and that is constantly being updated

Perintah pada Shell umumnya mengikuti struktur sebagai berikut:



Command adalah perintah yang dipanggil pada shell. Flags menunjukkan opsi yang dapat dipanggil oleh command tersebut, sedangkan Argument menyatakan variabel yang dipanggil oleh perintah tersebut.

- Untuk berpindah antar satu direktori ke direktori lain, dapat digunakan *Relative Directory*, dimana struktur direktori dapat diakses secara hirarkis.

Command	Deskripsi
.	direktoy saat ini
..	direktoy yang berada satu tingkat di atas direktori saat ini
~	home directory untuk pengguna saat ini
/	root directory
/apt	folder <b>apt</b> pada root

2. Selanjutnya, perintah-perintah berikut dapat digunakan pada saat berhubungan dengan pengaturan direktori:

Command	Deskripsi
ls	tampilkan daftar berkas pada direktori saat ini
pwd	print working directory - cetak direktori saat ini
cd	change directory - berpindah ke direktori baru
mkdir	membuat direktori baru
rmdir	menghapus direktori (isi folder harus kosong)
rm	hapus file atau direktori ( <code>rm -rf</code> untuk menghapus secara paksa)

3. Gunakan perintah `man <namaperintah>` atau `namaperintah --help` untuk menampilkan opsi apa saja yang dapat digunakan pada tiap perintah tersebut. Sebagai contoh berikut adalah manual page untuk perintah `ls`:

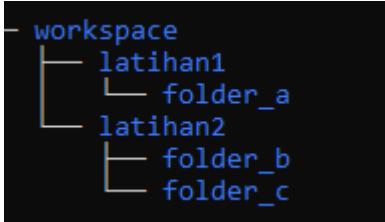
```
ls(1)                               User Commands                               LS(1)
NAME
    ls - list directory contents
SYNOPSIS
    ls [OPTION]... [FILE]...
DESCRIPTION
    List information about the FILEs (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
    Mandatory arguments to long options are mandatory for short options too.
    -a, --all
        do not ignore entries starting with .
    -A, --almost-all
        do not list implied . and ..
    --author
        with -l, print the author of each file
    -b, --escape
        print C-style escapes for nongraphic characters
    --block-size=SIZE
        with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below
    -B, --ignore-backups
        do not list implied entries ending with ~
    -c      with -lt: sort by, and show, ctime (time of last modification of file status information); with -l: show ctime and sort by name; otherwise: sort by
           ctime, newest first
    -C      list entries by columns
```

4. Untuk latihan selanjutnya, tampilkan isi dari **root directory** (satu tingkat di atas home directory), seperti

```

lrwxrwxrwx  1 root root      7 Aug  5 04:39 bin -> usr/bin
drwxr-xr-x  2 root root   4096 Aug  5 04:47 boot
drwxr-xr-x  8 root root   2780 Dec  1 16:39 dev
drwxr-xr-x 92 root root   4096 Dec  1 16:39 etc
drwxr-xr-x  3 root root   4096 Sep 29 17:17 home
-rw xr-xr-x  2 root root 631968 Sep 29 16:32 init
lrwxrwxrwx  1 root root      7 Aug  5 04:39 lib -> usr/lib
lrwxrwxrwx  1 root root      9 Aug  5 04:39 lib32 -> usr/lib32
lrwxrwxrwx  1 root root      9 Aug  5 04:39 lib64 -> usr/lib64
lrwxrwxrwx  1 root root     10 Aug  5 04:39 libx32 -> usr/libx32
drwx----- 2 root root 16384 Apr 10 2019 lost+found
drwxr-xr-x  2 root root   4096 Aug  5 04:39 media
drwxr-xr-x  6 root root   4096 Sep 29 17:17 mnt
drwxr-xr-x  2 root root   4096 Aug  5 04:39 opt
dr-xr-xr-x 92 root root      0 Dec  1 16:39 proc
drwx----- 2 root root   4096 Aug  5 04:42 root
drwxr-xr-x  9 root root   180 Dec  1 19:11 run
lrwxrwxrwx  1 root root      8 Aug  5 04:39 sbin -> usr/sbin
drwxr-xr-x  2 root root   4096 Jul 10 20:59 snap
drwxr-xr-x  2 root root   4096 Aug  5 04:39 srv
dr-xr-xr-x 11 root root      0 Dec  2 05:07 sys
drwxrwxrwt  8 root root   4096 Dec  2 02:28 tmp
drwxr-xr-x 14 root root   4096 Aug  5 04:40 usr
drwxr-xr-x 13 root root   4096 Aug  5 04:42 var
berikut:
```

5. Selanjutnya, buatlah struktur folder berikut pada home directory Anda (pengguna aktif saat ini):



Buat folder `workspace` yang didalamnya berisi dua buah folder, masing-masing folder `latihan1` dan `latihan2`. Di dalam folder `latihan1` berisi sebuah folder lain.

### Catatan

Berbagai perintah pada *shell* Linux sangat efektif apabila digunakan dengan baik. Perintah yang berbeda dapat dikombinasikan dan dioptimalkan untuk mempersingkat tugas-tugas yang diberikan. Sebagai contoh, beberapa perintah dapat menerima lebih dari satu argumen (misalnya, `mkdir folder1 folder2`). Penggunaan operator pada command juga dapat menyingkat operasi yang diinginkan (misalnya, `sudo apt update && sudo apt upgrade` dengan operator `AND_IF`), operasi ekspansi dengan bracket (misalnya `cp /alamat/file.txt{,.bak}`)

Sebagai catatan tambahan, Shell pada Linux bahkan mendukung **Regex!**

6. Untuk bekerja dengan file, beberapa perintah berikut dapat digunakan:

Command	Deskripsi
cp	copy a file
mv	move a file (also used to rename files)
rm	remove the given file
touch	create empty file, or change time-modified

Gunakan perintah di atas untuk membuat file `berkas1.txt` pada `folder_a` di dalam folder `latihan1`. Selanjutnya, lakukan perintah copy dan rename untuk menghasilkan struktur file seperti berikut:

```
workspace
└── latihan1
    └── folder_a
        └── berkas1.txt
└── latihan2
    ├── berkas2.dat
    └── folder_b
        ├── berkas2.dat
        └── latihan1
            └── folder_a
                └── berkas1.txt
└── folder_c
    └── berkas1.txt
```

untuk melakukan editing pada file, kebanyakan distro Linux menyediakan shell editor bawaan, misalnya nano.

7. Terakhir, hapus **folder\_b** berikut semua file dan sub-direktori di dalamnya. Hasil akhirnya sebagai berikut:

```
- workspace
  └── latihan1
    └── folder_a
        └── berkas1.txt
  └── latihan2
    ├── berkas2.dat
    └── folder_c
        └── berkas1.txt
```

Dengan demikian, latihan ini menunjukkan bagaimana manajemen terhadap file dilakukan di Linux. Berbagai perintah dapat digabungkan untuk memperoleh hasil yang diinginkan, misalnya membuat beberapa folder baru sekaligus atau membuat dan mengcopy berkas secara cepat.

### Latihan: Instalasi aplikasi

Ubuntu menggunakan APT (*Advanced Package Tool*) sebagai *package manager* untuk mengatur seluruh aplikasi yang digunakan. Sebuah Package manager menghubungkan antara Repository dan mesin linux yang kita gunakan. **Repositori Linux** merupakan lokasi penyimpanan tempat sistem Anda mengambil dan menginstal pembaruan dan aplikasi OS. Setiap repositori adalah kumpulan perangkat lunak yang disimpan pada server jarak jauh dan digunakan untuk menginstal dan memperbarui paket perangkat lunak pada sistem Linux.

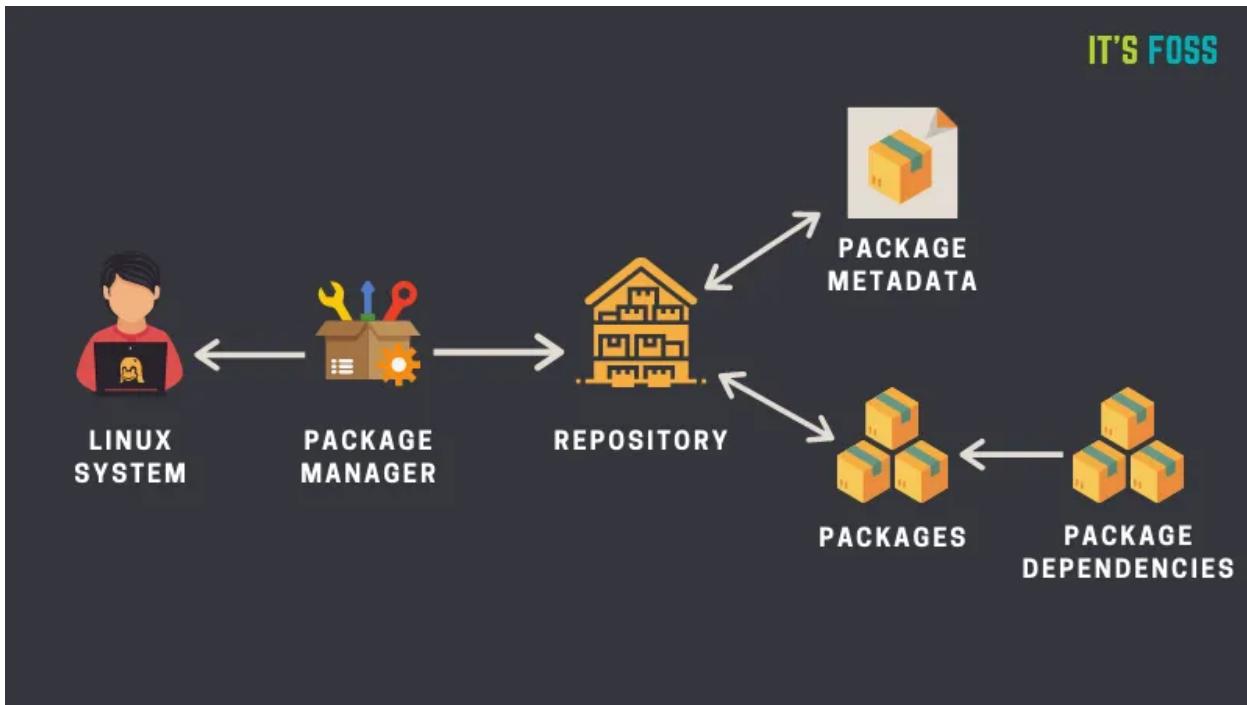


Fig. 1.8: Cara kerja Package Manager pada Linux

Pada pemanggilan berkas yang tersimpan di repository, berlaku kondisi sebagaimana akses informasi melalui internet pada umumnya: semakin jauh secara fisik lokasi server, semakin lambat koneksi yang akan diperoleh. Untuk itu, terdapat sejumlah repository lokal untuk tiap versi distribusi Linux. Sebagai contoh, <http://repo.ugm.ac.id/> berisi repository untuk beberapa distribusi Linux yang disimpan pada server UGM.

Terdapat beberapa antarmuka yang berhubungan dengan Package Manager yang dikenal di sistem operasi berbasis Debian, seperti *apt-get*, *aptitude*, maupun *apt*. Aplikasi antarmuka ini menghubungkan perintah dari pengguna (misalnya *sudo apt update*) dengan *package manager* (seperti APT) untuk kemudian melakukan operasi terkait dengan basisdata aplikasi pada repository.

Latihan berikut diberikan untuk memahamkan bagaimana menggunakan *apt* untuk memanajemen aplikasi pada Linux<sup>2</sup>.

1. Melakukan *update* basisdata APT Pada Ubuntu dan distro berbasis Debian lainnya, aplikasi yang terinstall disimpan pada basisdata. Untuk melakukan update atau pembaruan aplikasi dari repository, gunakan perintah:

```
sudo apt update
```

Keluaran perintah tersebut adalah seperti berikut:

<sup>2</sup> <https://itsfoss.com/apt-command-guide/>

```
danylaksono@DANYLAKSONO:~/dummy$ sudo apt update
[sudo] password for danylaksono:
Hit:1 https://dl.yarnpkg.com/debian stable InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [109 kB]
Hit:3 http://archive.ubuntu.com/ubuntu focal InRelease
Get:4 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [383 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [87.1 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [5524 B]
Get:9 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [75.5 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [12.1 kB]
Get:11 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [519 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [691 kB]
Get:13 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [9364 B]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/main Translation-en [171 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [11.3 kB]
Get:16 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [101 kB]
Get:17 http://archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [15.5 kB]
Get:18 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [699 kB]
Get:19 http://archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [137 kB]
Get:20 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [13.3 kB]
Fetched 3254 kB in 6s (580 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
4 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Hasil dari `apt update` adalah daftar aplikasi yang dapat diperbarui. Untuk mengeksekusi pembaruan aplikasi, gunakan perintah:

```
sudo apt upgrade
```

Kedua perintah di atas dapat diringkas menggunakan `&&` :

```
sudo apt update && sudo apt upgrade
```

**Caution:** Upgrade adalah hak *sudoers*. Apa yang terjadi jika perintah di atas dijalankan tanpa `sudo`?

2. `apt` dapat digunakan untuk melakukan instalasi paket aplikasi. sebagai contoh, perintah berikut akan menginstall aplikasi `tree`:

```
sudo apt install tree
```

selanjutnya, perintah `tree` dapat langsung digunakan pada shell.

1. Untuk menghapus aplikasi, gunakan perintah:

```
sudo apt purge <nama aplikasi>
```

atau

```
sudo apt remove <nama aplikasi>
```

perbedaan keduanya terletak pada kedalaman penghapusan. `purge` menghapus seluruh aplikasi berikut konfigurasi yang tersimpan, sedangkan `remove` hanya menghapus aplikasi dari daftar basisdata **APT**, tetapi meninggalkan konfigurasi dari aplikasi, sehingga apabila kelak dilakukan instalasi kembali, konfigurasi ini akan digunakan.

1. Sebagai latihan, lakukan instalasi untuk aplikasi editing bernama vim. vim dapat digunakan untuk mengedit sebuah file yang memiliki tingkat kompleksitas tinggi, dan akan membantu pada saat kita melakukan editing file, misalnya pada sebuah mesin server.

### 1.2.3 Pengaturan Pengguna dan Hak Akses

Sebagaimana disebutkan, Linux merupakan sistem operasi MultiUser dan Multiprogramming, dimana tiap berkas dimiliki oleh seorang pengguna, dan tiap pengguna memiliki hak akses tertentu. *Sudoers* merupakan kelas pengguna yang memiliki hak akses lebih tinggi (*elevated*) dibandingkan pengguna biasa, sehingga mampu melakukan berbagai perintah yang tidak diizinkan untuk pengguna yang tidak termasuk dalam **grup 27 (sudo)**, misalnya apt upgrade.

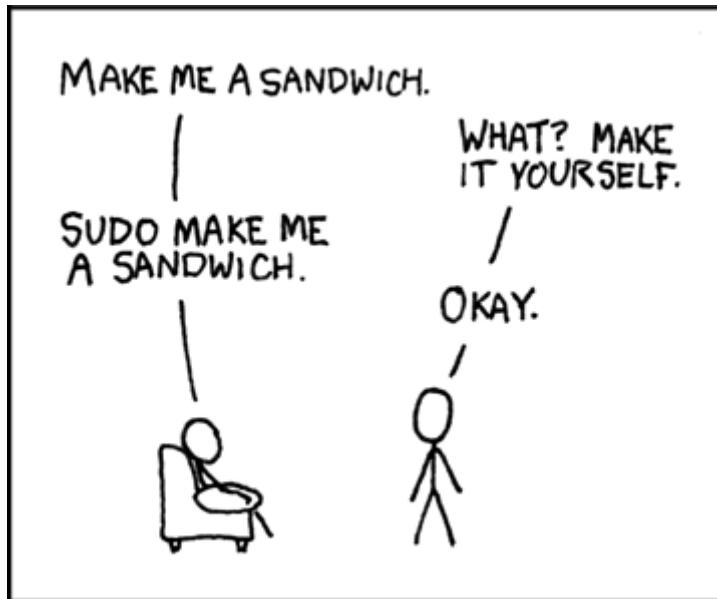


Fig. 1.9: Sumber: <https://xkcd.com/149/>

Beberapa perintah terkait user pada Linux adalah sebagai berikut:

Command	Deskripsi
whoami	print your username
id	print user id and group membership
users	list logged-in users (short)
who	list logged-in users (long)
finger	print information about users

### Latihan: Administrasi Pengguna

Latihan berikut akan memberikan gambaran mengenai pengaturan pengguna dan hak akses pada berkas atau direktori<sup>3</sup>.

1. Membuat user baru Akun pengguna baru dapat digunakan pada sebuah sistem Enterprise, dimana satu sistem operasi yang sama digunakan oleh beberapa pengguna yang berbeda. Untuk menambahkan pengguna baru, gunakan perintah berikut:

```
sudo adduser namauser
```

Setelah user baru dibuat, kita akan diminta untuk memasukkan nama pengguna dan password, serta beberapa informasi lainnya.

2. Merubah password Setelah membuat akun, langkah selanjutnya adalah merubah password untuk akun tersebut:

```
sudo passwd namauser
```

### Latihan: Pengaturan Grup

Grup pada Linux dapat digunakan untuk mendefinisikan peranan yang berbeda untuk tiap pengguna, misalnya membatasi akses pada direktori serta akses untuk perintah tertentu. Untuk memasukkan pengguna yang sudah dibuat pada sebuah grup, perintahnya adalah sebagai berikut:

1. Menambahkan pengguna ke grup

```
sudo usermod -a -G namagrup namauser
```

2. Jika grup yang dimaksud adalah **sudoer**, maka perintahnya adalah sebagai berikut:

```
usermod -aG sudo namauser
```

3. Selanjutnya, lakukan penambahan akses pengguna ini ke dalam **sudoer**:

```
sudo visudo
```

masukkan informasi nama user sebagai **sudoer**

```
namauser  ALL=(ALL) NOPASSWD:ALL
```

4. Berpindah akun (switch user) Untuk keperluan administrasi, seringkali pengguna harus berpindah dari satu akun (non-sudoer) ke akun lain (sudoer). Untuk itu, perintah berikut dapat digunakan:

```
su -u userkedua  
whoami
```

Untuk mengakses *superelevated user* atau **root**, perintah berikut dapat digunakan:

```
sudo su  
whoami
```

<sup>3</sup> <https://vitux.com/a-beginners-guide-to-user-management-on-ubuntu/>

Grup dapat digunakan untuk mengatur dan mengelompokkan pengguna pada satu hak akses yang sama. Selanjutnya adalah bagaimana melakukan pengaturan hak akses tersebut.

## Pengaturan Hak Akses

Sebagaimana disebutkan sebelumnya, akses terhadap file dan perintah pada Linux dapat dibatasi untuk user tertentu. Izin (*Permission*) ini pada linux dibagi menjadi:

- **owner** (u) - user yang membuat file. ini dapat dirubah menggunakan perintah `chown`
- **group** (g) - grup yang berhubungan dengan tiap file
- **others** (o) - orang lain yang bukan owner maupun anggota grup yang dapat mengakses file tersebut ('=semua orang')

Sebagai contoh, jika kita menggunakan perintah `ls -al` berikut:

```
danylaksono@DANYLAKSONO:~/dummy/workspace$ ls -al
total 16
drwxr-xr-x 4 danylaksono danylaksono 4096 Dec  2 04:58 .
drwxr-xr-x 3 danylaksono danylaksono 4096 Dec  2 04:57 ..
drwxr-xr-x 3 danylaksono danylaksono 4096 Dec  2 05:00 latihan1
drwxr-xr-x 3 danylaksono danylaksono 4096 Dec  2 05:28 latihan2
danylaksono@DANYLAKSONO:~/dummy/workspace$
```

akan muncul keterangan (`drwxr-xr-x`) mengenai status dan hak akses pada file tersebut, yaitu:

- **d** : menunjukkan bahwa berkas tersebut adalah sebuah *directory*
- **rwx** : permission (izin) yang diberikan pada pemilik file untuk *read, write dan execute*
- **r-x** : izin *grup*, pengguna dari grup danylaksono (kebetulan untuk contoh di atas nama pengguna dan grup sama) dapat melakukan *read dan execute* pada file tersebut
- **r-x** : izin untuk *other*, atau pengguna lain

Penulisan izin tersebut juga dapat dilakukan menggunakan **notasi numerik**:

- 0: Tidak ada izin
- 1: Jalankan (x)
- 2: Tulis (w)
- 4: Baca (r)

Sekarang, bagaimana cara menghitung izin untuk pengguna dan grup dalam notasi numerik? Cukup tambahkan nilai izin untuk mendapatkan nilai pengguna, grup, dan izin lainnya masing-masing.

Sebagai contoh:

- Izin *baca* (4), *tulis* (2) dan *jalankan* (1): izin **rwx** diterjemahkan menjadi 7 ( $4 + 2 + 1$ )
- Izin *baca* (4) dan *tulis* (2): izin **rw-** diterjemahkan menjadi 6 ( $4 + 2$ )
- Izin *tulis* (2) dan *jalankan* (1): izin **-wx** diterjemahkan menjadi 3 ( $2 + 1$ ), dan seterusnya.

**Warning:** Kira-kira apa pengaruh `chmod 777`?

Dengan menggunakan perintah `chmod`, izin suatu file dapat dirubah. perintah ini menggunakan notasi:

```
shum@sol:~ $ ls -l
total 20
drwx----- 2 shum      staff        4096 Jan 16 22:04 Mail
drwx----- 3 shum      staff        4096 Jan 16 14:15 csc128
drwxr-xr-x  2 shum      staff        4096 Jan 13 16:42 public
drwxr-xr-x  2 shum      staff        4096 Jan 16 14:07 public_html
-rw-r--r--  1 shum      staff       628 Jan 15 20:04 verse
```

The diagram illustrates the breakdown of the file listing from the command `ls -l`. It shows the fields and their meanings:

- file type**: The first column indicates the file type (e.g., directory, regular file).
- permissions**: The second column shows the permissions for the user (owner), group, and others. The permissions are broken down into three groups:
  - user permissions** (green arrows): r (readable), w (writeable), x (executable).
  - group permissions** (red arrows): r (readable), w (writeable), x (executable).
  - other (everyone) permissions** (blue arrows): r (readable), w (writeable), x (executable).
- number of hard links**: The third column shows the count of hard links.
- user (owner) name**: The fourth column shows the user name.
- group name**: The fifth column shows the group name.
- size**: The sixth column shows the file size.
- date/time last modified**: The seventh column shows the date and time the file was last modified.
- filename**: The eighth column shows the file name.

Fig. 1.10: Penjelasan hak akses pada Linux

- tanda minus (-), yang berarti “hapus izin ini”
- tanda tambah (+), yang berarti “tambahkan izin ini”
- tanda sama dengan (=), yang berarti “ubah izin menjadi persis seperti ini”.

Sebagai contoh:

```
chmod +x latihan.txt
```

maksudnya adalah memberi izin kepada semua pengguna untuk mengakses file tersebut. Ini sama dengan `chmod a+x latihan1.txt`, dimana **a** berarti ‘all’. Hal yang sama berlaku untuk *grup* (g) dan *other* (o).

Perintah lain yang berguna untuk merubah kepemilikan berkas pada Linux adalah `chown`. Format untuk perintah ini adalah:

```
chown user[:group] namafile
```

Perintah di atas akan merubah owner untuk file `namafile` menjadi `user` pada grup `group`. Selain file, perintah ini juga dapat digunakan untuk merubah kepemilikan sebuah folder atau direktori. Untuk kasus ini, flag `-R` berguna untuk merubah kepemilikan sebuah direktori secara rekursif, yaitu sekaligus merubah seluruh subdirektori dan berkas yang ada di dalamnya.

```
chown -R user[:group] direktori
```

Sebagai latihan untuk sesi ini, buat user kedua, kemudian rubah akses pada file `latihan1.txt` yang sudah dibuat sebelumnya agar user lain tidak dapat membaca file tersebut. Apa pengaruhnya apabila user pertama hendak mengakses file tersebut?

---

### Catatan

Penggunaan DAC (*Discretionary Access Control*) seperti di atas akan lebih baik apabila dilengkapi dengan pengaturan ACL (*Access Control List*) atau SELinux pada distribusi seperti Centos untuk manajemen berkas pada server.

---

## 1.3 Linux sebagai Sistem Operasi Server

Linux merupakan salah satu sistem operasi yang paling banyak digunakan untuk server, karena kemampuan enterprise dan biaya yang murah (bahkan gratis). Atas alasan itulah, Linux banyak dijumpai pada berbagai platform mesin server, baik dalam skala besar maupun kecil. Pada bagian ini akan dilakukan pengaturan Linux untuk sebagai server dengan menggunakan beberapa perangkat *webserver* dan *application server*.

### 1.3.1 Perangkat Lunak untuk Web Server dan Application Server

Sebuah aplikasi **webserver** berguna untuk mempublikasi file lokal agar dapat diakses melalui jaringan. Sebagai contoh, halaman HTML yang dibuat di komputer belum akan dapat diakses secara luas apabila tidak dipasang pada aplikasi webserver yang dapat mempublikasi data ini. Beberapa contoh perangkat Webserver adalah sebagai berikut:

- Apache HTTP Server
- Nginx
- LightHTTPd
- Microsoft IIS

- NodeJS
- Flask

Apache adalah salah satu webserver berbasis PHP yang paling sering digunakan. Dengan Apache, maka berkas yang tersimpan pada komputer akan dapat dionlinekan dengan pengaturan tertentu. Pengaturan yang dimaksud meliputi pengaturan keamanan, *virtual host*, *proxy*, dan lain sebagainya.

**Application Server** merupakan sebuah kelompok *framework* perangkat lunak yang memungkinkan sebuah *web application* untuk disajikan melalui internet. Contoh application server adalah *Java Servlet Container*, yaitu perangkat yang berfungsi untuk menyajikan aplikasi yang dibuat dengan bahasa Java pada internet. Beberapa Contoh Application Server adalah sebagai berikut:

- Apache Tomcat
- Eclipse Jetty
- ASP.Net
- ZendServer
- JBoss EAP

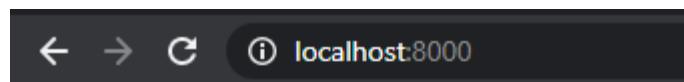
Jika sebuah *webserver* umumnya menyajikan *static content*(seperti HTML, CSS, Javascript, gambar, video, dst) maka sebuah *application server* menyajikan *bussiness logic* yang memungkinkan aplikasi yang disajikan bersifat dinamis.

### Latihan: Menjalankan sebuah Web Server Sederhana

Untuk sekedar menguji kemampuan *port forwarding* dari WSL2 dan demonstrasi webserver, kita akan coba untuk membuat sebuah webserver sederhana. Python menyediakan sebuah webserver sederhana dengan perintah singkat:

```
python3 -m http.server
```

maka folder tempat perintah tersebut dijalankan akan dapat diakses pada browser pada alamat `localhost:8000` (port default untuk server ini):



## Directory listing for /

---

- [file2.dat](#)
  - [latihan1/](#)
  - [latihan2/](#)
- 

Apabila kita membuat sebuah file HTML pada folder tersebut, maka berkas tersebut dapat dibuka pada browser sebagai sebuah halaman web.

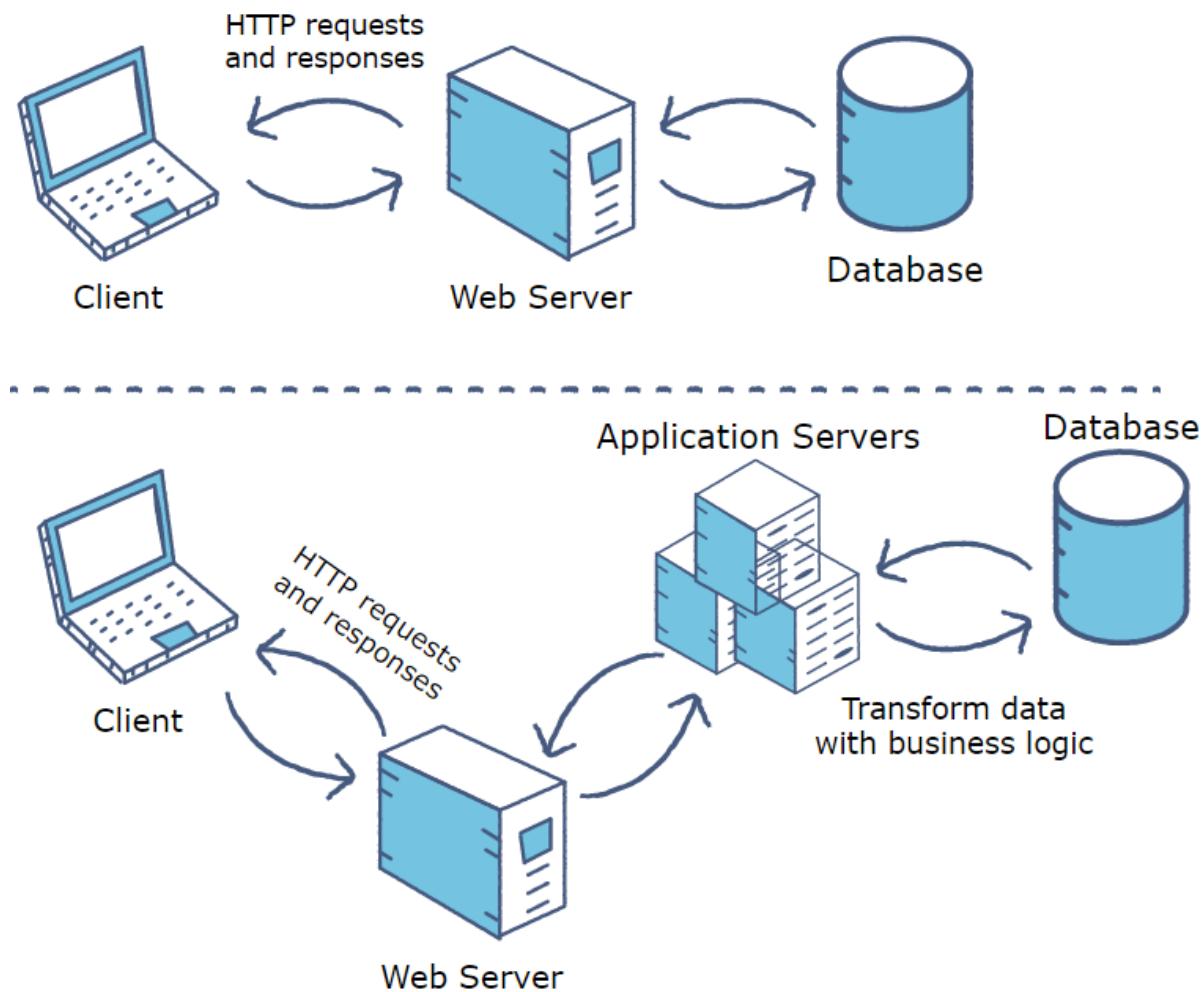


Fig. 1.11: Perbandingan App Server dan Web Server

### Instalasi Apache HTTP Server

Server web \*\*Apache HTTP Server \*\*adalah perangkat paling populer saat ini untuk menyajikan konten web di internet. Apache menyumbang lebih dari setengah dari seluruh situs web aktif di internet dengan berbagai fungsi dan kelebihan yang dimilikinya. Pada latihan ini akan dilakukan instalasi dan konfigurasi Apache untuk publikasi data dalam bentuk HTML sederhana.

Lakukan langkah berikut untuk melakukan instalasi Apache pada sistem Ubuntu WSL:

1. Update basisdata aplikasi di Ubuntu

```
sudo apt update
```

2. Instalasi Apache2 bash sudo apt install apache2
3. Pengaturan firewall. Firewall berfungsi untuk 'mencegat' terjadinya transfer data antara sistem lokal dengan jaringan internet. Pengaturan ini diperlukan agar apache diizinkan untuk mengakses port yang diperlukan untuk berkomunikasi dengan dunia luar

```
```bash
sudo ufw allow 'Apache'
```

```

Selanjutnya periksa status `ufw` menggunakan perintah:

```
```bash
sudo ufw status
```

```

Apabila keluarannya seperti berikut, artinya port berhasil dibuka:

```
```
Output
Status: active

To           Action      From
--           -----      ---
OpenSSH       ALLOW       Anywhere
Apache        ALLOW       Anywhere
OpenSSH (v6)  ALLOW       Anywhere (v6)
Apache (v6)   ALLOW       Anywhere (v6)
```

```

1. Jalankan Apache setelah instalasi

```
sudo service apache2 start
```

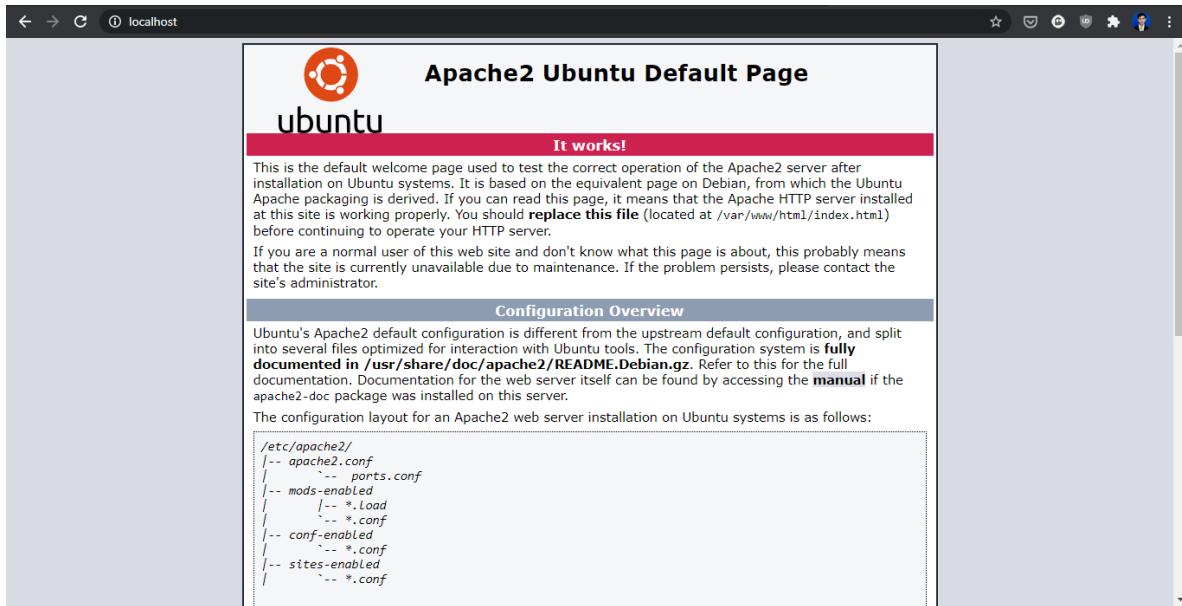
Periksa apakah Apache sudah dijalankan oleh Service

```
sudo service apache2 status
```

Apabila status menunjukkan 'running', artinya webserver Apache sudah berhasil dijalankan

```
(python38) danylaksono@DANYLAKSONO:~/dummy/workspace$ sudo service apache2 start
 * Starting Apache httpd web server apache2
 *
(python38) danylaksono@DANYLAKSONO:~/dummy/workspace$ sudo service apache2 status
 * apache2 is running
(python38) danylaksono@DANYLAKSONO:~/dummy/workspace$
```

1. Buka localhost melalui browser pada Windows



Apabila halaman Apache Ubuntu sudah terbuka, artinya instalasi Apache berhasil dengan baik

WSL menggunakan *port forwarding* dan perubahan DNS lola pada alamat localhost, sehingga localhost pada WSL akan sama dengan localhost pada Windows. Untuk melihat alamat IP dari port yang digunakan oleh WSL, gunakan perintah ip addr:

```
danylaksono@DANYLAKSONO:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: bond0: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 96:5a:1f:55:d6:c7 brd ff:ff:ff:ff:ff:ff
3: dummy0: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether f6:03:57:db:dc:fc brd ff:ff:ff:ff:ff:ff
4: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
5: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:0e:52:d5 brd ff:ff:ff:ff:ff:ff
    inet 172.17.27.221/28 brd 172.17.27.223 scope global eth0
        valid_lft forever preferred_lft forever
        inet6 fe80::215:5dff:fe0e:52d5/64 scope link
            valid_lft forever preferred_lft forever
```

pada bagian eth0, terdapat ip address yang akan membuka halaman yang sama dengan localhost apabila diakses di Windows. Dalam hal ini alamat tersebut adalah 172.17.27.221.

## Membuat Website sederhana

Setelah webserver selesai dipasang, selanjutnya adalah menggunakan webserver ini untuk mempublikasi halaman web dalam bentuk HTML. Apache memiliki direktori default dalam publikasi webnya di folder:

```
/var/www/html
```

Dengan demikian, seluruh file yang disimpan pada folder tersebut akan dapat dibuka melalui browser pada alamat `localhost`. Untuk menguji webserver ini, kita akan lakukan latihan sebagai berikut:

1. Masuk ke folder `/var/www/html` menggunakan perintah `cd`.
2. Buat sebuah folder baru dengan nama `trial` pada direktori `/html` tersebut.

```
sudo mkdir trial
```

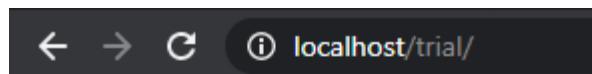
Perintah `mkdir` pada folder tersebut memerlukan akses `sudo`, karena owner dari folder adalah `root`.

3. Pada folder tersebut, buat sebuah file dengan nama `index.html`

```
sudo touch index.html
```

4. Isilah file tersebut dengan baris bahasa HTML berikut:

```
<html>
<head></head>
<body>
<h1> The hello world</h1>
<p> Ini paragraf html </p>
</body>
</html>
```



## The hello world

Ini paragraf html

5. Buka `http://localhost/trial:`  
folder berhasil dipanggil pada webserver

File html yang dibuat pada

### 1.3.2 Tomcat sebagai Servlet Aplikasi berbasis Java

Apache Tomcat adalah implementasi open-source dari Java Servlet, JavaServer Pages, Java Expression Language, dan teknologi Java WebSocket. Tomcat merupakan salah satu web Application yang paling banyak diadopsi di dunia saat ini. Tomcat mudah digunakan dan memiliki ekosistem add-on yang kuat. Apache Tomcat digunakan untuk publikasi berbagai aplikasi geospasial berbasis Java, seperti misalnya Geoserver, GeoTools dan Mapstore.

#### Instalasi Tomcat

Untuk melakukan instalasi Tomcat, lakukan langkah berikut: <https://linuxize.com/post/how-to-install-tomcat-9-on-ubuntu-18-04/>

1. Update apk, kemudian install OpenJDK. OpenJDK diperlukan oleh Tomcat yang berbasis Java untuk dapat dijalankan pada Ubuntu

```
sudo apt install default-jdk
```

2. Atas alasan keamanan, Tomcat tidak boleh dijalankan di bawah pengguna root. Kita perlu membuat pengguna dan grup sistem baru dengan direktori home /opt/tomcat yang akan menjalankan layanan Tomcat:

```
sudo useradd -r -m -U -d /opt/tomcat -s /bin/false tomcat
```

3. Unduh instalasi Tomcat 9. Pada saat penulisan modul ini, Versi terbaru Tomcat adalah 9.0.40. Sesuaikan versi Tomcat yang digunakan dengan melihat versi rilis terbaru pada [halaman ini](#). Gunakan wget untuk mengunduh binary Tomcat untuk Instalasi:

```
wget http://www-eu.apache.org/dist/tomcat/tomcat-9/v9.0.40/bin/apache-tomcat-9.0.  
↳40.tar.gz -P /tmp
```

4. Setelah unduhan selesai, gunakan perintah berikut untuk mengekstrak dan memindah file Tomcat pada folder / opt/tomcat:

```
sudo tar xf /tmp/apache-tomcat-9*.tar.gz -C /opt/tomcat
```

5. Untuk keperluan praktis, lebih baik apabila dibuat sebuah Symbolic Link untuk mengacu pada binary yang dimaksud:

```
sudo ln -s /opt/tomcat/apache-tomcat-9.0.40 /opt/tomcat/latest
```

6. Agar folder tomcat dimiliki oleh pengguna tomcat yang telah dibuat sebelumnya, gunakan perintah berikut:

```
sudo chown -RH tomcat: /opt/tomcat/latest
```

Demikian pula, rubah pengaturan akses pada script yang ada di dalamnya agar dapat dieksekusi:

```
sudo sh -c 'chmod +x /opt/tomcat/latest/bin/*.sh'
```

- Untuk memudahkan manajemen, Tomcat perlu dijalankan sebagai service:

```
sudo nano /etc/systemd/system/tomcat.service
```

masukkan baris berikut pada file yang dibuka:

```
[Unit]
Description=Tomcat 9 servlet container
After=network.target

[Service]
Type=forking

User=tomcat
Group=tomcat

Environment="JAVA_HOME=/usr/lib/jvm/default-java"
Environment="JAVA_OPTS=-Djava.security.egd=file:///dev/urandom -Djava.awt.
↪headless=true"

Environment="CATALINA_BASE=/opt/tomcat/latest"
Environment="CATALINA_HOME=/opt/tomcat/latest"
Environment="CATALINA_PID=/opt/tomcat/latest/temp/tomcat.pid"
Environment="CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"

ExecStart=/opt/tomcat/latest/bin/startup.sh
ExecStop=/opt/tomcat/latest/bin/shutdown.sh

[Install]
WantedBy=multi-user.target
```

- Lakukan aktivasi untuk Tomcat yang sudah diinstall:

```
sudo systemctl daemon-reload
sudo systemctl start tomcat
sudo systemctl status tomcat
```

Aktifkan tomcat pada saat mesin dimulai:

```
sudo systemctl enable tomcat
```

- Izinkan Firewall untuk Tomcat. Sama seperti pada Apache, firewall perlu dibuka untuk mengizinkan Tomcat dapat diakses:

```
sudo ufw allow 8080/tcp
```

- Buka localhost:8080 pada browser di Windows untuk memeriksa apakah Tomcat sudah berhasil dijalankan

**Apache Tomcat/9.0.40**

If you're seeing this, you've successfully installed Tomcat. Congratulations!



**Recommended Reading:**  
[Security Considerations How-To](#)  
[Manager Application How-To](#)  
[Clustering/Session Replication How-To](#)

[Server Status](#)

[Manager App](#)

[Host Manager](#)

**Developer Quick Start**

[Tomcat Setup](#)

[First Web Application](#)

[Realms & AAA](#)

[JDBC DataSources](#)

[Examples](#)

[Servlet Specifications](#)

[Tomcat Versions](#)

**Managing Tomcat**

For security, access to the [manager webapp](#) is restricted. Users are defined in:

`$CATALINA_HOME/conf/tomcat-users.xml`

In Tomcat 9.0 access to the manager application is split between different users.

[Read more...](#)

**Release Notes**

[Changelog](#)

**Documentation**

[Tomcat 9.0 Documentation](#)

[Tomcat 9.0 Configuration](#)

[Tomcat Wiki](#)

Find additional important configuration information in:

`$CATALINA_HOME/ RUNNING.txt`

Developers may be interested in:

**Getting Help****FAQ and Mailing Lists**

The following mailing lists are available:

`tomcat-announce`

Important announcements, releases, security vulnerability notifications. (Low volume).

`tomcat-users`

User support and discussion

`taglibs-user`

User support and discussion for [Apache Taglibs](#)

**Catatan**

Saat panduan ini ditulis, WSL2 belum memiliki support untuk `systemd`, sehingga perintah `sudo systemctl daemon-reload` dan seterusnya di atas akan menghasilkan error.

Untuk saat ini, berikut adalah cara yang dapat digunakan sebagai *workaround*. Metode ini diambil dari [repository github](#) berikut:

1. Install Git. Git akan digunakan untuk melakukan clone pada repository yang dimaksud:

```
sudo apt install git
```

2. Lakukan clone pada repository di atas. Sebelum menjalankan perintah ini, pastikan pengguna aktif memiliki izin akses pada folder di dalamnya:

```
git clone https://github.com/DamionGans/ubuntu-wsl2-systemd-script.git
```

3. Masuk ke dalam folder hasil clone, kemudian lakukan eksekusi pada script di dalamnya:

```
cd ubuntu-wsl2-systemd-script/
bash ubuntu-wsl2-systemd-script.sh
```

4. Tunggu sampai script selesai di eksekusi, kemudian lakukan restart pada konsol WSL (tutup jendela konsol, kemudian buka kembali)

5. Untuk menguji apakah workaround sudah berjalan, ketikkan perintah:

```
systemctl
```

apabila tidak muncul error, maka perbaikan sistem berhasil dilakukan, dan `systemctl` sudah bisa digunakan

## 1.4 Free and Open Source Software-GIS

Saat Anda sedang membaca panduan ini, barangkali dapat kita asumsikan bahwa Anda saat ini menggunakan sistem operasi Windows 10, dengan beberapa perangkat lunak GIS berbayar maupun open source di dalamnya. Pada bagian ini akan dibahas mengenai perangkat lunak ‘bebas dan terbuka’ (Free and Open Source) yang difungsikan khususnya untuk aplikasi di bidang Informasi Geospasial.

### 1.4.1 Mengenal Licensi Perangkat Lunak OpenSource

Seringkali terdapat kesalah-pahaman mengenai aplikasi Open Source. Banyak yang menganggap bahwa apabila suatu perangkat lunak diberi label ‘opensource’ maka dengan sendirinya aplikasi tersebut akan gratis alias tidak berbayar. Pada bagian ini akan dijelaskan secara singkat mengenai pengertian free dan opensource, serta bagaimana pembangun aplikasi (developer) menerapkan dan memilih lisensi perangkat lunak ini.

#### Free as in Free-Speech dan Free-Drinks

Perangkat lunak gratis atau ‘free’ mengindikasikan bahwa kita tidak perlu membayar biaya apapun pada saat menggunakan perangkat lunak ini. Meskipun demikian, tidak semua perangkat lunak gratis adalah open source. Demikian juga sebaliknya, tidak semua aplikasi yang bersifat open source maka dengan sendirinya akan menjadi gratis.

Pada bagian mengenai sejarah Linux seperti yang kita jumpai di bagian awal buku ini, terdapat pembahasan mengenai bagaimana Kernel Linux yang dikembangkan oleh Linus Torvalds kemudian dapat digunakan secara bebas oleh siapapun, dimana setiap orang bebas membuat copy, memodifikasi, merilis Linux dengan versinya sendiri, bahkan memungut biaya untuk kode yang telah dimodifikasinya (seperti pada ElementaryOS). Hal ini dimungkinkan karena Linus memberikan lisensi [GNU/GPL versi 2.0 \(GPL-2.0\)](#), sebagaimana diterbitkan oleh Free Software Foundation yang mengatur hal-hal tersebut di atas. Bagaimana jika Linus, misalnya, meminta satu dollar untuk tiap penggunaan kernel tersebut? atau ia melarang orang lain untuk mendistribusikan ulang kernel Linux yang dimilikinya, atau melakukan hal-hal lain seperti yang dapat dilakukan pada kode tersebut di atas secara cuma-cuma?

Perlu dipahami bahwa ‘Free’ tidak selalu berarti aplikasi tersebut bebas untuk digunakan apa saja. Terdapat perbedaan antara pengertian **free as in free beer** dan **free as in free speech**: **gratis** berbeda dengan **bebas**. “*Free as in free beer*” mengacu pada biaya (yaitu uang) yang harus dikeluarkan untuk menggunakan perangkat lunak tersebut, sedangkan “*free as in free speech*” mengacu pada apa yang boleh Anda lakukan dengan perangkat lunak tersebut. Dalam pengembangan perangkat lunak, dimana biaya produksi massal relatif kecil, biasanya pengembang memang menyediakan perangkat lunak tanpa biaya. Richard Stallman, perintis GNU, merupakan pelopor lisensi bebas terbuka ini, yang kemudian membuka pintu untuk pengembangan Linux oleh Linus Torvalds.

“*‘Free software’ is a matter of liberty, not price. To understand the concept, you should think of “free” as in “free speech”, not as in “free beer”.*”

(**Free Software Definition, Richard Stallman**)

Terdapat 10 faktor berbeda yang mendefinisikan mengenai penggunaan perangkat lunak open source sebagaimana dijelaskan dalam OSD (Open Source Definition):

---

#### Catatan

Bagian ini tidak diterjemahkan agar tidak merubah makna

---

1. **Free Redistribution** The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.



Fig. 1.12: “Think Free Speech, not Free Beer”

2. **Source Code** The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost, preferably downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.
3. **Derived Works** The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.
4. **Integrity of The Author's Source Code** The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.
5. **No Discrimination Against Persons or Groups** The license must not discriminate against any person or group of persons.
6. **No Discrimination Against Fields of Endeavor** The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.
7. **Distribution of License** The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.
8. **License Must Not Be Specific to a Product** The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.
9. **License Must Not Restrict Other Software** The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.
10. **License Must Be Technology-Neutral** No provision of the license may be predicated on any individual technology or style of interface.

Perbedaan penggunaan definisi dari lisensi tersebut memberikan ruang bagi pengembang untuk memilih lisensi yang sesuai dengan kebutuhannya, sebagaimana penjelasan di bawah ini:

### Lisensi Perangkat Lunak

Sebagaimana disebutkan di atas, pengembang aplikasi menggunakan lisensi untuk mendefinisikan batasan apa yang boleh dan tidak boleh dilakukan terhadap perangkat lunak yang mereka buat. *Lisensi open source* adalah lisensi yang sesuai dengan Open Source Definition - singkatnya, lisensi tersebut memungkinkan perangkat lunak untuk digunakan, dimodifikasi, dan dibagikan secara bebas. Untuk disetujui oleh Open Source Initiative (juga dikenal sebagai OSI), lisensi harus melalui proses peninjauan lisensi oleh [Open Source Initiative](#).

Para pembuat aplikasi (*developer*) memberikan lisensi sebagai bagian yang tidak dapat dipisahkan dari kode sumber (*source code*) aplikasi yang mereka buat. Pada beberapa lisensi, aturan terhadap *derived works*, misalnya, dibatasi sehingga produk turunan tidak boleh menggunakan lisensi yang lebih ketat dibanding lisensi yang diberikan. Sebagai contoh, dokumen ini menggunakan lisensi [CC-BY-SA-4.0](#) sebagaimana disebutkan pada halaman depan. Ini berarti, sebagian atau keseluruhan modul ini, berikut turunannya, **dilarang untuk digunakan secara komersial**.

Berikut adalah rangkuman lisensi perangkat lunak dan batasan masing-masing:

---

### Catatan

						
Type	Permissive	Permissive	Permissive	Copyleft	Copyleft	Copyleft
Provides copyright protection	✓ TRUE	✓ TRUE	✓ TRUE	✓ TRUE	✓ TRUE	✓ TRUE
Can be used in commercial applications	✓ TRUE	✓ TRUE	✓ TRUE	✓ TRUE	✓ TRUE	✓ TRUE
Provides an explicit patent license	✓ TRUE	✗ FALSE	✗ FALSE	✗ FALSE	✗ FALSE	✗ FALSE
Can be used in proprietary (closed source) projects	✓ TRUE	✓ TRUE	✓ TRUE	✗ FALSE	✗ FALSE partially	✗ FALSE for web
Popular open-source and free projects	Kubernetes Swift Firebase	Django React Flutter	Angular.js JQuery, .NET Core Laravel	Joomla Notepad++ MySQL	Qt SharpDevelop	SugarCRM Launchpad

Fig. 1.13: Perbandingan Lisensi Perangkat Lunak FOSS

Terdapat perbedaan antara lisensi perangkat lunak dan lisensi dokumen. Lisensi perangkat lunak seperti disebutkan di atas membatasi penggunaan khusus perangkat lunak, kode sumber dan (mungkin) dokumentasi yang menyertainya. Lisensi buku terbuka seperti disebutkan untuk dokumen ini berlaku pada naskah tertulis.

---

Beberapa lisensi Open Source yang populer adalah sebagai berikut:

- Apache License 2.0
- BSD 3-Clause “New” or “Revised” license
- BSD 2-Clause “Simplified” or “FreeBSD” license
- GNU General Public License (GPL)
- GNU Library or “Lesser” General Public License (LGPL)
- MIT license
- Mozilla Public License 2.0
- Common Development and Distribution License
- Eclipse Public License version 2.0

Website berikut memberikan perbandingan dan panduan yang dapat digunakan untuk memilih perangkat lunak mana yang ingin Anda gunakan untuk perangkat lunak Anda tersebut. Website ini juga dapat menjadi panduan pada saat menggunakan perangkat lunak *open source*.

<https://choosealicense.com/licenses/>

---

### Catatan

Pembahasan ini seharusnya telah memberikan gambaran kepada Anda mengenai jawaban pertanyaan, seperti, “*Se-andainya saya membuat QGIS yang saya distribusikan dengan nama saya sendiri, apakah saya diizinkan?*”

---

### 1.4.2 Sekilas Tentang Perangkat FOSS-GIS

Perangkat lunak FOSS (Free and Open Source) dapat dijumpai dalam berbagai bentuk, mulai kernel (seperti Linux), Sistem Operasi, perangkat lunak pengolah kata (*word processor*), analisis data, hingga *machine learning* dan analisis *big data*. Demikian pula, pada bidang Sistem Informasi Geografis, terdapat banyak aplikasi free and open source yang dapat digunakan untuk menggantikan dan/atau melengkapi perangkat lunak *proprietary* yang digunakan saat ini.

Berbagai perangkat Lunak ini (kita sebut saja sebagai perangkat **FOSS-GIS**, *Free and Open Source GIS*) tersedia untuk berbagai keperluan di bidang SIG, mulai dari akuisisi data, analisis, penyimpanan data spasial hingga visualisasi pada media desktop dan web. Dapat dikatakan bahwa ekosistem aplikasi FOSS-GIS telah sedemikian matang sehingga hampir tiap aplikasi berbayar untuk keperluan GIS dapat diperoleh satu atau beberapa aplikasi open source yang dapat menggantikannya.

FOSS-GIS tidak hanya tentang perangkat lunak. Sebagaimana disebutkan, ekosistem Open Source pada proses bisnis geospasial meliputi berbagai bidang, [diantaranya](#):

- **Open Data** menerapkan prinsip bebas dan terbuka untuk data geospasial. Prinsip bebas dan terbuka ini lebih dari sekadar menerbitkan informasi dengan akses gratis. Data terbuka menyediakan mekanisme bagi peserta untuk memberikan kontribusi sebagai mitra yang setara dalam pengumpulan dan peninjauan terhadap data.
- **Open Education** menerapkan prinsip-prinsip open source untuk pembuatan bahan ajar yang memungkinkan organisasi untuk berbagi bahan silabus mengurangi biaya dan menjangkau khalayak yang lebih luas.



Fig. 1.14: Beberapa perangkat lunak FOSS-GIS

- **Open Science** menggabungkan ide-ide ini untuk mewujudkan tujuan penelitian dengan lebih baik, dengan berbagai data yang digunakan untuk mendukung kesimpulan, di samping perangkat lunak sopensource yang digunakan untuk analisis.
- **Open Standard** mempromosikan interoperabilitas antara aplikasi, organisasi, dan bidang usaha. Standar terbuka adalah alat utama yang memungkinkan praktisi geospasial untuk bekerja sama, dengan manfaat tambahan untuk menghindari penguncian teknologi (*vendor-locked*).

Beberapa perangkat lunak FOSS-GIS yang populer antara lain:

- QGIS
- Grass GIS
- Orfeo Toolbox
- GDAL/OGR
- Geoserver
- PostGIS
- Berbagai Library dalam berbagai bahasa (Python, Java, Javascript, etc.), seperti Geopandas, Numpy, Proj4, LeafletJS, dst (*the list is too long!*)

[Link ini](#) berisi daftar panjang berbagai aplikasi, data dan tutorial tentang perangkat lunak GIS, baik yang berbayar maupun khususnya yang bersifat FOSS. Anda perlu meluangkan waktu yang tidak sebentar untuk menjelajah seluruh daftar yang tersedia.

### Komunitas FOSS-GIS

Seperti yang dapat kita lihat pada Linux dan proyek Open Source lainnya, pengguna dan komunitas adalah kekuatan utama berkembangnya perangkat lunak FOSS. Dengan lisensi yang bebas dan terbuka, pengembangan aplikasi menjadi optimal karena banyak yang berkontribusi dalam membagikan *bug*, menyumbang ide pengembangan, maupun menyumbang dana untuk pengembangan aplikasi (Konsepsi yang dikenal sebagai Hukum Linus, “*Given enough eyeballs, all bugs are shallow*”). Diantara yang dapat diamati pada perangkat lunak berbasis Open Source adalah pengembangannya yang luar biasa cepat. Versi terbaru dari Ubuntu dirilis tiap 6 (tiga) bulan sekali, dengan versi LTS tiap dua tahun sekali. Pada tiap rilis, dapat dijumpai versi baru, bug fix, berbagai fitur dan modul yang merupakan masukan dari pengguna serta hasil pengembangan secara terus menerus. Platform seperti Github dan GitLab memungkinkan pengguna dan pengembang untuk bertukar ide sambil mengembangkan aplikasi sebagai bagian dari komunitas.

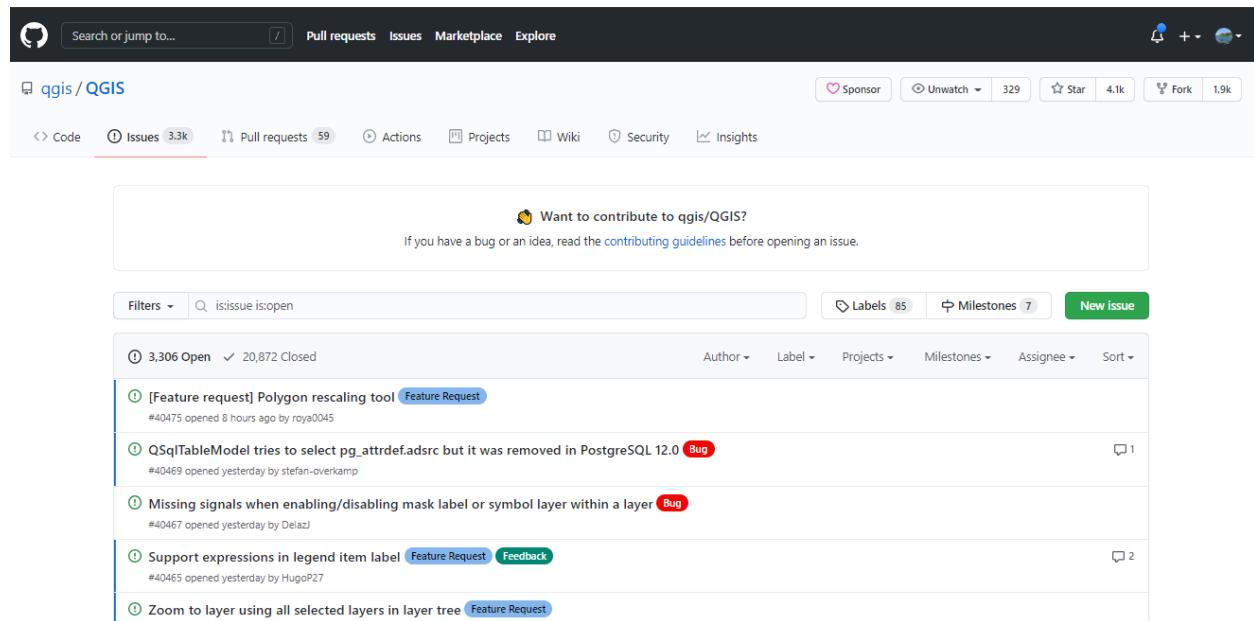


Fig. 1.15: Halaman Issues pada Github QGIS. Dapat dilihat beberapa *Feature Request* dan *Bug* yang diajukan oleh komunitas

Sebagai sebuah komunitas matang dengan sejarah yang dapat dilacak hingga awal tahun 80an melalui MOSS (*Map Overlay and Statistical System*) dan Grass GIS (*Geographic Resources Analysis Support System*). Komunitas FOSS-GIS berkembang dengan adanya kebutuhan akan aplikasi GIS yang mampu menangani berbagai fungsi yang semakin kompleks seiring perkembangan teknologi komputasi. *Open Grass Fundation* (OGF) dimulai sejak tahun 1992, dan menjadi cikal bakal dari OGC (*Open Geospatial Consortium*) yang merupakan lembaga standarisasi internasional untuk data dan informasi geospasial.

OSGeo merupakan lembaga yang menjadi pendorong utama komunitas FOSS-GIS saat ini. Open Source Geospatial Foundation (OSGeo) adalah organisasi nirlaba yang misinya adalah untuk mendorong adopsi global teknologi geospasial terbuka dengan menjadi yayasan perangkat lunak inklusif yang dikhususkan untuk filosofi terbuka dan pembangunan berbasis masyarakat partisipatif. Di bawah payung *OSGeo Projects*, berbagai aplikasi open source GIS memiliki wadah untuk tumbuh dan berkembang hingga menjadi penggerak utama di berbagai aspek di bidang geospasial.

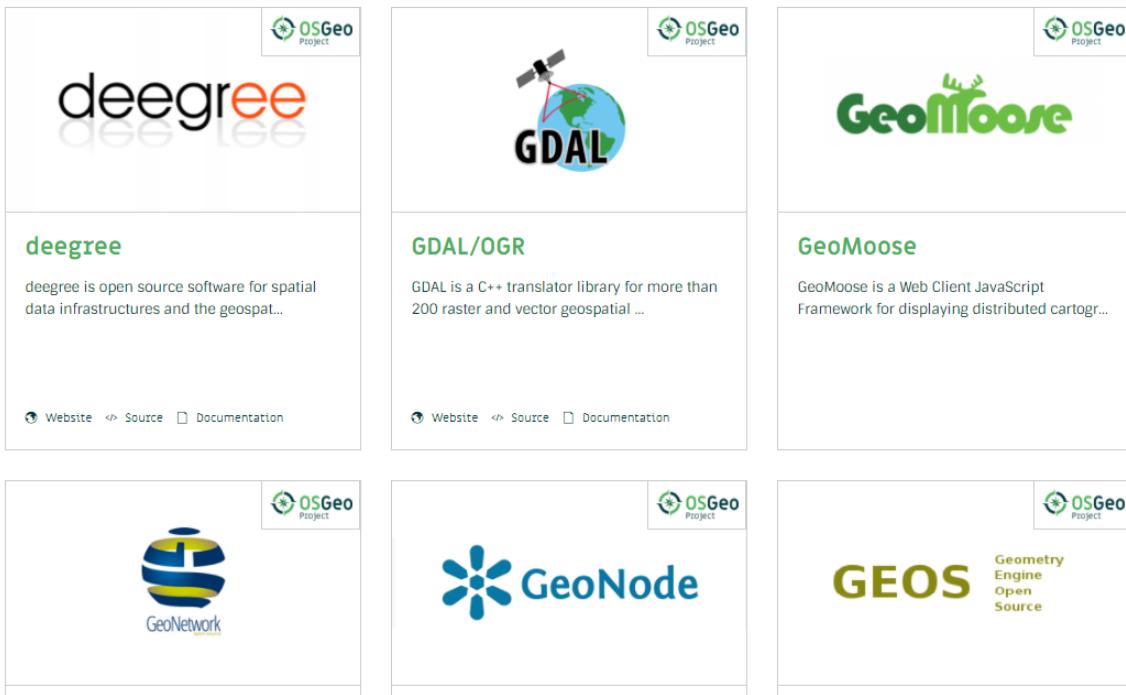


Fig. 1.16: Beberapa aplikasi yang berada di bawah payung OSGeo Projects

### Latihan: Menggunakan OSGeo Live

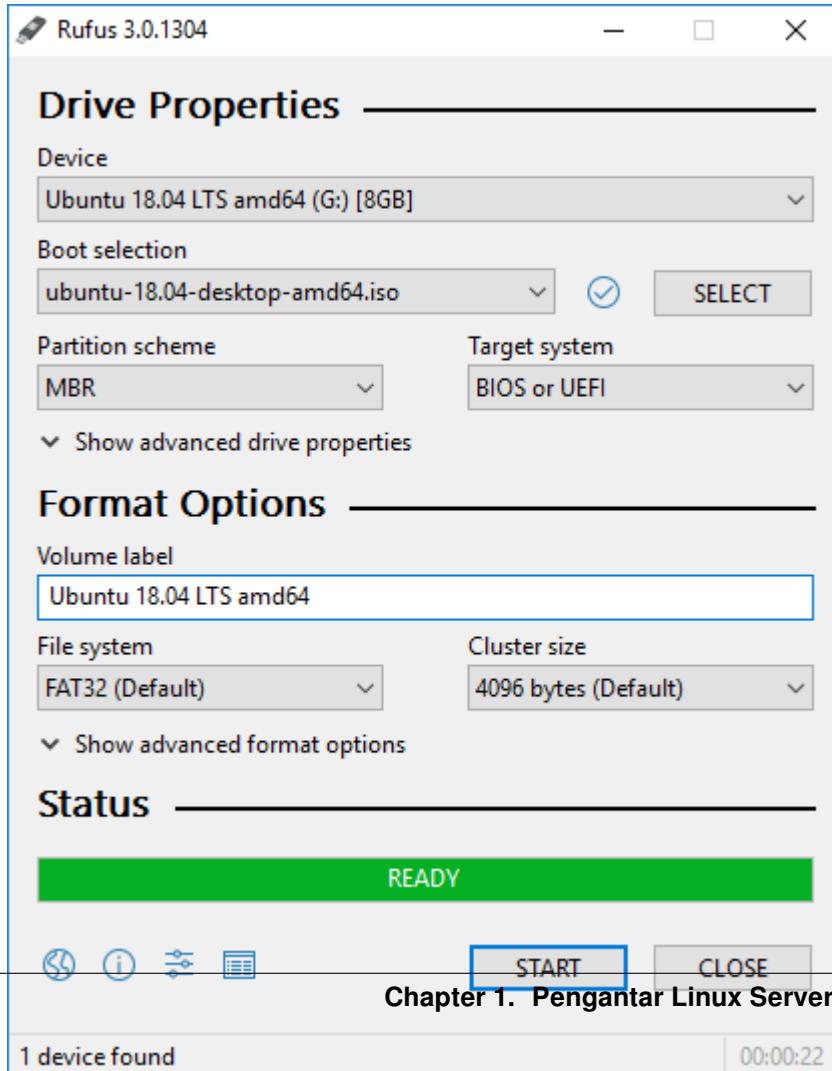
Dengan banyaknya aplikasi Open Source yang tersedia (bahkan hanya yang di bawah payung OSGeo saja sudah sangat banyak!), tentu untuk mencoba instalasi satu-persatu aplikasi tersebut akan memakan waktu. OSGeo telah menyediakan distro khusus yang telah diisi dengan semua aplikasi OSGeo. [OSGeo Live](#) merupakan distro khusus berbasis *Lubuntu* yang telah dilengkapi dengan beragam aplikasi pada OSGeo Project:

Sebagaimana distro Linux lainnya, OSGeoLive dapat digunakan sebagai sebuah [Linux Live](#) yang dipasang pada DVD maupun USB pada saat Boot. Berbeda dengan instalasi sistem operasi pada umumnya, Linux Live tidak akan meninggalkan jejak pada harddisk di komputer tempat ia dijalankan. Seluruh proses sistem operasinya akan diajalan dari DVD dan USB, dan begitu selesai, sistem berikut data yang dibuat juga akan hilang. Pada latihan ini akan dibuat sebuah mesin Linux Live menggunakan distro OSGeo Live.

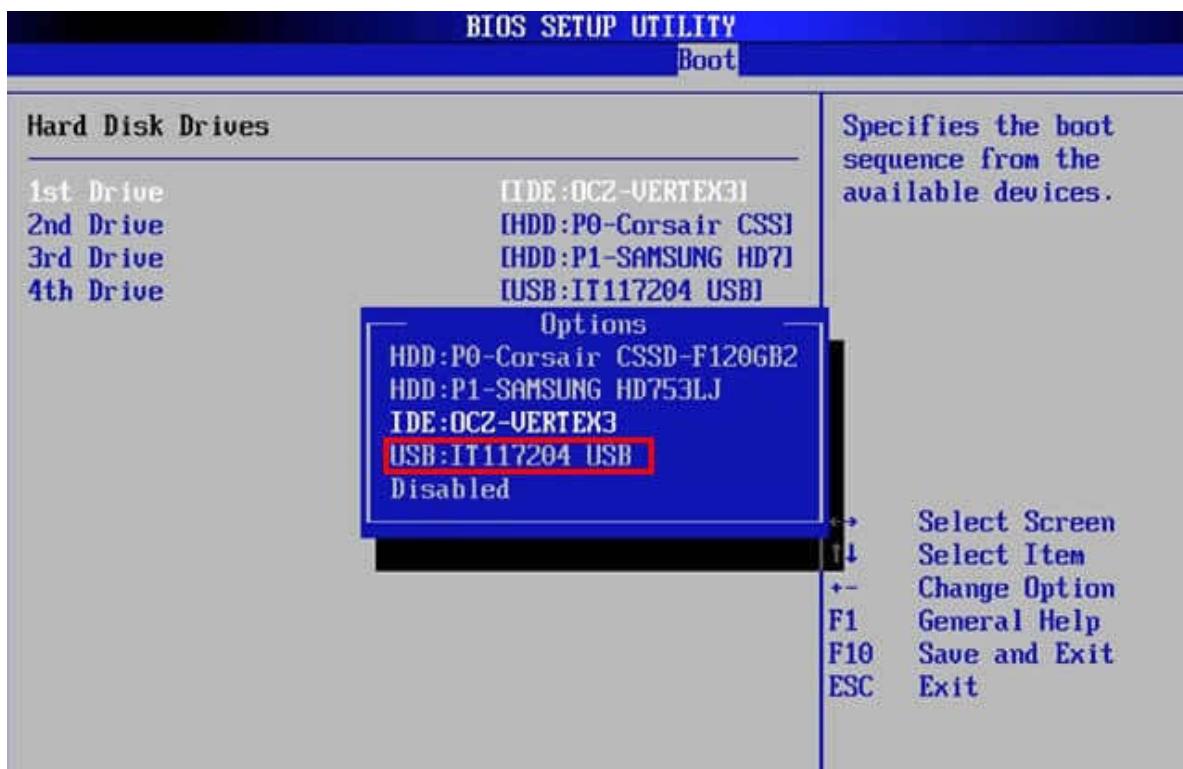
1. Langkah pertama adalah mengunduh perangkat lunak yang dapat digunakan untuk membuat Linux Live USB. Diantara perangkat yang dapat digunakan adalah [LinuxLiveUSB](#) dan [Rufus](#). Pada latihan ini akan diunduh **Rufus** untuk membuat Live OSGeo pada USB. Unduh Rufus dari alamat web rufus di atas atau [riles terbaru pada Github](#).
2. Unduh ISO dari OSGeo Live (**osgeolive iso**) pada alamat <https://live.osgeo.org/en/download.html>. Perlu diingat bahwa ukuran unduhan adalah sekitar 4 GB, jadi pastikan bahwa Anda memiliki koneksi internet yang memadai
3. Buka Rufus, kemudian masukkan **USB Flashdisk** kosong yang akan digunakan sebagai Bootable USB. Rufus akan mendeteksi secara otomatis USB Flashdisk yang akan digunakan.
4. Selanjutnya, pada bagian **Boot Selection** masukkan **ISO OSGeo Live** yang sudah diunduh sebelumnya. Biarkan opsi lain dengan pengaturan *default*, kemudian klik **Start**.



Fig. 1.17: Beberapa aplikasi pada OSGeo Live



5. Setelah proses selesai, lakukan restart Komputer. Pastikan bahwa Laptop atau Komputer Anda menggunakan USB sebagai **prioritas Boot pertama**. Pengaturan untuk ini dilakukan melalui BIOS, dan memiliki bentuk yang berbeda untuk tiap jenis BIOS. Pengaturan ini biasanya dapat dijumpai pada menu **Boot Priority**. Silahkan mengacu pada dokumentasi tiap BIOS untuk penjelasan lebih lanjut.



6. Tunggu sampai GRUB (Boot Manager Linux) memanggil OSGeo. Setelah selesai, Linux akan dimulai. Gunakan OSGeo Live seperti menggunakan sistem operasi seperti biasa



Terdapat banyak perangkat lunak FOSS-GIS pada OSGeo Live, baik yang berupa aplikasi maupun library seperti Geopan-

das. OSGeo Live memungkinkan kita untuk mencoba aplikasi tersebut tanpa perlu menginstall apapun pada komputer kita.

### 1.4.3 Menggunakan Perangkat Lunak OpenSource untuk GIS

Sepanjang bagian ini kita telah membahas beberapa aspek penggunaan perangkat lunak FOSS-GIS, seperti biaya yang rendah, pengembangan fitur dan rilis yang terjadwal, banyaknya pilihan yang dapat digunakan, komunitas yang aktif, serta berbagai organisasi yang mengatur standarisasi dan dukungan penuh pada pengembangan aplikasi GIS berbasis open source. Meskipun demikian, penggunaan perangkat FOSS-GIS di Indonesia sendiri tergolong masih rendah. Salah satu sebabnya adalah kepercayaan terhadap perangkat berbiaya rendah (atau gratis) masih kurang, disamping juga kurangnya SDM yang mampu mengembangkan dan mengadopsi penggunaan perangkat lunak FOSS-GIS di berbagai bidang.

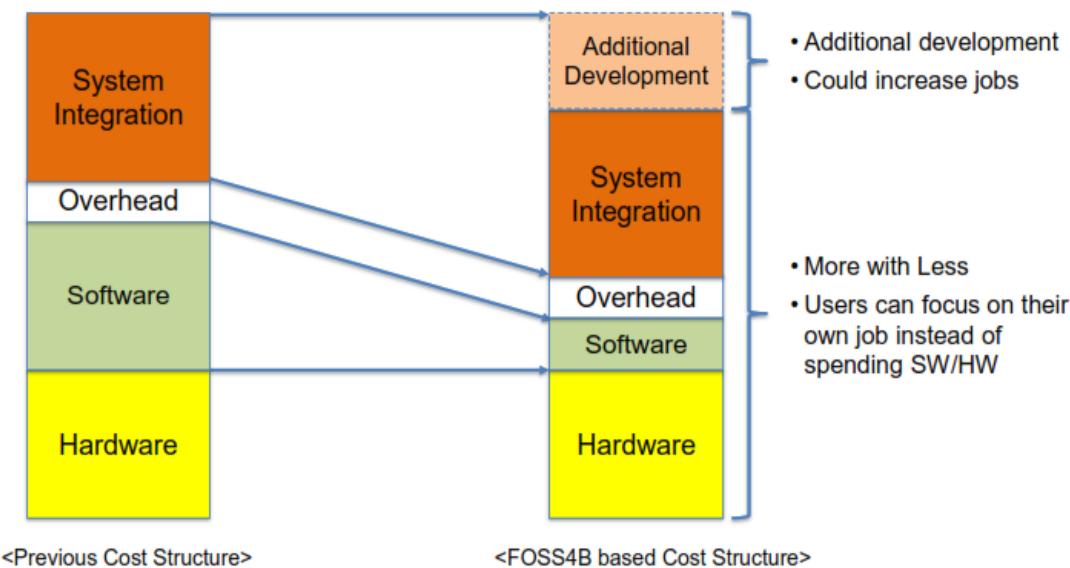


Fig. 1.18: Keuntungan dari adopsi perangkat lunak Open Source (Sumber Gambar: Shin (2014))

Pengembangan perangkat lunak open source pada berbagai instansi sebenarnya dapat lebih menguntungkan dalam jangka panjang dibandingkan dengan penggunaan perangkat lunak berbayar. Penggunaan perangkat lunak Open source mungkin membutuhkan kemampuan dan kurva belajar yang cukup terjal, tetapi dengan pelatihan dan pengembangan secara terus-menerus, pada gilirannya instansi akan berkembang dengan sendirinya, karena berbagai masalah pada perangkat lunak akan dapat diselesaikan sendiri. Selain itu, dukungan komunitas baik lokal maupun internasional untuk perangkat lunak open source dapat dijumpai dengan mudah. Terakhir, lembaga pemerintah maupun swasta dapat ‘memesan’ fitur tertentu pada perangkat lunak Open Source dengan menjadi sponsor pengembangan fitur yang diminta, sehingga disamping memperoleh keuntungan untuk lembaga itu sendiri, dengan sendirinya juga berkontribusi pada komunitas dan masyarakat umum.

Disamping itu, penggunaan perangkat lunak open source menjamin dukungan pada standar yang didukung secara internasional. Hal ini terutama diperlukan seiring dengan kebijakan satu peta yang sedang didorong oleh pemerintah, dimana tiap wali data memiliki kewajiban untuk menyediakan infrastruktur berbagi-pakai data geospasial. Penggunaan perangkat lunak open source akan menjamin interoperabilitas (misalnya melalui standar OGC), dan menghindari *vendor lock* pada beberapa perangkat lunak tertentu.

## PENGANTAR BASISDATA RELATIONAL

Basisdata merupakan salah satu elemen penting pada aplikasi Enterprise GIS. Dengan basisdata, sebuah sistem dapat didesain untuk digunakan oleh multi-user, dengan hak akses untuk tiap pengguna yang berbeda-beda. Demikian pula, basisdata memungkinkan data dalam jumlah besar untuk disimpan pada satu lokasi penyimpanan yang aman, memungkinkan analisis data secara cepat serta menjamin keselamatan data apabila terjadi masalah pada data tersebut.

Bagian ini membahas mengenai:

*Pengantar mengenai Basisdata Relational* : Konsep basisdata relational serta instalasi basisdata PostgreSQL dan PostGIS dengan menggunakan Docker

*Administrasi pengguna pada Basisdata* : Penjelasan tentang pengaturan multi-user pada Basisdata PostgreSQL (User, Roles dan Privileges)

*Basisdata Spasial dengan PostGIS* : Bagaimana menggunakan ekstensi PostGIS untuk menginput data spasial, melakukan analisis serta visualisasi data spasial dan menghubungkan basisdata dengan QGIS

### 2.1 Basisdata Relational

Basisdata relational merupakan salah satu model basisdata pada DBMS (Database Management System). Model Relational banyak digunakan sejak tahun 90an hingga saat ini karena berbagai kelebihannya dalam memanajemen data dalam jumlah besar. Pada bagian ini akan dibahas mengenai basisdata relasional, khususnya PostgreSQL.

#### 2.1.1 Pemodelan data pada Basisdata Relasional

Kita ambil sebuah contoh untuk kasus penanggulangan kebencanaan: anda ditugaskan oleh pimpinan anda untuk melakukan inventarisasi data yang diperoleh dari hasil pemetaan akibat bencana banjir di suatu daerah. Anda misalnya berurusan dengan data-data jumlah pengungsian dari masing-masing desa, kerusakan pada tiap desa, lokasi kantung-kantung pengungsian, kebutuhan tiap kepala, dan lain sebagainya. Apabila dinyatakan dalam bentuk tabel, data-data tersebut mungkin tampak seperti berikut:

*Tabel Data Desa*

Nomor Induk Desa	Nama Desa	Kecamatan	Kabupaten	Jumlah penduduk	Luas Wilayah (m <sup>2</sup> )
10007	Sukaluyu	Nanggung	Bogor	7970	4353824
10008	Hambaro	Nanggung	Bogor	5947	3248530
10009	Kalong Liud	Nanggung	Bogor	6468	3531915
20001	Purasari	Leuwiliang	Bogor	47553	25944353
20002	Puraseda	Leuwiliang	Bogor	23941	13061263

Tabel Posko Pengungsi

No Registrasi Posko	Nama Posko	Kapasitas posko	Jumlah relawan
12912	A	2300 orang	72
12913	B	320 orang	23
12914	C	500 orang	29
12915	D	1200 orang	40

Tabel Asal Pengungsi

ID	Posko	Asal Pengungsi
001	12912	10007
002	12912	10009
003	12912	20002
004	12913	20001

Apabila suatu ketika Anda ditanya, ‘Dari mana saja asal pengungsi di Posko A?’, apa jawaban yang akan anda berikan? Apa yang akan anda lakukan untuk memperoleh jawaban yang diinginkan? Pertama-tama, Anda mungkin akan melihat pada masing-masing tabel, kemudian mencari informasi yang sesuai pada tiap tabel tersebut. Tentunya hal ini mudah dilakukan apabila data yang disediakan cukup sedikit. Bagaimana jika data yang ada berjumlah ratusan atau bahkan mungkin ribuan data?

Di sinilah peranan **Basisdata**. Sebuah sistem basisdata menghubungkan kumpulan data menjadi sebuah sistem yang terintegrasi. Dapat dikatakan, basisdata merupakan kumpulan dari tabel-tabel yang saling terhubung. Dengan menggunakan basisdata, Anda dapat mengambil data dari suatu tabel berdasarkan data dari tabel yang lain.

Gambaran umum sebuah basisdata adalah *sekumpulan tabel yang saling terhubung*. Hubungan antar tabel ini dapat dilakukan dengan berbagai macam cara. Salah satunya adalah dengan menggunakan cara yang Anda lihat pada contoh tabel-tabel di atas.

## 2.1.2 Pengantar Basisdata Relational

**Sistem Manajemen Basis Data Relasional (RDBMS)** adalah kumpulan program dan fungsi berbasis tabel yang menyediakan antarmuka antara pengguna dan aplikasi dan basis data, memberikan cara sistematis untuk membuat, memperbarui, menghapus, mengelola, dan mengambil data. Sebagian besar sistem manajemen database relasional menggunakan bahasa pemrograman SQL untuk mengakses database dan banyak yang mengikuti properti *ACID* (*Atomicity, Consistency, Isolation, Durability*) dari database:

- **Atomicity:** Jika ada pernyataan dalam transaksi yang gagal, seluruh transaksi gagal dan database dibiarkan tidak berubah.
- **Consistency:** Transaksi harus memenuhi semua protokol yang ditentukan oleh sistem - tidak ada transaksi yang diselesaikan sebagian.
- **Isolation:** Tidak ada transaksi yang memiliki akses ke transaksi lain yang belum selesai. Setiap transaksi bersifat independen.
- **Durability:** Setelah transaksi dilakukan, transaksi akan tetap dilakukan melalui penggunaan log transaksi dan cadangan.

c

Basisdata relasional menyimpan data (entitas dunia nyata) dalam bentuk tabel serta menyediakan cara yang efisien, intuitif, dan fleksibel untuk menyimpan dan mengakses informasi pada tabel tersebut secara terstruktur. Tabel pada basisdata relasional juga dikenal sebagai **relasi**, terdiri dari kolom yang berisi satu atau lebih kategori data; dan baris, juga dikenal sebagai **record**, berisi sekumpulan data yang ditentukan oleh kategori.

Aplikasi Database Management System/DBMS (seperti PostgreSQL) mengakses data dengan menentukan **kueri**, yang menggunakan operasi untuk mengidentifikasi atribut, memilih *tuples*, dan menentukan relasi pada data. Model relasional untuk manajemen basis data dikembangkan oleh ilmuwan komputer IBM Edgar F. Codd pada tahun 1970.

### Beberapa istilah pada Basisdata Relasional

Sebagai review, berikut adalah beberapa istilah yang dikenal pada basisdata relasional:

- **Tabel:** Merupakan struktur penyimpanan dasar dari basis data objek relasional, terdiri dari satu atau lebih kolom (column) dan nol atau lebih baris (row).
- **Row (baris):** Baris merupakan kombinasi dari nilai-nilai kolom dalam tabel; sebagai contoh, informasi tentang suatu departemen pada tabel ‘Departemen’. Baris seringkali disebut dengan record atau tuple.
- **Column (kolom):** Kolom menggambarkan jenis data pada tabel; Kolom di definisikan dengan nama kolom dan tipe data beserta panjang data tertentu.
- **Field:** Field merupakan pertemuan antara baris dan kolom. Sebuah field menggambarkan satu keterangan mengenai data. Jika pada suatu field tidak terdapat data, maka field tersebut dikatakan memiliki nilai “null” atau kosong.
- **Primary key:** Primary key atau kunci utama merupakan kolom atau kumpulan kolom yang secara unik membedakan antara baris yang satu dengan lainnya; Kolom yang merupakan primary key ini tidak boleh mengandung nilai “null”, dan nilainya harus unik (berbeda antara baris satu dengan lainnya).
- **Foreign key:** Foreign key atau kunci tamu merupakan kolom atau kumpulan kolom yang mengacu ke primary key pada tabel yang sama atau tabel lain. Foreign key ini dibuat untuk memaksakan aturan-aturan relasi pada basis data. Nilai data dari foreign key harus sesuai dengan nilai data pada kolom dari tabel yang diacunya atau bernilai “null”.
- **Relasi** merupakan interaksi antara tabel yang menunjukkan hubungan antara objek-objek yang ada di dunia nyata. Relasi menyatakan hubungan antar tabel sebagai model dari dunia nyata.

## Table also called Relation

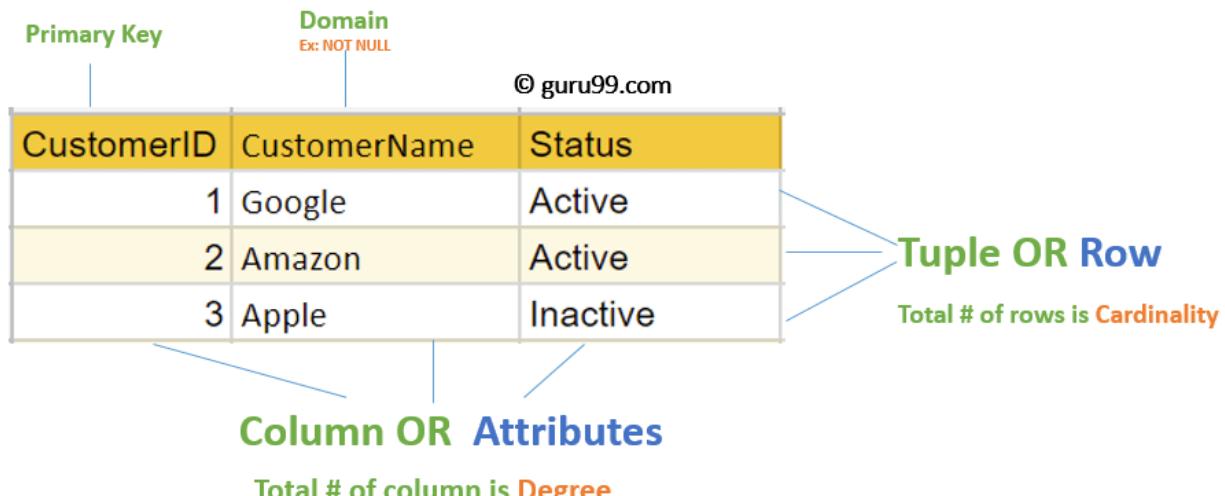
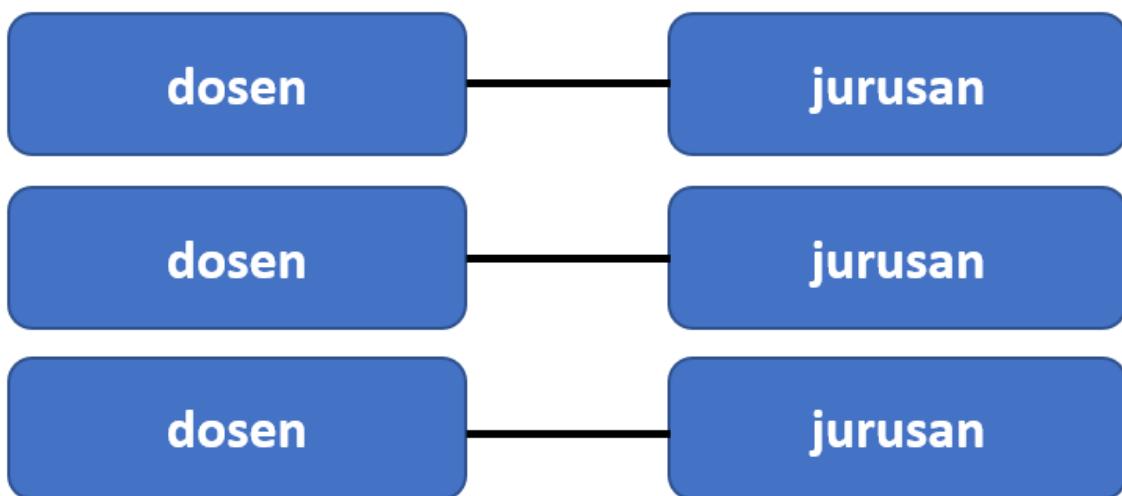


Fig. 2.1: Tabel atau relasi pada Basisdata Relasional

### Relasi pada basisdata relasional

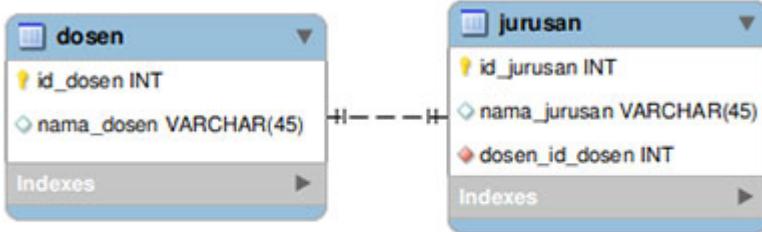
Jika kita pahami bahwa entitas dunia nyata dimodelkan sebagai sebuah **relasi** (atau tabel) pada basisdata relasional, maka hubungan antar entitas tersebut di dunia nyata juga dapat kita modelkan. Pada sebuah basisdata relasional, hubungan antar entitas dapat disederhanakan menjadi tiga bentuk saja: One to One, One to Many dan Many to Many. Berikut adalah model relasi pada sebuah basisdata relasional<sup>1</sup> tersebut:

1. **One to One (1 to 1)** Relasi database model ini terjadi apabila sebuah data terdapat pada 2 buah tabel, dan hanya diperbolehkan satu data saja pada masing masing tabel (*unique record*), sama halnya seperti *primary key*, record yang ada pada model ini tidak boleh ada yang sama.



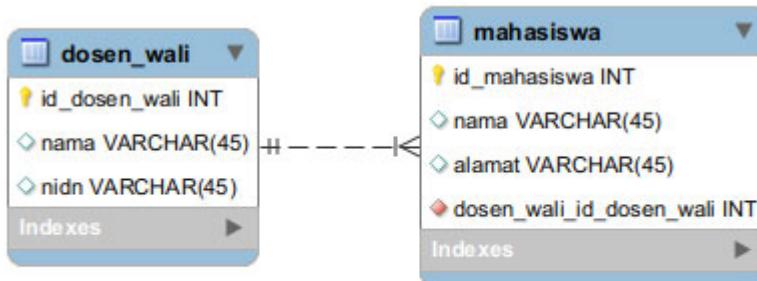
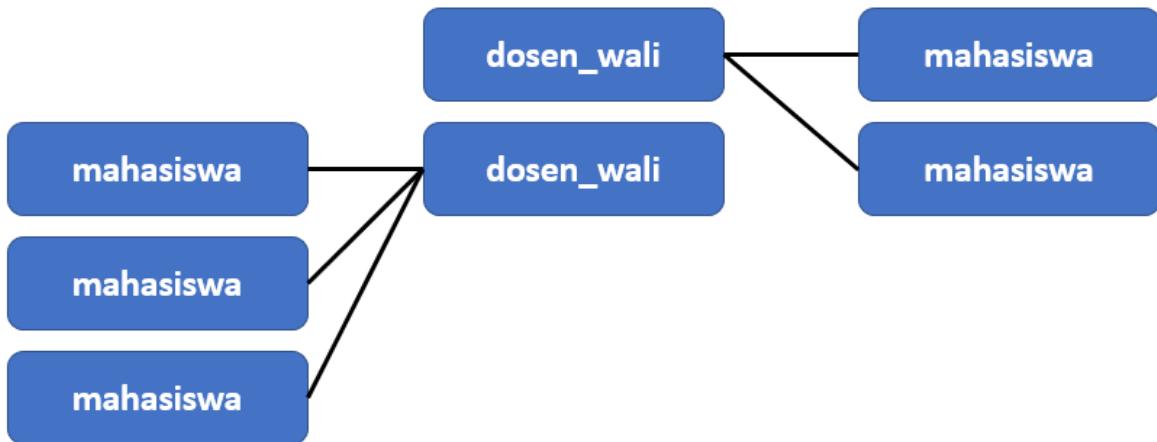
Dalam bentuk tabel relasional, hasilnya adalah seperti ini:

<sup>1</sup> <https://aantamim.id/relasi-tabel-database/>



## Relasi One to One

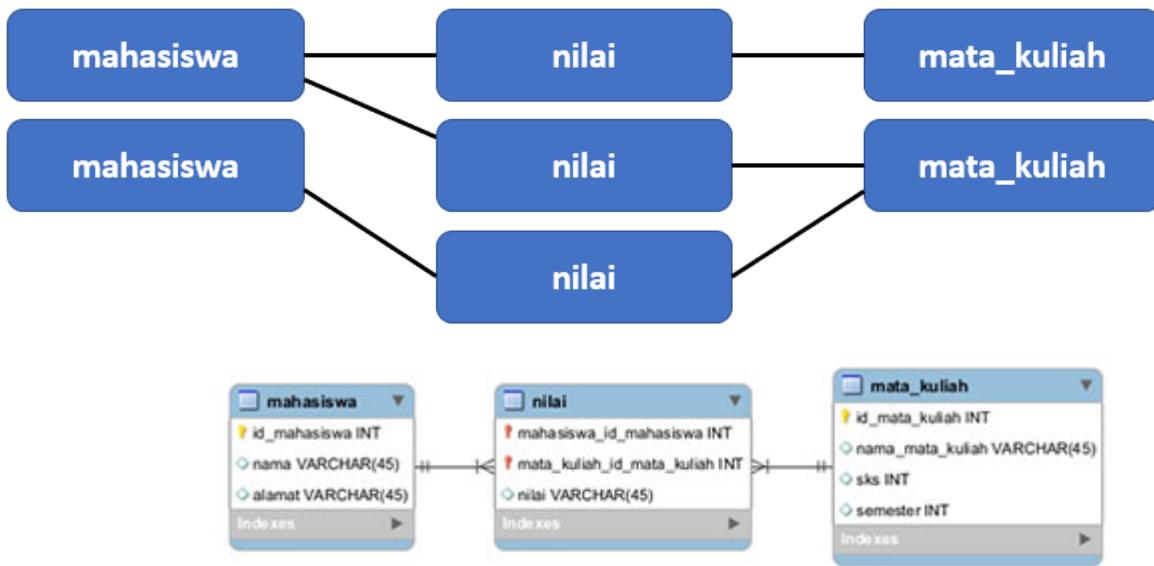
2. **One to Many (1 to n)** Relasi database model ini membolehkan data yang sama pada tabel kedua, tapi hanya membolehkan data yang bersifat unique (unik) pada tabel pertama. Jadi pada model tabel kedua boleh memiliki beberapa data yang sama.



## Relasi One to Many

Dalam bentuk tabel:

3. **Many to many (n to m)** Berbeda dengan kedua model diatas, relasi database model ini membolehkan beberapa data yang sama baik pada tabel pertama maupun tabel kedua. Dengan demikian tidak ada unique record di kedua tabel tersebut.



## Relasi Many to Many

Apabila kita gambarkan hubungan yang kompleks antar semua entitas pada basisdata, kita akan memperoleh sebuah diagram yang disebut dengan **Diagram Entity Relationship (Diagram ER)**.

### Operasi pada Basisdata Relasional

Apabila kita menganggap basisdata sebagai sebuah kumpulan dari entitas yang saling terhubung, maka seperti halnya sebuah bilangan matematika, kita dapat menggunakan berbagai operasi pada entitas ini untuk memperoleh jawaban dari pertanyaan yang kita ajukan. Codd sendiri awalnya mendefinisikan delapan operasi berbeda, yang kemudian disebut dengan **Relational Algebra**. Relational Algebra merupakan bahasa prosedural yang menggunakan operator *unary* atau *binary* pada relasi atau tabel. Hasil dari Aljabar relasional ini adalah sebuah tabel lain yang menggambarkan hubungan dari kedua entitas.

Pada sebuah RDBMS, operasi aljabar relasional ini diwujudkan menggunakan bahasa SQL, seperti yang akan dibahas sebagai berikut.

### 2.1.3 Structured Query Language (SQL)

Apabila Anda menggunakan komputer untuk membuat basisdata pada contoh di awal bab ini, Anda akan dengan cepat menjawab pertanyaan yang diajukan sebelumnya. Apabila Anda mencari: ‘*Darimana saja asal pengungsi di posko A?*’, maka komputer akan langsung melihat tabel **Posko Pengungsi** untuk melihat nomor identifikasi masing-masing posko pengungsian, kemudian mencocokkannya dengan tabel **Asal Pengungsi** untuk melihat Desa apa saja yang ada di sana. Sekarang, bagaimana kita menanyakan pertanyaan seperti ini kepada komputer?

Bagaimana kita membuat suatu sistem basisdata dan mengajukan pertanyaan (pertanyaan=‘QUERY’) kepada komputer agar kita memperoleh data yang kita inginkan? Untuk itulah kita akan mempelajari mengenai bahasa **SQL (Structured Query Language)**. Bahasa inilah yang akan kita gunakan untuk ‘bertanya’ atau mengajukan ‘query’ kepada komputer. Kita juga menggunakan bahasa SQL untuk menyelesaikan berbagai macam operasi pada aljabar relasional yang kita gunakan untuk mendapatkan jawaban dari pertanyaan yang kita inginkan.

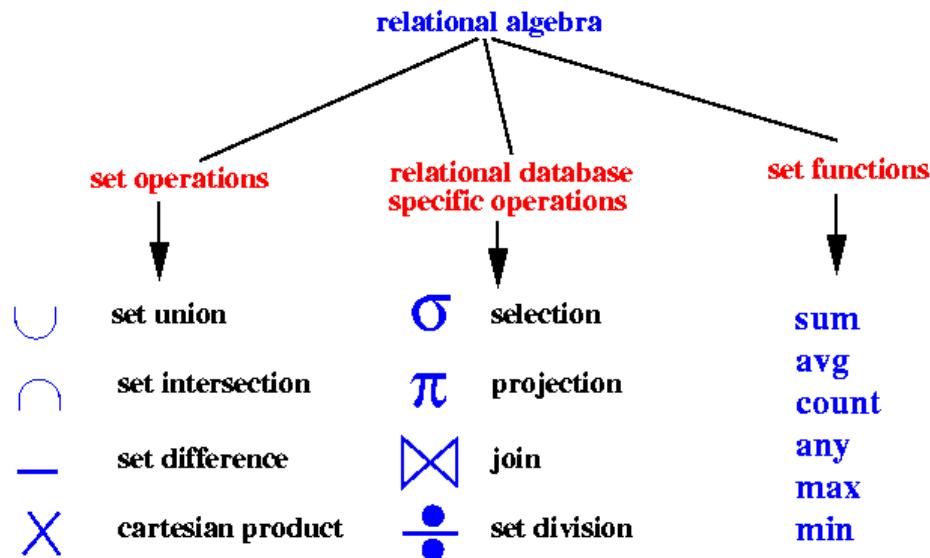


Fig. 2.2: Relational Algebra pada Basisdata Relational

### Bahasa SQL

SQL merupakan serangkaian perintah yang kita gunakan untuk membuat dan melakukan manajemen terhadap suatu basisdata di komputer. Misalnya, untuk membuat sebuah tabel seperti ini:

Tabel Anggota

No_id	Nama	Jenis_kelamin
21451	Budi Darmawan	L
21569	Anggi Lestari	P
21584	Joko Anantoko	L

Dalam bahasa SQL akan tertulis seperti ini:

```
CREATE TABLE Anggota
(
  No_id INTEGER NOT NULL,
  nama VARCHAR(20),
  jenis_kelamin VARCHAR(2)
);
```

(continues on next page)

(continued from previous page)

```
INSERT INTO anggota VALUES ('21451', 'Budi Darmawan', 'L');
INSERT INTO anggota VALUES ('21569', 'Anggi Lestari', 'P');
INSERT INTO anggota VALUES ('21584', 'Joko Anantoko', 'L');
PRIMARY KEY no_id;
```

Berikut adalah penjelasan beberapa perintah dasar di atas:

```
CREATE TABLE Anggota (
```

- **CREATE TABLE** Fungsi dari perintah ini adalah untuk membuat sebuah tabel baru, yang tentu saja isinya masih kosong. Untuk mengisi tabel yang baru dibuat ini kita gunakan perintah INSERT INTO dan VALUES seperti yang akan dijelaskan di bawah.
- **Anggota** Adalah nama/judul dari tabel pertama yang akan kita buat pada basisdata ini. Judul tabel ini merupakan variable, artinya, Anda dapat menggunakan nama apapun yang Anda inginkan, asalkan merupakan satu buah kata (kita lihat, tanda spasi pada kata-kata ‘bidang tanah’ diganti dengan tanda underscore/garis bawah. Ini untuk menjaga variable tersebut hanya merupakan satu kata).

```
No_id INTEGER NOT NULL,
nama VARCHAR(20),
jenis_kelamin VARCHAR(2)
```

- **No\_id, Nama dan Jenis\_kelamin** No\_id merupakan nama kolom pertama yang kita isikan (lihat tabel pada contoh di atas agar lebih jelas). No\_id ini adalah kolom yang akan kita fungsikan untuk memuat identitas unik masing-masing orang. Demikian pula, Nama dan Jenis\_kelamin merupakan nama-nama kolom yang kita buat pada tabel anggota yang akan kita isi dengan keterangan-keterangan yang kita inginkan.
- **Tipe Data** INTEGER dan VARCHAR yang anda gunakan pada baris-baris perintah tadi berfungsi untuk menjelaskan tipe data dari kolom yang Anda buat. Berikut beberapa tipe data yang dapat anda gunakan:
  - INTEGER : Menyatakan data berupa bilangan bulat. Panjang data yang dapat dimuat adalah antara -2147483648 sampai 2147483647.
  - FLOAT : Menyatakan data berupa bilangan desimal
  - VARCHAR : Menyatakan sembarang karakter (huruf, angka, atau karakter yang lain). Apabila anda menambahkan bilangan dalam tanda kurung di belakangnya, misalnya VARCHAR(2), artinya anda membatasi maksimal hanya 2 karakter yang akan ditampilkan, tidak boleh lebih.
  - TEXT : Menampilkan baris teks. Bedanya dengan tipe data character adalah panjang data yang bisa ditampung. Anda bisa memasukkan beberapa baris kalimat pada tipe data ini.
  - DATE : Memasukkan tanggal ke dalam kolom
  - BOOLEAN : Apabila anda memilih tipe data ini untuk sebuah kolom, artinya kolom tersebut hanya bisa memiliki dua macam nilai: ‘TRUE’ atau ‘FALSE’. Biasanya tipe data ini digunakan untuk menyatakan pilihan. Perhatikan bahwa tiap kali anda membuat sebuah kolom, anda harus memisahkannya dengan baris perintah selanjutnya menggunakan tanda koma (,). Ini adalah salah satu aturan penulisan pada bahasa SQL.

```
PRIMARY KEY no_id;
```

- **PRIMARY KEY** Pada baris perintah ini, kita memberi tahu komputer bahwa kolom No\_id akan kita jadikan sebagai PRIMARY KEY. Primary key adalah kolom yang akan kita gunakan sebagai penghubung dengan tabel lain. Dalam hal ini, data yang berada pada kolom no\_id haruslah bersifat unik. Artinya, data-data pada kolom ini harus berbeda antara satu dengan yang lainnya.

### Menggunakan Query untuk menanyakan pertanyaan pada Basisdata

Di atas kita telah melihat penjelasan bagaimana bahasa SQL digunakan untuk membuat sebuah tabel baru. Pada bagian ini kita akan menanyakan beberapa pertanyaan pada tabel tersebut.

Sekarang, dengan menggunakan tabel anggota di atas sebagai acuan, kita dapat menanyakan pertanyaan berikut:

```
SELECT * FROM anggota
```

SELECT dan FROM pada baris perintah di atas digunakan untuk menampilkan isi dari tabel yang dipanggil (pada contoh adalah tabel anggota). Tanda asterisk (\*) di atas digunakan untuk menampilkan keseluruhan data pada semua kolom di tabel tersebut. Apabila anda memanggil perintah tersebut pada sebuah perangkat lunak basisdata (seperti PostgreSQL yang akan kita pelajari nanti), maka perintah tersebut akan menampilkan seluruh isi tabel anggota.

Perintah tersebut akan menampilkan semua isi kolom nama pada tabel anggota, seperti berikut:

Nama
Budi Darmawan
Anggi Lestari
Joko Anantoko

Apabila anda hanya ingin menampilkan satu baris tertentu, Anda dapat menggunakan perintah WHERE:

```
SELECT * FROM anggota WHERE No_id = 21451
```

hasilnya:

21569	Anggi Lestari	P
-------	---------------	---

Apa keluaran dari bahasa SQL berikut?

```
SELECT nama FROM anggota WHERE jenis_kelamin = 'L'
```

Sebuah perintah SQL dapat menjadi sangat kompleks: semakin rumit informasi yang ingin kita peroleh, semakin banyak relasi yang terlibat, maka akan semakin rumit pula Query yang harus kita ajukan. Meskipun demikian, SQL dapat memperoleh informasi yang cukup rumit hanya dengan beberapa perintah sederhana, sehingga sangat ideal untuk keperluan analisis data dalam jumlah besar, khususnya data spasial.

### DDL, DML, DCL dan TCL

Kita telah melihat beberapa perintah SQL yang dapat digunakan untuk ‘bertanya’ pada basisdata menggunakan Query tertentu. Perintah (*command*) dalam bahasa SQL seperti yang disebutkan di atas dapat dikelompokkan menjadi empat kelompok besar: *Data Definition Language* (DDL), *Data Manipulation Language* (DML), *Data Control Language* (DCL) dan *Transaction Control Language* (TCL). Keempat jenis perintah ini membagi bahasa SQL ke dalam jenis operasi yang dilakukan:

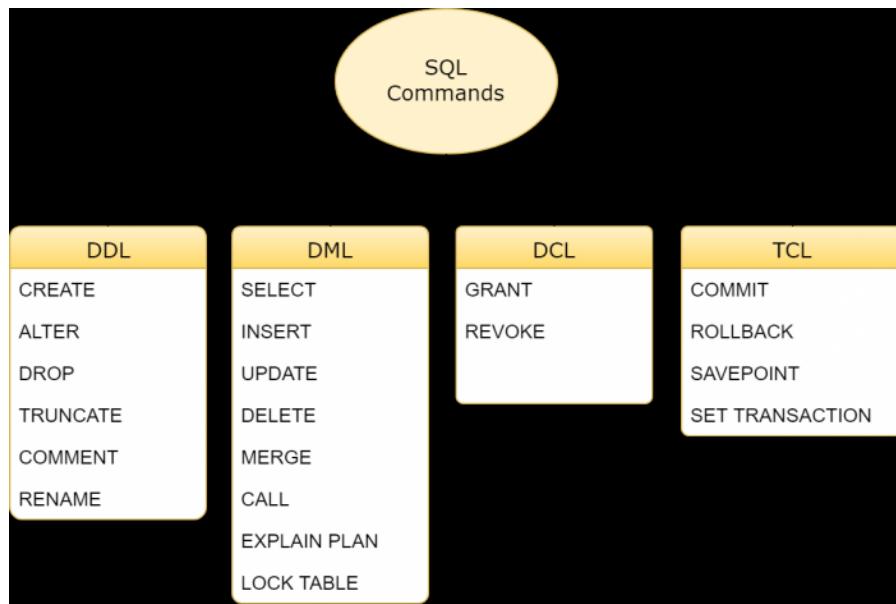


Fig. 2.3: Perintah dalam bahasa SQL

Berikut penjelasan untuk tiap operasi:

- *Data Definition Language* (DDL)

Perintah yang dikelompokkan sebagai DDL digunakan pada saat pembuatan skema dan pendefinisian basisdata, serta bagaimana data disimpan pada basidata.

- CREATE - Digunakan untuk membuat basisdata baru serta objek-objek yang terdapat di dalamnya (TABLE, INDEX, VIEW, FUNCTION, dan sebagainya)
- ALTER - Merubah struktur basidata dan objek di dalamnya
- DROP - Menghapus basisdata atau objek di dalamnya
- TRUNCATE - Menghapus semua *record* pada basisdata
- RENAME - Mengganti nama sebuah objek basisdata

- *Data Manipulation Language (DML)*

DML terdiri dari sebagian besar perintah yang kita gunakan pada sebuah basisdata. DML berkaitan dengan manipulasi data dan objek pada basisdata, termasuk untuk menyimpan, memodifikasi, mengambil data, menghapus dan memperbarui data pada basisdata.

- SELECT - Digunakan untuk memilih dan mengambil data dari basisdata berdasarkan kriteria tertentu
- INSERT - Digunakan untuk memasukkan data ke dalam relasi
- UPDATE - Memungkinkan pengubahan nilai dari beberapa atribut dalam tupel yang ada.
- DELETE - Digunakan untuk menghapus tupel dari tabel.
- MERGE - Menggabungkan data baru pada data yang sudah ada

Penggunaan tiap perintah di atas akan dikontrol oleh CONSTRAINT yang telah ditetapkan pada relasi atau basisdata tersebut. Contoh konstrain yang diterapkan pada basisdata adalah pembatasan tipe data pada kolom sebuah tabel.

Berikut adalah contoh dari penggunaan perintah DML:

### Insert Operation

Operasi Insert menambahkan data baru pada sebuah tabel sesuai dengan kolom yang tersedia



CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive
4	Alibaba	Active

### Update Operation

Pada tabel di bawah, status CustomerName=Apple berubah dari inactive menjadi active



CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive
4	Alibaba	Active

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Active
4	Alibaba	Active

**Delete Operation** Pada operasi penghapusan, suatu kriteria diberikan melalui Query untuk



CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Active
4	Alibaba	Active

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
4	Alibaba	Active

### Select Operation



CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
4	Alibaba	Active

Operasi berikut memilih Customer dengan nama “Amazon”

- *Data Control Language (DCL)*

Perintah pada DCL sebagian besar terkait dengan pemberian hak, permission, dan pengaturan kontrol lain pada basisdata.

- GRANT - Memberikan hak dan izin (*Privileges*) kepada pengguna untuk mengakses sebuah basisdat
- REVOKE - Menghilangkan hak pada pengguna
- *Transaction Control Language (TCL)*

TCL digunakan untuk mengatur transaksi pada sebuah basisdata, yaitu:

- COMMIT - menyimpan operasi transaksi
- ROLLBACK - kembali pada rekaman transaksi sebelumnya pada saat terjadi kesalahan
- SET TRANSACTION - mendefinisikan karakteristik transaksi

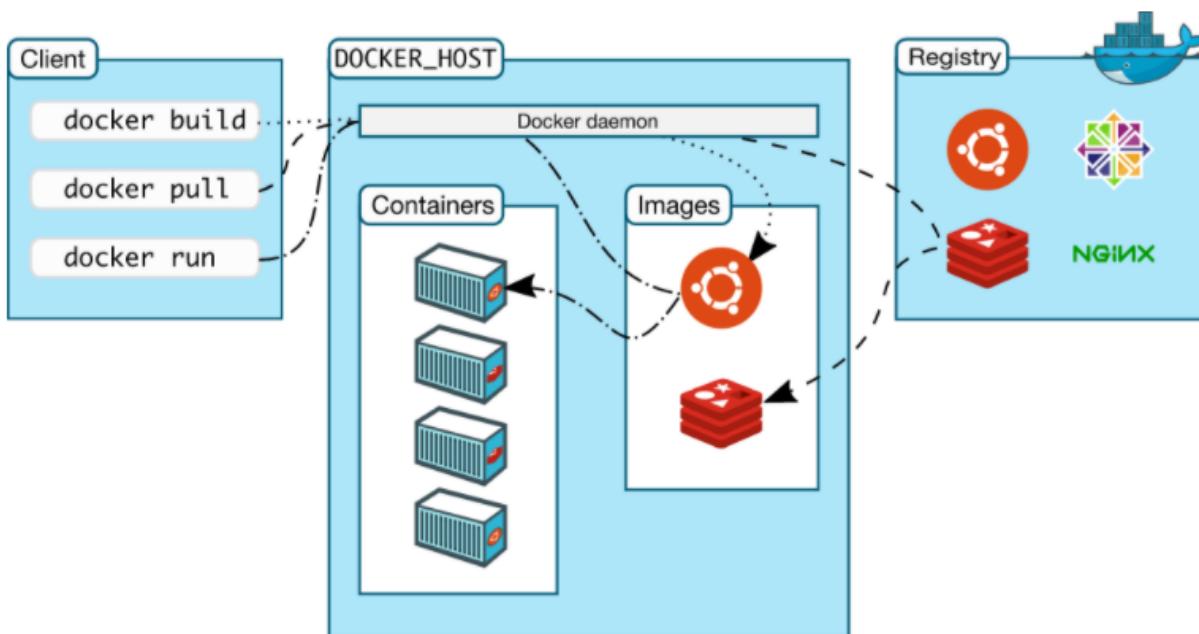
Setelah ini, kita akan menggunakan PostgreSQL untuk melakukan Query menggunakan Bahasa SQL. Terlebih dahulu lakukan instalasi sesuai dengan panduan yang diberikan sebagai berikut.

## 2.1.4 Instalasi dan Pengaturan Basisdata PostgreSQL

PostgreSQL merupakan perangkat lunak yang digunakan untuk melakukan manajemen basisdata, sedangkan PostGIS merupakan perangkat lunak tambahan ('Plugin') pada PostgreSQL yang digunakan untuk menambahkan fungsi keruangan (spasial) pada PostgreSQL. Pada bagian ini diuraikan cara membuat basisdata dengan menggunakan PostgreSQL/PostGIS, dengan lebih dahulu membahas mengenai instalasi PostgreSQL dan PostGIS menggunakan Docker.

### Latihan: Instalasi dan Pengaturan Docker

Pada latihan ini akan dibahas mengenai instalasi Docker. Docker nantinya akan kita gunakan sebagai Container untuk mempermudah instalasi PostgreSQL dan PostGIS. Arsitektur Docker adalah sebagai berikut:



Langkah berikut merujuk pada [Panduan Resmi Instalasi Docker](#):

1. Lakukan update, kemudian lakukan instalasi beberapa dependencies untuk Docker

```
sudo apt-get install \
apt-transport-https \
ca-certificates \
curl \
gnupg-agent \
software-properties-common
```

2. Tambahkan GPG official untuk Docker:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

3. Periksa fingerprint untuk GPG yang telah ditambahkan ke Ubuntu

```
sudo apt-key fingerprint 0EBFCD88
```

4. Tambahkan repository baru. Docker memiliki repository tersendiri sehingga perlu memberi tahu apt untuk mengambil instalasi dari repository ini:

```
sudo add-apt-repository \
"deb [arch=amd64] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) \
stable"
```

5. Update kembali apt:

```
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

6. Uji apakah docker sudah berhasil diinstall

```
sudo docker run hello-world
```

7. Memasukkan user ke dalam grup docker. Langkah ini perlu untuk menghilangkan sudo pada perintah untuk memanggil docker.

```
sudo usermod -aG docker $USER
newgrp docker
```

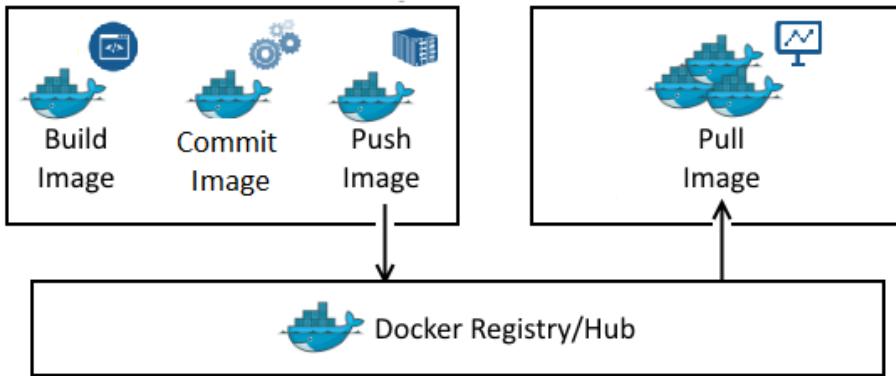
---

### Catatan

Docker memerlukan Systemd untuk menjalankan servicenya, sama seperti Tomcat pada materi sebelumnya. Untuk itu, apabila muncul pesan kesalahan terkait `systemd` atau `systemctl` pada WSL, coba kembali langkah-langkah untuk mengoreksi system seperti yang telah disebutkan sebelumnya.

---

Setelah instalasi Docker berhasil, selanjutnya adalah menggunakan Docker untuk melakukan instalasi Image untuk PostGIS. Image adalah paket aplikasi yang dibuat oleh developer untuk memuat serangkaian perintah dan dapat diduplikasi oleh siapapun yang menghendaki. Image pada Docker tersimpan pada Docker Hub sebagai satu repository tempat Image dibagikan:



### Latihan: Memulai Docker untuk PostGIS

Untuk keperluan latihan ini kita akan menggunakan Image PostGIS yang dibuat oleh [Kartoza](#). Sebenarnya, terdapat banyak Image yang sudah dibuat oleh banyak developer. Kita dapat mencari Image terkait PostGIS menggunakan perintah ‘search’ pada Docker:

```
docker search postgis
```

hasilnya:

```
danylaksono@DNPC:~/workspace/panduan-fossgis$ docker search postgis
NAME                           DESCRIPTION                                     STARS   OFFICIAL   AUTOMATED
mdillon/postgis                PostGIS container based on the official Post...   416     [OK]
kartoza/postgis                Ready to run PostGIS geospatial database       138     [OK]
postgis/postgis                 PostGIS is a spatial database extender for P...   54      [OK]
geographica/postgis             This repository contains Docker image builds...  13      [OK]
nickblah/postgis                Automated and up-to-date Docker images for a...   7       [OK]
campnocamp/postgis              PostgreSQL + PostGIS                               6       [OK]
openmaptiles/postgis            A PostgreSQL Docker image with support for G...   6       [OK]
postgis/postgis-build-env       Environments for PostGIS CI testing, with va...   4       [OK]
pamtراك06/postgis-pgrouting-osm postgis/pgrouting with latest osm tools : os...   3       [OK]
yagajs/postgis                  Postgis based on official Postgres repository  2       [OK]
chewai/postgis                  PostgreSQL and PostGIS in a Docker container  2       [OK]
geonode/postgis                 PostGIS image for GeoNode databases               2       [OK]
waiganjo/postgis-hll           Postgres 11 with hll, postgis, pg_cron exten...  1       [OK]
makinacorus/postgis            postgis                                         1       [OK]
noangntvn/postgis              Postgresql with postgis extensions can run o...  1       [OK]
neowaylabs/postgis              https://github.com/neowaylabs/docker-images/      1       [OK]
corpusops/postgis               A small and secure Postgis image.                   1       [OK]
uroybd/postgis-alpine          This image is based on official postgres alp...  0       [OK]
calthorpeanalytics/postgis     Fork of https://github.com/appropriate/docke...  0       [OK]
corpusops/postgis-bare         https://github.com/corpusops/docker-postgres/  0       [OK]
projectairws/postgis           0
mapcentia/postgis              PostGIS for GC2                                0       [OK]
comes/postgis                  postgresql with postgis extensions installed  0       [OK]
showpass/postgis                PostGIS images built by Showpass                 0       [OK]
danylaksono@DNPC:~/workspace/panduan-fossgis$
```

Image PostGIS yang dibuat oleh tim Kartoza dapat dijumpai pada Docker Hub sebagai berikut: <https://hub.docker.com/r/kartoza/postgis/>

Berikut adalah langkah instalasi Image PostGIS dengan Docker<sup>2</sup>:

1. Membuat Persistance Volume. Persistance Volume digunakan agar Docker menyimpan berkas pada lokasi yang dapat diakses, yaitu pada mesin WSL. Berikut perintahnya:

<sup>2</sup> <https://alexurquhart.com/post/set-up-postgis-with-docker/>

```
docker volume create pg_data
```

2. Untuk menginstall PostGIS kita gunakan perintah berikut:

```
docker run --name=postgis -d -e POSTGRES_USER=postgres -e POSTGRES_PASS=postgres -  
-e POSTGRES_DBNAME=gis -e ALLOW_IP_RANGE=0.0.0.0/0 -p 5432:5432 -v pg_data:/var/  
lib/postgresql --restart=always kartoza/postgis:13.0
```

- docker run –name = postgis memberi tahu Docker bahwa container baru kami akan diberi nama postgis
- -d menjalankan kontainer di latar belakang (mode terpisah)
- -e POSTGRES\_USER = postgres the -eflag menyetel variabel lingkungan di dalam penampung. Yang ini digunakan untuk mengonfigurasi nama peran login di PostgreSQL yang akan memiliki hak superuser (admin) di database. Anda dapat mengubah nama ini menjadi apa pun yang Anda inginkan.
- -e POSTGRES\_PASS = postgres menyetel variabel lingkungan yang akan menyetel kata sandi peran login menjadi `kata sandi. Anda dapat mengatur ini menjadi apa pun yang Anda inginkan.
- -e POSTGRES\_DBNAME = gis seperti yang bisa Anda tebak, variabel lingkungan memberi tahu container untuk membuat database baru di server dengan nama gis. Setelah database dibuat maka extension PostGIS akan diaktifkan di dalamnya.
- -e ALLOW\_IP\_RANGE = 0.0.0.0 / 0 memberi tahu container untuk mengkonfigurasi PostgreSQL untuk menerima koneksi dari siapa pun. Jika Anda tidak mengatur ini maka database hanya akan menerima koneksi dari alamat menggunakan subnet jaringan Docker.
- -p 5432: 5432 memetakan port 5432 pada VM host ke port 5432 pada container. Ini diperlukan karena server database mendengarkan koneksi pada port 5432 secara default.
- -v pg\_data: / var / lib / postgresql memberi tahu sistem file container untuk memasang volume pg\_data yang baru saja kita buat ke jalur / var / lib / postgresql. Ini berarti bahwa data apa pun yang disimpan atau dibuat container di direktori itu akan disimpan dalam volume pg\_data.
- –restart = selalu membuat kebijakan restart untuk penampung Anda. Sekarang penampung Anda akan mulai setiap kali mesin virtual Docker dimulai. Jika ini tidak disetel, Anda harus memulai penampung secara manual setiap kali VM di-boot dengan docker start postgis
- kartoza / postgis: 13.0 memberitahu Docker untuk menarik repositori kartoza / postgis dari Docker Hub, menggunakan PostgreSQL versi 9.6 dan PostGIS versi 2.4. Anda dapat melihat versi lain yang tersedia di [sini](#)

3. Tunggu sampai instalasi selesai. Periksa apakah mesin yang digunakan sudah menyala menggunakan perintah:

```
docker ps
```

Selanjutnya, PostGIS dan PostgreSQL siap untuk digunakan.

```
danylaksono@DNPC:~/workspace/panduan-fossgis$ docker ps  
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES  
23ddba6106e1        kartoza/postgis:13.0   "/bin/sh -c /scripts..."   7 hours ago       Up 7 hours          0.0.0.0:5432->5432/tcp   postgis  
danylaksono@DNPC:~/workspace/panduan-fossgis$ -
```

## 2.1.5 Menggunakan Query SQL pada Basisdata

Untuk mengirimkan SQL pada Basisdata PostgreSQL dapat digunakan beberapa aplikasi antarmuka yang berbeda. `psql` dan `pgAdmin` barangkali merupakan antarmuka untuk basisdata PostgreSQL yang paling terkenal. Sejatinya, koneksi pada basisdata dapat dilakukan pada berbagai aplikasi, dengan syarat terdapat port dan host yang terbuka untuk melakukan koneksi pada PostgreSQL seperti yang akan kita lakukan nanti dengan QGIS. Untuk memulai, terlebih dahulu akan kita gunakan kedua aplikasi yang telah disebutkan di atas.

### Latihan: Menggunakan `psql` untuk melakukan query sederhana

`psql` merupakan aplikasi untuk manajemen client PostgreSQL. Dengan aplikasi ini, script SQL dapat dibuat melalui *command prompt*. Contohnya, koneksi pada basisdata dapat dilakukan sebagai berikut:

```
psql -d database -h host -U user -W
```

Aplikasi ini tersedia bersama dengan instalasi PostgreSQL dan PostGIS yang sudah kita install menggunakan **Docker Kartoza**. Dalam hal ini, aplikasi ini sekarang tersimpan di dalam mesin Docker, dan bukan pada WSL, sehingga kita perlu melakukan [langkah tambahan](#) untuk melakukan akses tersebut:

1. Masuk ke dalam Docker menggunakan perintah `docker exec`

```
docker exec -it postgis bash
```

2. Install text editor, misalnya `nano`:

```
apt-get update && apt-get install nano
```

3. Masuk ke folder instalasi PostgreSQL

```
cd /etc/postgresql/13/main/
```

4. Dengan menggunakan `nano`, lakukan editing pada file `pg_hba.conf`. Ubah peer menjadi `trust` pada **Database administrative login by Unix domain socket** seperti berikut:

```
#  
# Database administrative login by Unix domain socket  
local   all      postgres          trust  
  
# TYPE  DATABASE        USER        ADDRESS             METHOD  
# "all" means this rule is applied without regard to connection settings in pg_hba.conf or pg_ident.conf
```

5. Keluar dari mesin dan lakukan restart untuk kontainer PostGIS

```
docker restart postgis
```

6. Gunakan `psql` seperti berikut:

```
docker exec -it postgis psql -U postgres
```

```
danylaksono@DANYLAKSONO:~/dummy$ docker restart postgis
postgis
danylaksono@DANYLAKSONO:~/dummy$ docker exec -it postgis psql
psql (13.1 (Debian 13.1-1.pgdg100+1))
Type "help" for help.
```

maka kita akan masuk ke shell untuk `psql`: `postgres=#`

7. Pada shell ini kita dapat langsung mengetikkan Query yang kita inginkan, misalnya: `postgres=# SELECT * FROM anggota;`
8. Selain melalui konsol, kita juga dapat membuat query menggunakan *command*, seperti berikut:

```
docker exec -it postgis psql -U postgres -d latihan -c 'SELECT * FROM anggota'
```

### Catatan

Tentunya untuk kedua contoh SELECT di atas, basisdata latihan dan tabel anggota harus sudah tersedia terlebih dahulu.

Untuk latihan ini, coba gunakan query pada tabel anggota seperti di atas menggunakan `psql`, yaitu dari pembuatan tabel hingga mengajukan pertanyaan. Buat basisdata baru dengan nama Latihan, kemudian masukkan tabel anggota. Gunakan perintah-perintah DDL dan DML untuk membuat dan mengoperasikan basisdata ini melalui `psql`.

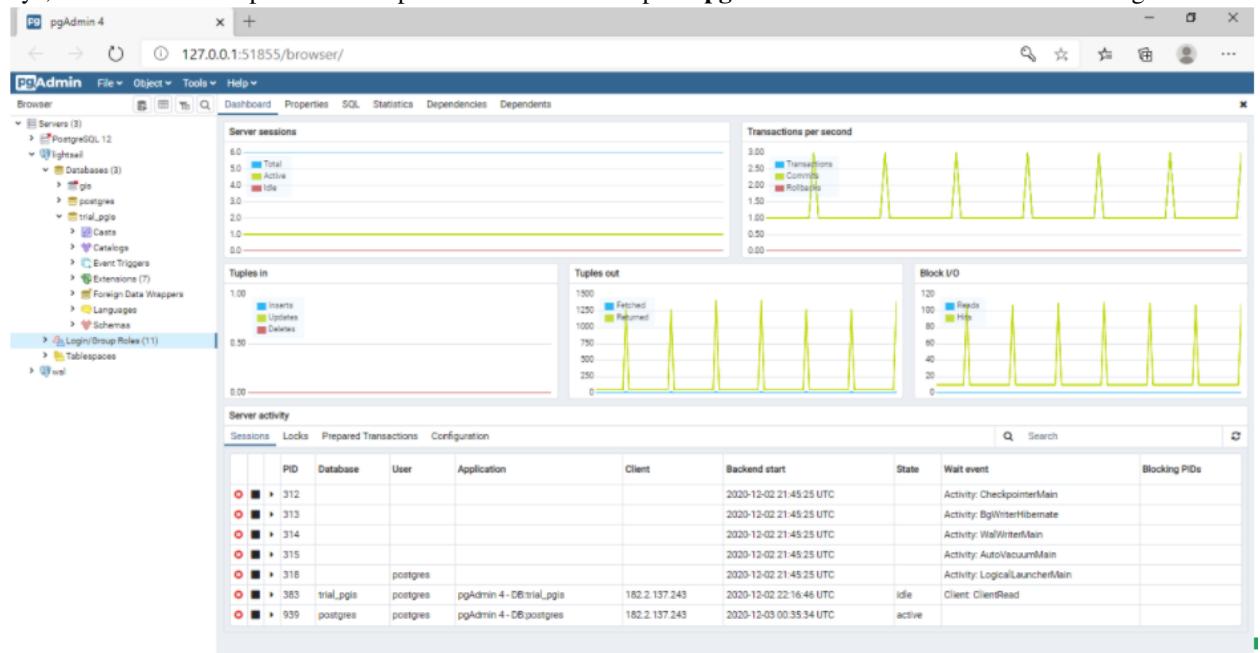
Daftar perintah untuk `psql` dapat dilihat pada: <https://www.postgresqltutorial.com/postgresql-cheat-sheet/>.

### Latihan: Menggunakan pgAdmin sebagai GUI Basisdata PostgreSQL

Pada bagian ini, **pgAdmin4** akan digunakan sebagai antarmuka untuk PostgreSQL. pgAdmin merupakan antarmuka PostgreSQL yang menyediakan GUI (*Graphical User Interface*) sehingga memudahkan untuk mengajukan Query, manajemen basisdata, memonitor transaksi, dan seterusnya.

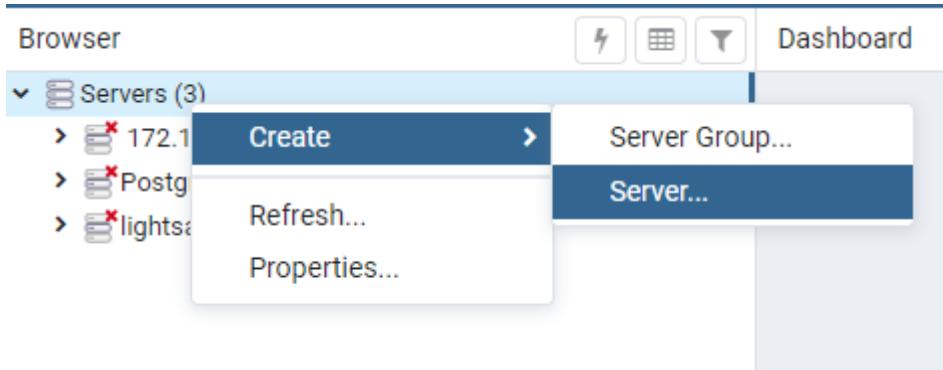
Terlebih dahulu unduh pgAdmin4 dari alamat berikut: <https://www.pgadmin.org/download/>

Selanjutnya, lakukan instalasi pada sistem operasi Windows. Tampilan **pgAdmin4** setelah instalasi adalah sebagai

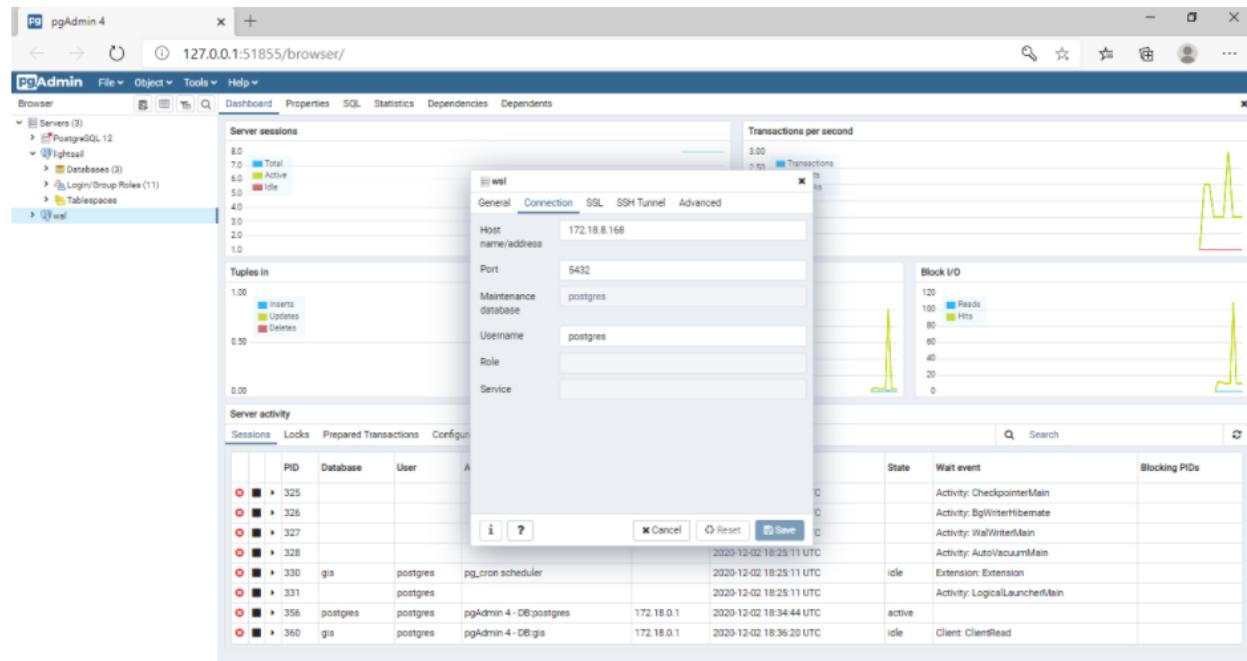


berikut:

Koneksi dapat dilakukan pada basisdata melalui pgAdmin4 sebagai berikut. Klik kanan pada Server, kemudian pilih Create Server.



Masukkan nama dan parameter koneksi seperti contoh berikut:



dimana *alamat host* adalah ip address yang digunakan oleh mesin Docker.

untuk mendapatkan alamat tersebut, gunakan perintah:

```
ip addr
```

kemudian lihat ip address untuk `eth0`. Alamat ini adalah alamat IP yang digunakan oleh Windows untuk berkomunikasi dengan WSL, sehingga kita dapat menggunakankannya untuk terhubung dengan PostgreSQL. Pada kasus instalasi PostgreSQL langsung di mesin server, alamat yang dicari ini dapat dibuat cukup dengan menggunakan `localhost` saja. Diagram berikut menggambarkan bagaimana Windows terhubung pada WSL dengan menggunakan koneksi melalui IP Address yang dimaksud.

Setelah terhubung, PostgreSQL akan menampilkan beberapa menu yang dapat digunakan untuk melakukan manajemen basisdata. Untuk melakukan Query pada basisdata tertentu, klik pada basisdata tersebut, kemudian gunakan menu `Query Tool` untuk membuat query terhadap basisdata ini.

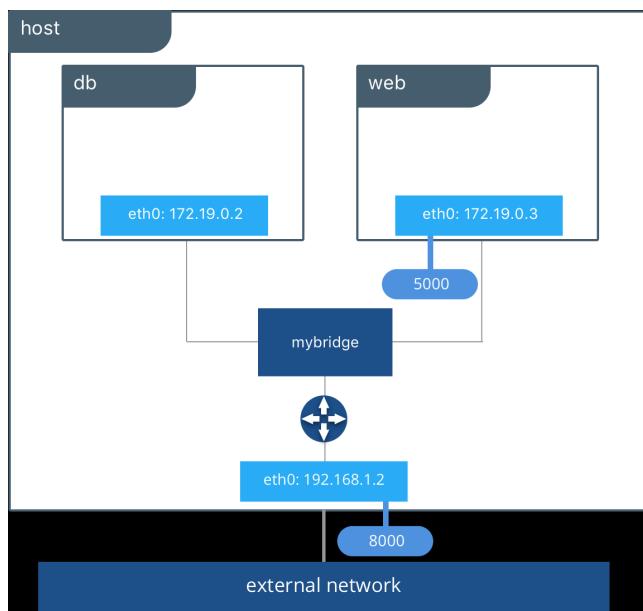
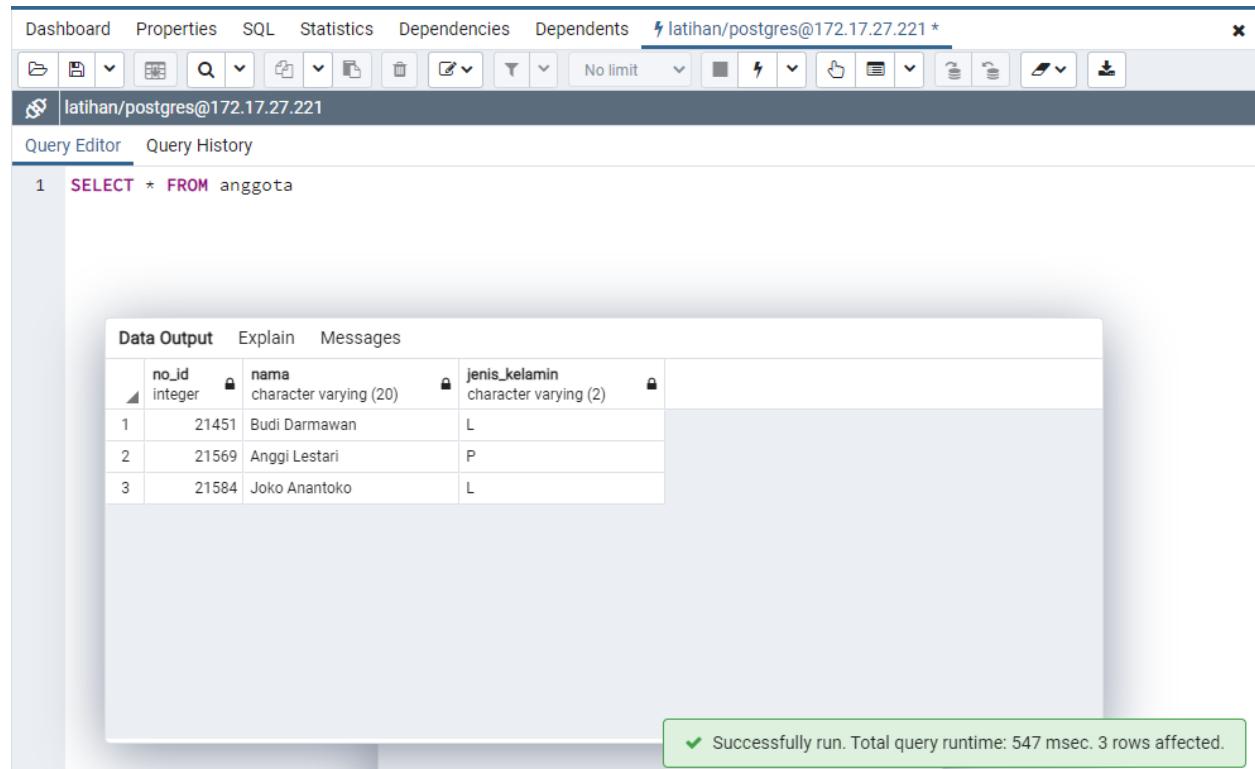


Fig. 2.4: Koneksi pada WSL

Screenshot of the pgAdmin interface showing the database structure. The left sidebar displays a tree view of servers, databases, and tables. The 'gis' database is selected. A context menu is open over the 'latihan' schema, listing options such as Create, Refresh..., Delete/Drop, CREATE Script, Disconnect Database..., Maintenance..., Backup..., Restore..., Grant Wizard..., Query Tool..., and Properties... .

Perintah SQL seperti yang kita sebutkan di atas dapat digunakan pada jendela Query yang tersedia.



The screenshot shows the pgAdmin 4 interface. At the top, there's a toolbar with various icons for database management. Below the toolbar, the connection status is shown as "latihan/postgres@172.17.27.221". The main area has tabs for "Query Editor" (which is selected) and "Query History". In the "Query Editor" tab, the following SQL query is written:

```
1 | SELECT * FROM anggota
```

Below the query, the "Data Output" tab is active, displaying a table with three rows of data:

	no_id	nama	jenis_kelamin
1	21451	Budi Darmawan	L
2	21569	Anggi Lestari	P
3	21584	Joko Anantoko	L

At the bottom right of the data output window, there's a green success message: "Successfully run. Total query runtime: 547 msec. 3 rows affected."

## Catatan

Mesin Docker PostGIS yang kita panggil pada WSL akan tetap berjalan sampai kita matikan. Untuk memeriksa mesin apa saja yang sedang berjalan, kita dapat gunakan:

```
docker ps
```

Maka Docker akan menampilkan daftar mesin yang sedang berjalan. Untuk menghentikan semua mesin, kita gunakan perintah:

```
docker stop $(docker ps -aq)
```

dan untuk memulai kembali mesin tersebut, cukup panggil dengan menggunakan:

```
docker start postgis
```

dimana `postgis` adalah nama mesin yang kita definisikan menggunakan parameter `--name` pada saat memanggil image ini. Dalam hal ini Docker tidak akan mengunduh instalasi baru karena berkas tersebut masih ada pada WSL.

(continues on next page)

(continued from previous page)

## 2.2 Administrasi Pengguna pada Basisdata

Pada basisdata enterprise, concurrent use atau pemakaian bersama menjadi kebutuhan utama pada sistem yang berjalan. Sama seperti pada sistem operasi Linux yang sudah dibahas sebelumnya, terdapat pengaturan untuk memberikan izin pada basisdata yang memungkinkan pengguna berbeda mempunyai izin akses yang berbeda.

### 2.2.1 Roles, Group dan Privileges

Izin pada pengguna di PostgreSQL mirip dengan pemberian izin pada Linux: pembagian peran (role) dapat diberikan pada user maupun grup, dengan tiap role memiliki hak akses (privileges) yang berbeda

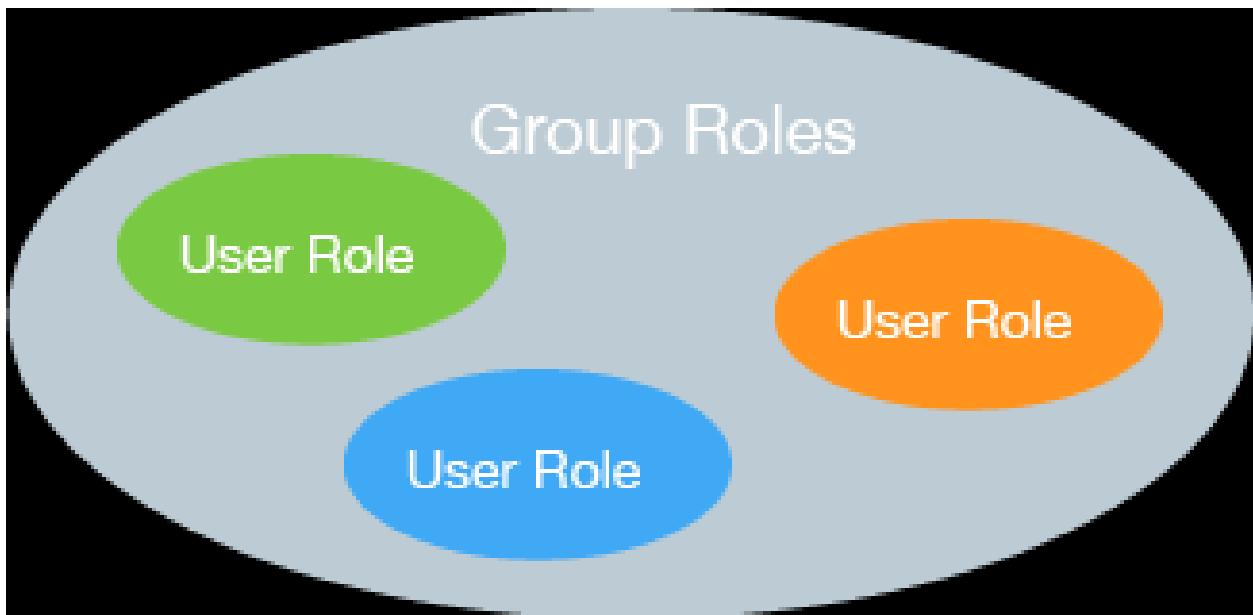


Fig. 2.5: Roles pada PostgreSQL dapat diberikan pada grup maupun pada user

Berikut adalah beberapa peran yang dapat diberikan pada pengguna atau grup:

- **SUPERUSER** - SUPERUSER database perlu mendapat perhatian khusus. Intinya, peran dengan atribut ini dapat membuat SUPERUSER lain. Faktanya, atribut ini diperlukan untuk membuat peran SUPERUSER lainnya. Karena peran dengan atribut ini mengabaikan semua pemeriksaan izin, berikan hak istimewa ini dengan bijaksana.
- **CREATEDB** - Memungkinkan peran untuk membuat database.
- **CREATEROLE** - Dengan atribut ini, peran dapat mengeluarkan perintah CREATE ROLE, yaitu membuat peran lain.

- **LOGIN** - Memungkinkan kemampuan untuk login. Nama peran dengan atribut ini dapat digunakan dalam perintah koneksi klien.

Pada PostgreSQL, pengaturan ini dapat dijumpai pada menu Create Roles, seperti berikut:

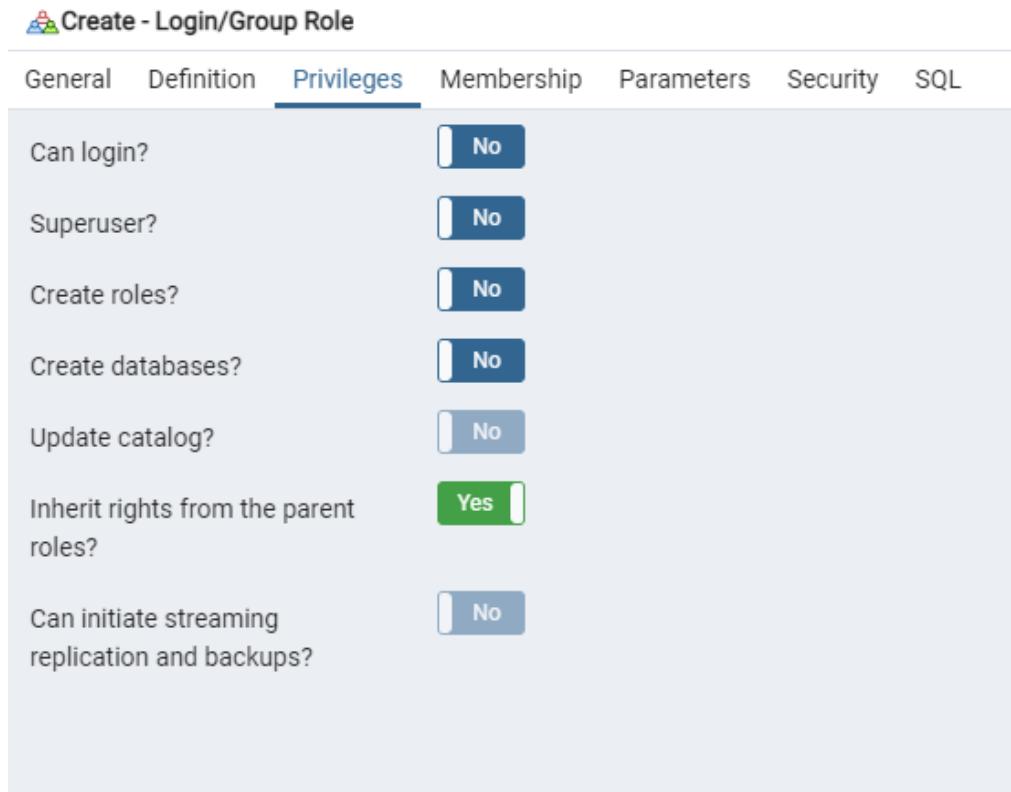


Fig. 2.6: Menu pengaturan Roles pada PostgreSQL

## 2.2.2 Pengaturan hak akses pada PostgreSQL

Dengan PostgreSQL, Anda dapat membuat pengguna dan peran dengan izin akses secara *granular*. Pengguna atau peran baru harus secara selektif diberikan izin yang diperlukan untuk setiap objek database. Ini memberi banyak kekuatan kepada pengguna akhir, tetapi pada saat yang sama, itu membuat proses pembuatan pengguna dan peran dengan izin yang benar berpotensi rumit.

PostgreSQL memungkinkan Anda memberikan izin langsung ke pengguna database. Namun, sebagai praktik yang baik, Anda disarankan untuk membuat **banyak peran** dengan kumpulan izin tertentu berdasarkan aplikasi dan persyaratan akses. Kemudian tetapkan peran yang sesuai untuk setiap pengguna. Peran harus digunakan untuk menerapkan model dengan hak istimewa terendah untuk mengakses objek database. Pengguna master pada PostgreSQL harus digunakan hanya untuk tugas administrasi database seperti membuat pengguna, peran, dan database lain. Pengguna master tidak boleh digunakan oleh aplikasi. Berikut adalah contoh pembagian peran dan pemberian izin:

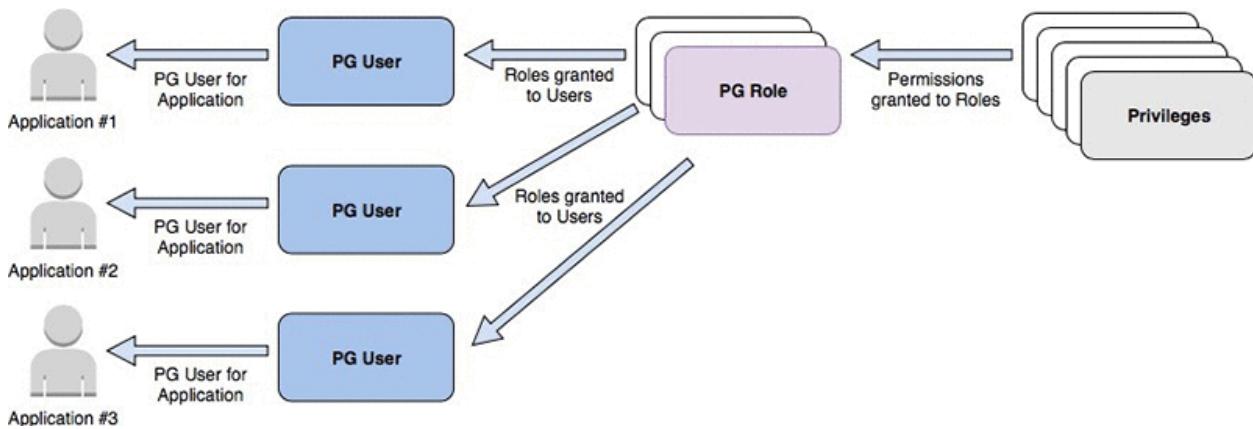


Fig. 2.7: Contoh pengaturan untuk pembagian peran pada PostgreSQL

### Membuat pengguna dan grup baru

Pengguna pada PostgreSQL dibuat melalui query berikut:

```

CREATE USER username [ [ WITH ]
    SYSID uid
    | [ ENCRYPTED | UNENCRYPTED ] PASSWORD 'password'
    | CREATEDB | NOCREATEDB
    | CREATEUSER | NOCREATEUSER
    | IN GROUP groupname [, ...]
    | VALID UNTIL 'abstime' ]

```

Sebagai contoh, perintah berikut akan membuat **pengguna baru** dengan nama ‘user1’ dan password ‘password1’:

```
CREATE USER user1 WITH PASSWORD 'password1';
```

Adapun **grup** merupakan salah satu cara untuk mengelompokkan pengguna tertentu. Grup dapat dibuat dengan perintah berikut:

```
CREATE GROUP sales WITH USER user1;
```

### Menerapkan peran baru

**Role** merupakan cara PostgreSQL mengatur batasan akses oleh pengguna. Tiap Role dapat memiliki pengaturan *Privileges* yang berbeda untuk membatasi hak akses yang dapat dilakukan oleh tiap-tiap pengguna. Berikut adalah bagaimana peran dibuat dengan menggunakan bahasa SQL:

```
CREATE ROLE user_viewer WITH LOGIN PASSWORD 'pass2';
```

PostgreSQL 13 telah dilengkapi dengan beberapa roles default yang dapat digunakan, yaitu sebagai berikut:

Role	Allowed Access
pg_read_all_settings	Read all configuration variables, even those normally visible only to superusers.
pg_read_all_stats	Read all pg_stat_* views and use various statistics related extensions, even those normally visible only to superusers.
pg_stat_scan_tables	Execute monitoring functions that may take ACCESS SHARE locks on tables, potentially for a long time.
pg_monitor	Read/execute various monitoring views and functions. This role is a member of pg_read_all_settings, pg_read_all_stats and pg_stat_scan_tables.
pg_signal_backend	Signal another backend to cancel a query or terminate its session.
pg_read_server_files	Allow reading files from any location the database can access on the server with COPY and other file-access functions.
pg_write_server_files	Allow writing to files in any location the database can access on the server with COPY and other file-access functions.
pg_execute_server_program	Allow executing programs on the database server as the user the database runs as with COPY and other functions which allow executing a server-side program.

Peran *default* tersebut dapat diberikan pada pengguna tertentu melalui perintah:

```
GRANT pg_read_server_files TO user1;
```

Demikian pula, perintah tersebut dapat dilakukan untuk grup, sehingga hak akses menjadi milik semua pengguna pada grup tersebut.

### Latihan: Menggunakan Pengaturan Roles untuk membatasi akses

Untuk latihan ini, kita akan gunakan tabel anggota pada latihan sebelumnya. Kita akan membuat skema pengaturan pengguna melalui grup, user dan roles untuk membatasi akses pada tabel anggota. Kita akan menggunakan Query pada PostgreSQL untuk keperluan ini (bukan antarmuka PostgreSQL - menggunakan Query relatif lebih sederhana).

	no_id integer	nama character varying (20)	jenis_kelamin character varying (2)
1	21451	Budi Darmawan	L
2	21569	Anggi Lestari	P
3	21584	Joko Anantoko	L

- Langkah pertama adalah membuat *role* baru untuk pengguna kita nantinya. Role ini diatur agar hanya dapat membaca tabel tertentu:

```
CREATE ROLE readonly_tabel;
GRANT CONNECT ON DATABASE latihan TO readonly_tabel;
GRANT USAGE ON SCHEMA public TO readonly_tabel;
```

Tiga perintah di atas berfungsi untuk membuat role baru, memberikan hak akses atas basisdata latihan kepada role tersebut, serta memberi akses pada skema public yang terdapat di dalam basisdata latihan. Hasilnya adalah sebagai berikut:

The screenshot shows a PostgreSQL terminal window with the connection identifier `latihan/postgres@172.17.27.221`. The `Query Editor` tab is active, displaying the following SQL commands:

```
1 CREATE ROLE readonly_tabel;
2 GRANT CONNECT ON DATABASE latihan TO readonly_tabel;
3 GRANT USAGE ON SCHEMA public TO readonly_tabel;
```

Below the query editor, the `Messages` tab is visible, showing the output of the commands:

```
GRANT
Query returned successfully in 231 msec.
```

2. Kita akan membatasi akses ke tabel anggota agar role `readonly_tabel` hanya dapat membaca kolom id dan jenis kelamin saja.

```
GRANT SELECT (no_id, jenis_kelamin) ON TABLE public.anggota TO readonly_tabel;
```

pada perintah di atas, user budi hanya dapat mengakses kolom no\_id dan jenis\_kelamin

3. Membuat pengguna baru untuk akses dengan role ini

```
CREATE USER budi WITH PASSWORD 'secret';
GRANT readonly_tabel TO budi;
```

4. Lakukan koneksi ke PostgreSQL dengan user baru ini. Gunakan antarmuka PostgreSQL untuk Logout, kemudian login kembali dengan user budi dan password secret.
5. Buat query untuk memilih seluruh isi tabel anggota:

```
SELECT * FROM TABLE anggota
```

hasilnya adalah error karena user budi tidak diizinkan untuk mengakses seluruh isi tabel:

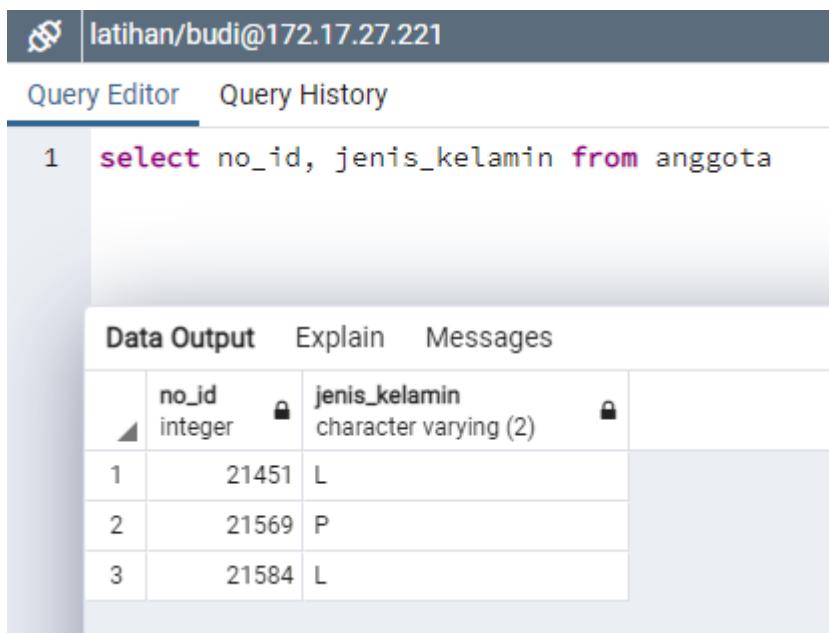
The screenshot shows a PostgreSQL terminal window with the connection identifier `latihan/budi@172.17.27.221`. The `Query Editor` tab is active, displaying the following SQL command:

```
1 Select * from anggota
```

Below the query editor, the `Messages` tab is visible, showing the error output:

```
ERROR: permission denied for table anggota
SQL state: 42501
```

6. Apabila kita hanya memilih kolom `no_id` dan `jenis_kelamin`, maka hasilnya:



The screenshot shows a PostgreSQL query editor interface. At the top, it displays the connection information: `latihan/budi@172.17.27.221`. Below this, there are tabs for "Query Editor" and "Query History", with "Query Editor" being active. A single-line query is entered in the editor area: `1 select no_id, jenis_kelamin from anggota`. Below the editor is a "Data Output" section containing a table with three rows of data. The table has two columns: `no_id` (integer) and `jenis_kelamin` (character varying(2)). The data is as follows:

	<code>no_id</code>	<code>jenis_kelamin</code>
1	21451	L
2	21569	P
3	21584	L

Latihan di atas menunjukkan bagaimana pengaturan role pada pengguna dapat digunakan untuk membatasi akses pada basisdata, tabel, kolom maupun view. Dengan demikian kita dapat mengatur hak akses yang cukup kompleks untuk berbagai kebutuhan.

### 2.2.3 Mengatur koneksi multi-pengguna

Apabila terdapat beberapa pengguna, maka hak akses (privileges) untuk tiap pengguna ini dapat ditentukan dengan mengurnya melalui roles, group maupun langsung ke pengguna itu sendiri.

#### Latihan: Pengaturan Privileges

PostgreSQL memiliki dua tingkat otorisasi, satu di tingkat database, yang disebut otentifikasi berbasis `host`, dan satu lagi di tingkat yang lebih detil pada tabel, `view`, dan `sequence`. Adapun daftar permission yang dapat diberikan pada pengguna adalah:

Privilege	short name	Description
SELECT	r	Can read data from the object.
INSERT	a	Can insert data into the object.
UPDATE	w	Can change data in the object.
DELETE	d	Can delete data from the object.
RULE	R	Can create a rule on the table
REFERENCES	x	Can create a foreign key to a table. Need this on both sides of the key.
TRIGGER	t	Can create a trigger on the table.
TEMPORARY	T	Can create a temporary table.
EXECUTE	X	Can run the function.
USAGE	U	Can use the procedural language.
ALL		All appropriate privileges. For tables, this equates to arwdRxt

Berikut adalah contoh pemberian izin kepada pengguna untuk mengakses basisdata tertentu:

```
GRANT { { SELECT | INSERT | UPDATE | DELETE | RULE | REFERENCES | TRIGGER }
[, ...] | ALL [ PRIVILEGES ] }
ON [ TABLE ] tablename [, ...]
TO { username | GROUP groupname | PUBLIC } [, ...] [ WITH GRANT OPTION ]
```

Sebagai contoh, untuk memberikan izin pada ‘bob’ untuk memodifikasi tabel **supplier**:

```
GRANT INSERT, UPDATE, DELETE ON TABLE suppliers TO bob;
```

### Latihan: Terhubung dengan server basisdata

Sebagai latihan, gunakan koneksi berikut untuk terhubung pada basisdata PostGIS yang tersimpan di Amazon Lightsail:  
host: 54.254.9.31 port: 5432

user 1:

- username: view\_only
- password: ‘viewer’
- role: pg\_read\_server\_file

user 2:

- username: editor\_data
- password: ‘editor’
- role: pg\_write\_server\_file

user 3 (SUPERUSER):

- username: postgres
- password: ‘postgres’

## 2.3 Basisdata Spasial dengan PostGIS

Pada bagian ini akan dibahas mengenai bagaimana menggunakan PostGIS untuk melakukan analisis spasial sederhana. QGIS akan digunakan sebagai antarmuka utama untuk melakukan koneksi pada PostGIS dan menampilkan hasil analisis.

---

### Catatan

Latihan ini akan menggunakan QGIS, sehingga terlebih dahulu lakukan instalasi QGIS.

Adapun Data untuk latihan ini dapat diperoleh dari link berikut:

<https://github.com/danylaksono/sleman-dataset/>

Gunakan Github untuk melakukan clone atau unduh seluruh data sebagai zip

---

### 2.3.1 Pengantar PostGIS

Dalam implementasi GIS **generasi pertama**, semua data spasial disimpan dalam file datar dan perangkat lunak GIS khusus diperlukan untuk menafsirkan dan memanipulasi data. Sistem pengelolaan generasi pertama ini dirancang untuk memenuhi kebutuhan pengguna yang semua data yang diperlukan berada dalam domain organisasi pengguna. Mereka adalah sistem berpemilik dan mandiri yang secara khusus dibangun untuk menangani data spasial.

Sistem spasial **generasi kedua** menyimpan beberapa data dalam database relasional (biasanya “atribut” atau bagian non-spasial) tetapi masih kekurangan fleksibilitas yang diberikan dengan integrasi langsung.

*“Database spasial sejati lahir ketika orang mulai memperlakukan fitur spasial sebagai objek database kelas satu.”*

Database spasial sepenuhnya mengintegrasikan data spasial dengan database relasional objek. Orientasi berubah dari *GIS-centric* menjadi *database-centric*.

Untuk memanipulasi data selama kueri, database biasa menyediakan fungsi seperti menggabungkan string, melakukan operasi hash pada string, melakukan matematika pada angka, dan mengekstraksi informasi dari tanggal. Database spasial menyediakan satu set lengkap fungsi untuk menganalisis komponen geometris, menentukan hubungan spasial, dan memanipulasi geometri. Fungsi spasial ini berfungsi sebagai blok bangunan untuk setiap proyek tata ruang.

Mayoritas dari semua fungsi spasial pada PostGIS dapat dikelompokkan ke dalam salah satu dari lima kategori berikut:

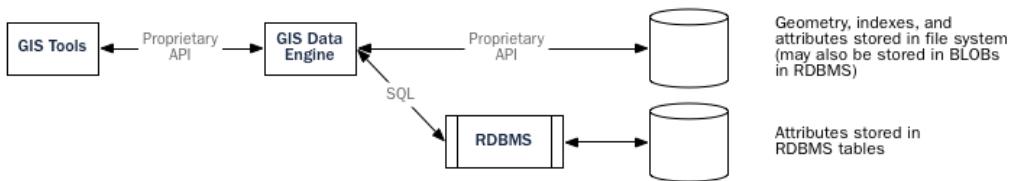
- **Konversi:** Fungsi yang mengonversi antara geometri dan format data eksternal.
- **Manajemen:** Fungsi yang mengatur informasi tentang tabel spasial dan administrasi PostGIS.
- **Retrieval:** Fungsi yang mengambil properti dan pengukuran Geometri.
- **Perbandingan:** Fungsi yang membandingkan dua geometri sehubungan dengan hubungan spasialnya.
- **Pembuatan data:** Fungsi yang menghasilkan geometri baru dari orang lain. Daftar fungsi yang mungkin sangat besar, tetapi sekumpulan fungsi umum ditentukan oleh OGC SFSQL dan diimplementasikan (bersama dengan fungsi tambahan yang berguna) oleh PostGIS.

### Evolution of GIS Architectures

#### First-Generation GIS:



#### Second-Generation GIS:



#### Third-Generation GIS:

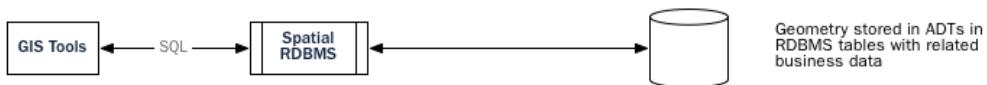
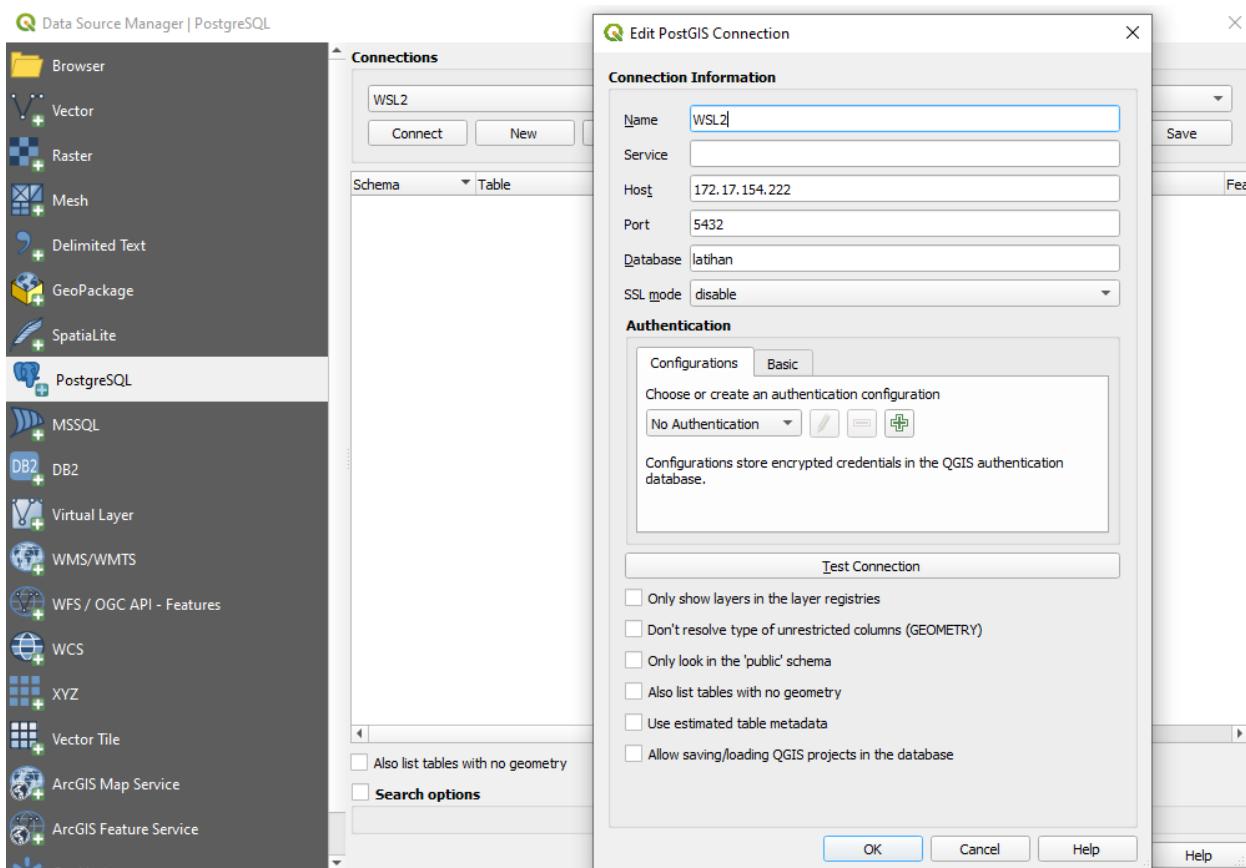


Fig. 2.8: Evolusi Basisdata Spasial untuk GIS

### 2.3.2 Menggunakan QGIS sebagai antarmuka PostGIS

QGIS barangkali merupakan perangkat lunak GIS OpenSource yang paling terkenal. QGIS memiliki berbagai dukungan format data spasial, termasuk untuk koneksi terhadap basisdata PostGIS. Pada latihan ini akan digunakan QGIS untuk melakukan koneksi pada basisdata.

QGIS memiliki berbagai opsi untuk terhubung dengan beragam format data yang berbeda. Koneksi ke basisdata PostgreSQL (dan dengan demikian PostGIS), dapat dilakukan melalui menu Data Source Manager sebagai berikut:



Lakukan koneksi pada basisdata PostGIS yang tersimpan pada WSL. Gunakan menu connect untuk menyambungkan basisdata sesuai dengan parameter yang diperlukan. Untuk saat ini, basisdata ini masih kosong tanpa ada isi apapun. Kita akan gunakan data dari unduhan di atas untuk mengisi basisdata ini dengan menggunakan QGIS.

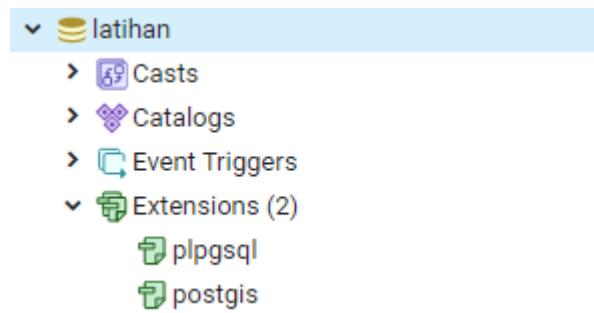
#### Latihan: Memasukkan Data pada PostGIS

Berikut adalah data yang tersedia pada link unduhan yang disebutkan di atas: Data vector yang digunakan berupa data di daerah Kabupaten Sleman, antara lain: \* Data jaringan jalan (line) \* Data jaringan sungai (line) \* Data bangunan (polygon) \* Data titik-titik penting (point) \* Data batas administrasi (polygon) \* Data penggunaan lahan (multipolygon) \* Data Kawasan Rawan Bencana (KRB) BNPB 2010 (polygon) Data berupa raster terdiri atas: \* Raster populasi penduduk Indonesia dengan resolusi 100 meter yang diambil dari [www.worldpop.org.uk](http://www.worldpop.org.uk) \* Raster SRTM dengan resolusi 30 meter.

Untuk latihan ini, terlebih dahulu kita masukkan data-data tersebut di atas ke dalam Basisdata PostGIS. Kita akan menggunakan basisdata latihan yang kita buat pada latihan sebelumnya. Akan tetapi, kita perlu untuk terlebih dulu mengaktifkan ekstensi PostGIS pada basisdata latihan. Gunakan `psql` atau `pgAdmin4` untuk menjalankan query berikut:

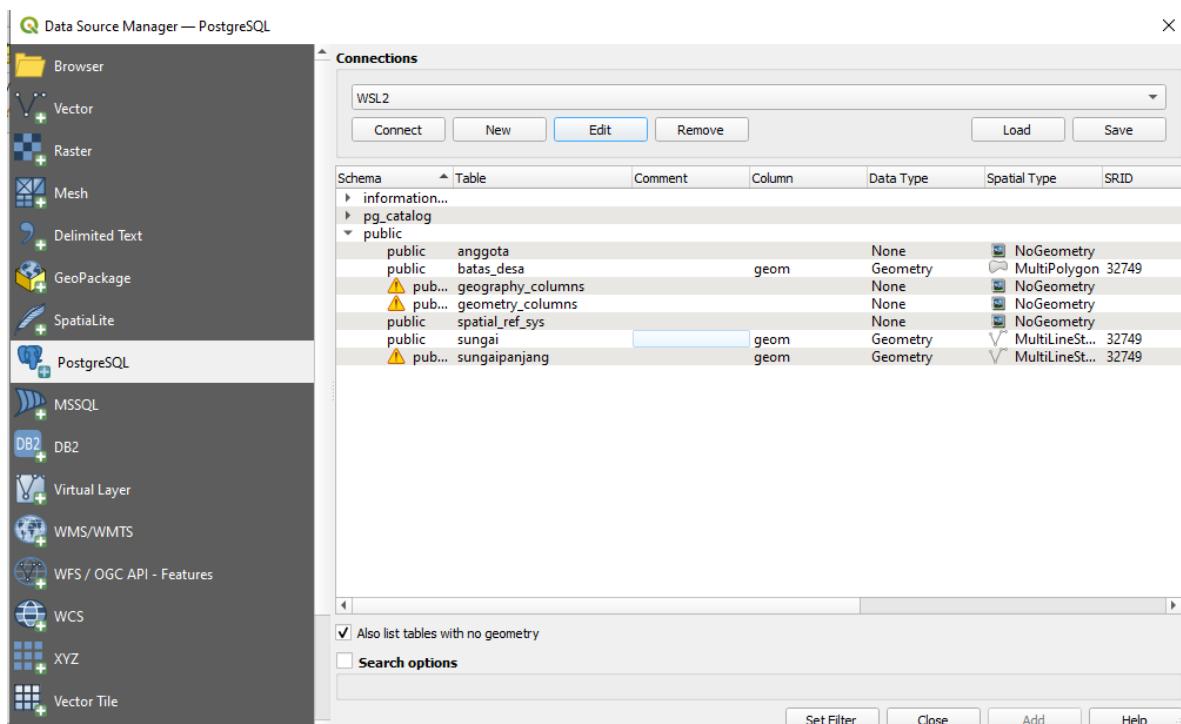
```
CREATE EXTENSION postgis;
```

Kita dapat memeriksa bahwa ekstensi postgis telah diaktifkan pada basisdata ini menggunakan pgAdmin:



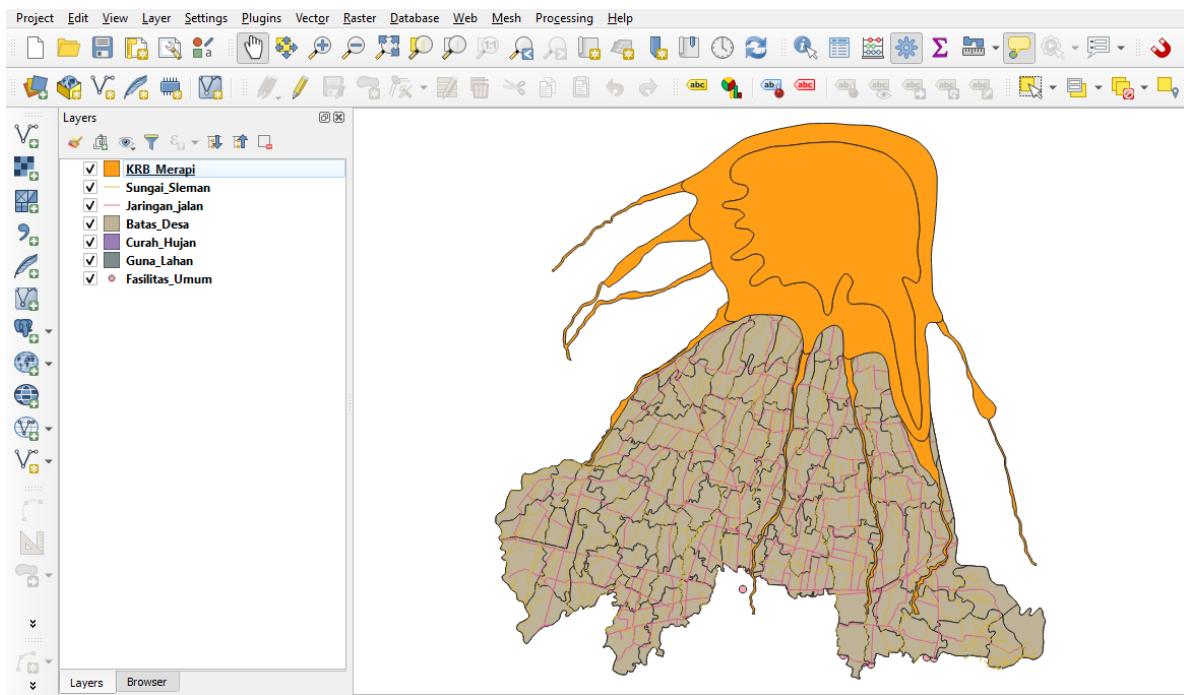
Dengan demikian, basisdata latihan sudah dapat digunakan untuk menyimpan dan menganalisis data geospasial. Untuk itu kita akan masukkan data vektor dan data raster di atas ke dalam basisdata. QGIS memiliki menu yang praktis untuk keperluan ini.

- Untuk data vektor, kita akan gunakan QGIS untuk memasukkan data pada PostGIS. Pastikan bahwa basisdata telah terkoneksi melalui menu Data Source Manager pada langkah sebelumnya.

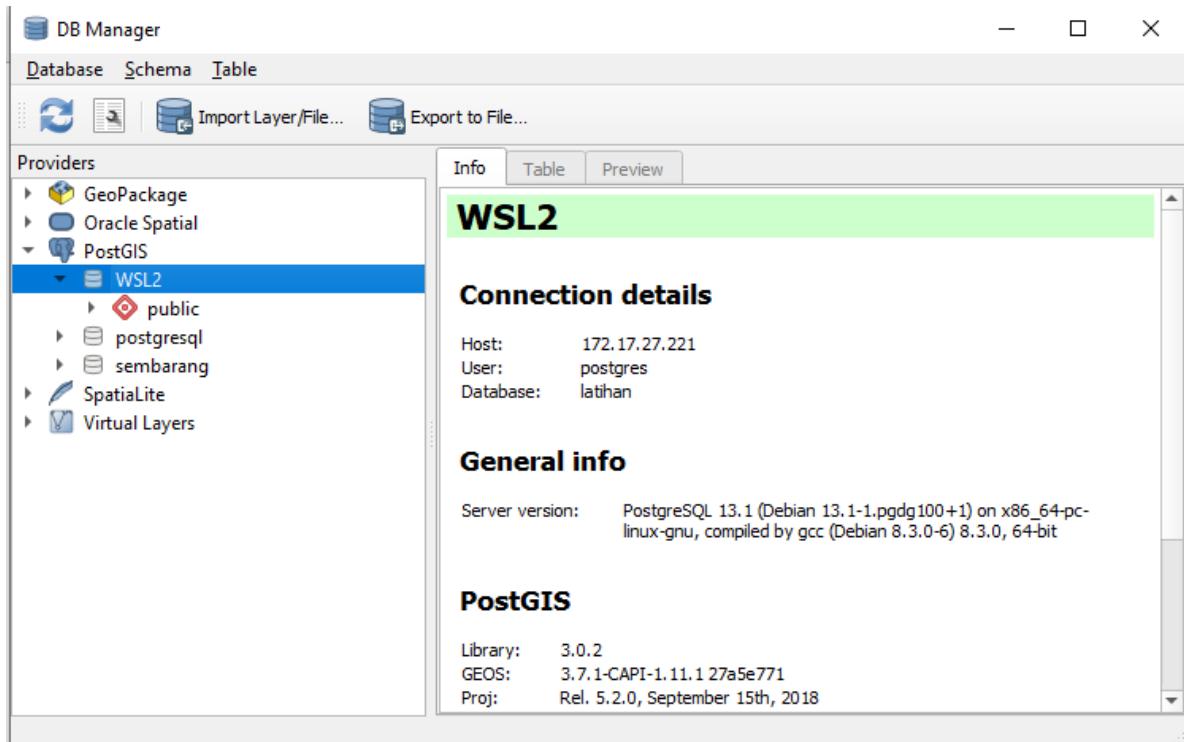


Centang also list tables with no geometry akan memunculkan tabel anggota yang kita gunakan pada latihan sebelumnya. Parameter koneksi pada antarmuka tersebut sama dengan parameter koneksi yang diperlukan untuk menghubungkan pgAdmin dan PostgreSQL pada latihan sebelumnya.

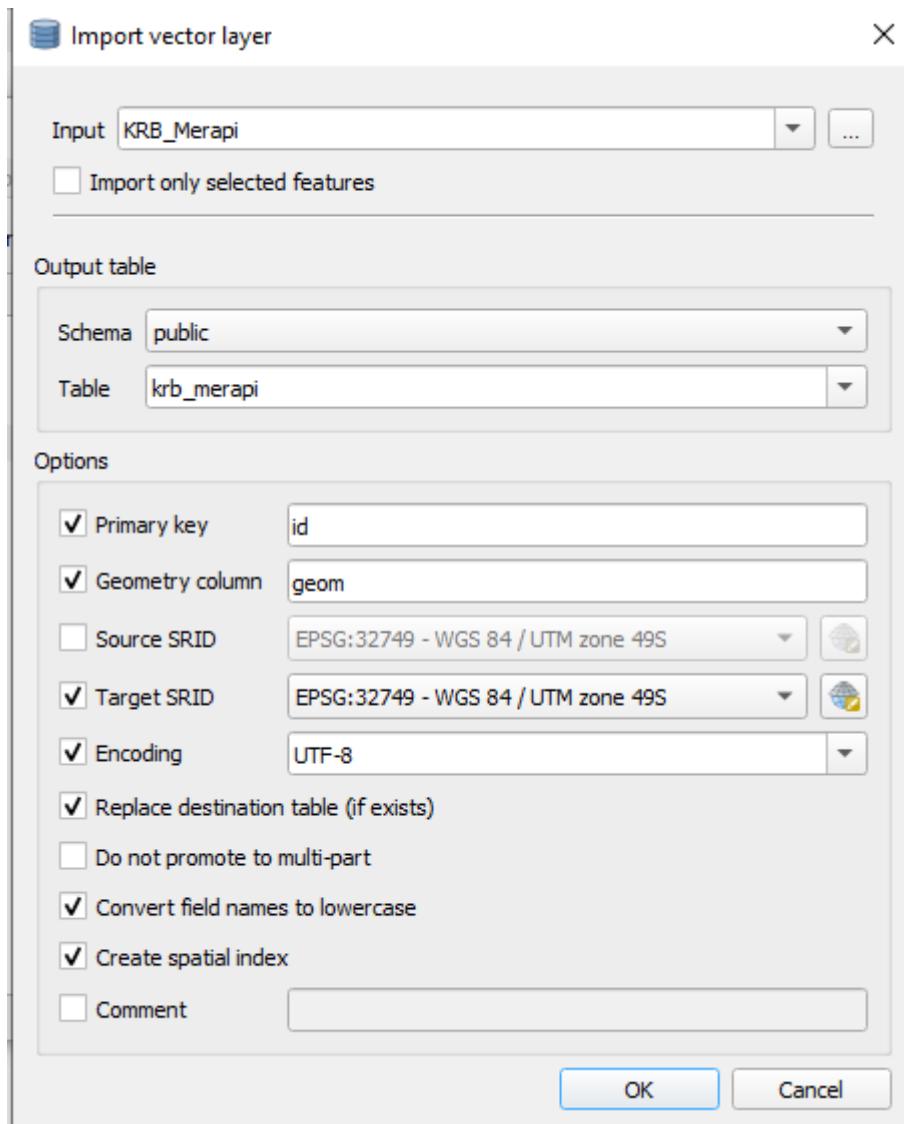
- Selanjutnya, tambahkan tiap layer **shapefile** pada data unduhan ke dalam QGIS, seperti berikut:



3. Buka menu **DBManager**, kemudian lakukan koneksi ke basisdata (cukup dengan membuka/expand koneksi basisdata yang dimaksud)



4. Gunakan menu Import Layer/File... untuk mengkonversi layer pada QGIS menjadi tabel pada PostGIS, seperti berikut.



Penggunaan Spatial Index akan sangat membantu apabila data yang digunakan cukup besar.

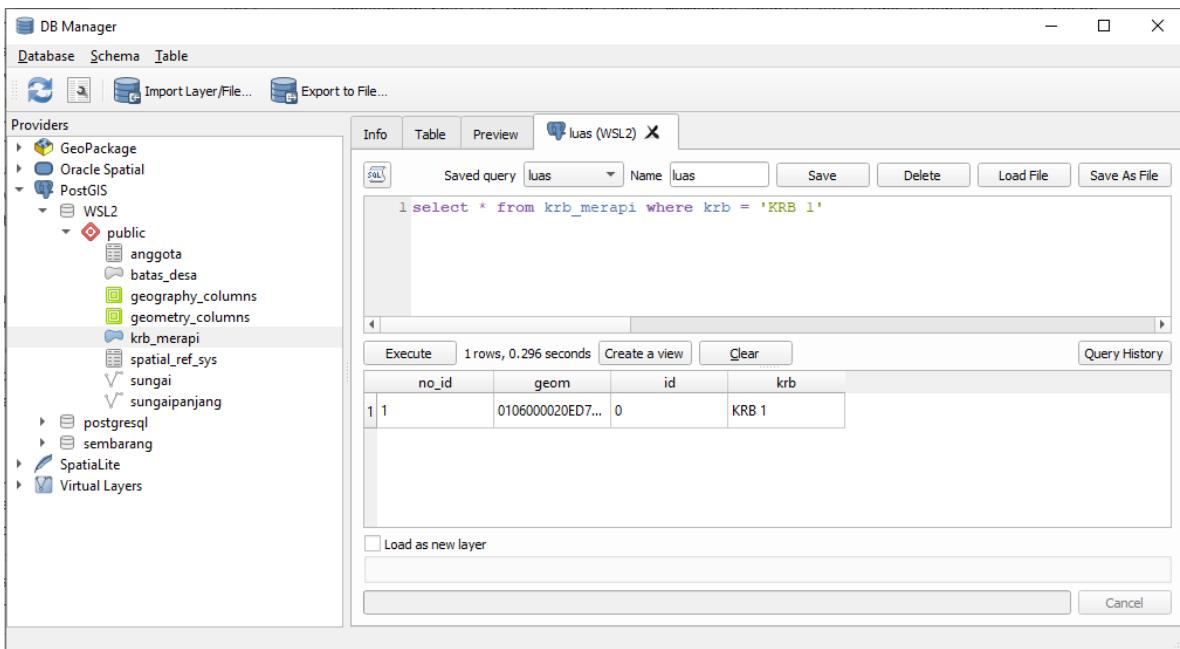
---

### Catatan

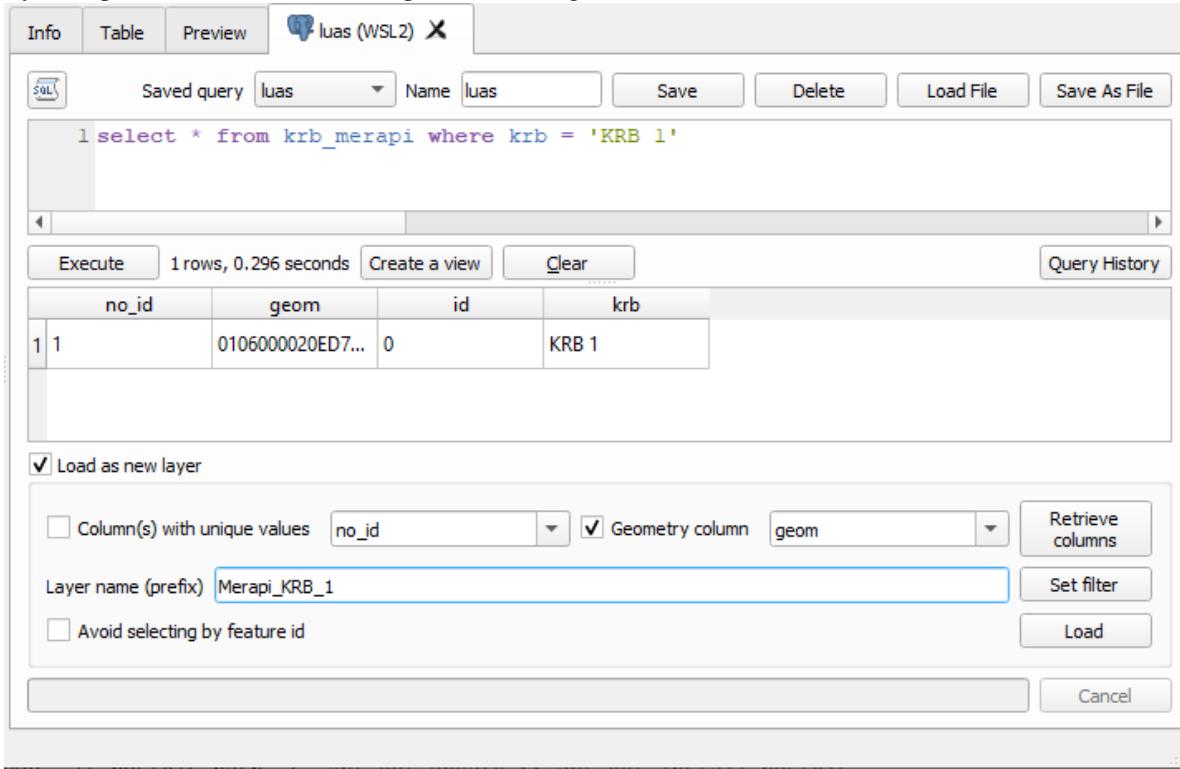
Karena suatu alasan, QGIS sepertinya tidak mendukung penggunaan huruf besar pada nama tabel maupun kolom. Gunakan huruf kecil pada nama **Table** dan centang **Convert field names to lowercase** untuk memastikan tidak ada masalah pada saat menjalankan SQL nantinya.

---

5. Gunakan menu SQL Window untuk memasukkan Query:



6. Hasil Query dapat disimpan dalam bentuk View (tabel baru) maupun layer pada QGIS. Klik pada centang Load as New Layer



Klik Load untuk memuat layer tersebut pada QGIS

Pada pembahasan di bawah ini terdapat contoh-contoh penggunaan SQL pada PostGIS untuk menyelesaikan berbagai permasalahan. Sebelum itu, kita perlu memasukkan terlebih dahulu data raster ke dalam basisdata PostGIS. Untuk menambahkan raster ke PostGIS kita dapat menggunakan fungsi `raster2pgsql`.

- Untuk data raster, terlebih dahulu kita perlu mengakses mesin Docker dengan mengcopikan data pada root mesin

tersebut, sebagai berikut:

```
docker exec -it postgis bash -c "apt-get update && apt-get install postgis"
docker cp dem.tif postgis:/
```

ini apabila kita asumsikan data raster yang kita copykan bernama dem.tif.

2. Selanjutnya, gunakan raster2pgsql untuk mengkonversi raster menjadi sql melalui operasi *pipe*:

```
docker exec -it postgis bash -c 'raster2pgsql -s 32749 -I -C -M dem.tif -F -t_
→100x100 public.rstdem | psql -U postgres -d latihan'
```

Dengan demikian, raster2pgsql akan mengkonversi data dem.tif menjadi tabel rstdem pada basisdata latihan.

Setelah semua data selesai kita muat pada QGIS, saatnya meng-eksplorasi berbagai fungsi analisis PostGIS. Berikut adalah beberapa contoh penggunaan fungsi pada PostGIS untuk keperluan analisis spasial pada data vektor maupun raster.

### Latihan: Query Spasial pada data vektor

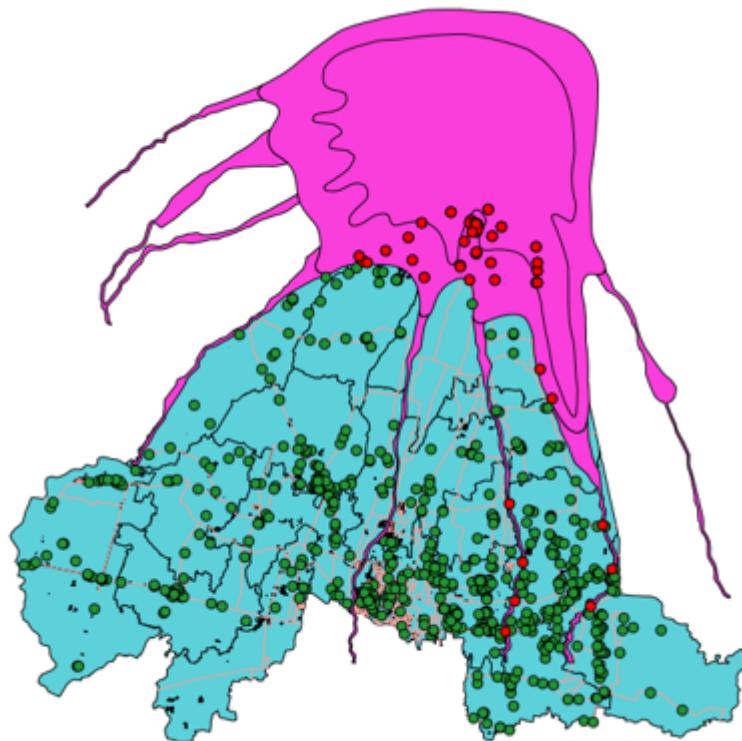
Berikut adalah beberapa latihan yang dapat dilakukan menggunakan QGIS dan PostGIS. Gunakan DBManager (Menu Database > DBManager) pada QGIS untuk mengakses PostGIS,

Selanjutnya, buat query-query berikut ini. Sesuaikan atribut dan nama kolom dengan data yang digunakan.

- Mencari titik-titik penting (POI) yang berada pada kawasan rawan bencana:

```
select poi.id_0, poi.geom
from poi, krb
where st_within(poi.geom, krb.geom) = True
```

hasilnya,



- Buffer jalan utama sejauh 100 meter

```
select id, name, st_buffer(geom, 100) from jalan
where type = 'primary'
```

- Menampilkan jumlah bangunan pendidikan di berbah

```
select count(id_0) as jumlah
from poi
where st_within(poi.geom, (
select geom
from kecamatan
where kec = 'Berbah')) and fungsi = 'Pendidikan'
```

- Total Panjang Jalan utama di Kecamatan Berbah

```
select sum(st_length(jalan.geom)) as total
from jalan, kecamatan
where jalan.type = 'primary' and st_intersects(jalan.geom,
(select geom from kecamatan where kec = 'Berbah'))
```

- Mencari jalan yang berjarak 5 kilometer dari STIE YKPN

```
select jalan.id, jalan.name, jalan.type
from jalan, poi
where ST_DWithin(poi.geom, jalan.geom, 5000)
and poi.nama = 'STIE YKPN'
```

- Mencari kecamatan paling luas di Kabupaten Sleman

```
select kec, st_area(geom) as luas
from kecamatan
where st_area(geom) in
(
    select max(st_area(geom))
    from kecamatan
)
```

- Menampilkan ruas jalan terpanjang di Kecamatan Berbah

```
select name, st_length(geom) as panjang
from jalan
where st_length(geom) in
(
    select max(st_length(jalan.geom))
    from jalan, kecamatan
    where st_intersects(jalan.geom, kecamatan.geom) and kec='Berbah'
)
```



- Mencari kecamatan yang dilewati oleh Jalan Kabupaten

```
select kec
from kecamatan, jalan
where st_intersects(jalan.geom,kecamatan.geom)
and jalan.geom in (select geom
from jalan
where name='Jln. Kabupaten')
```

SQL window - tugasbds [PostGIS]

SQL query: query15

```
select kec
from kecamatan, jalan
where st_intersects(jalan.geom,kecamatan.geom)
and jalan.geom in (select geom
from jalan
where name='Jln. Kabupaten')
```

Execute (F5) 3 rows, 0.0 seconds

Result:

kec
1 Gamping
2 Mlati
3 Sleman

Load as new layer

X Close

- Mencari luas Hutan

```
select st_area(geom) as luas from gunalahan
where guna_lahan='HUTAN'
```

SQL window - tugasbds [PostGIS]

SQL query:

```
select st_area(geom) as luas from gunalahan
where guna_lahan='HUTAN'
```

Execute (F5) 1 rows, 0.0 seconds

Result:

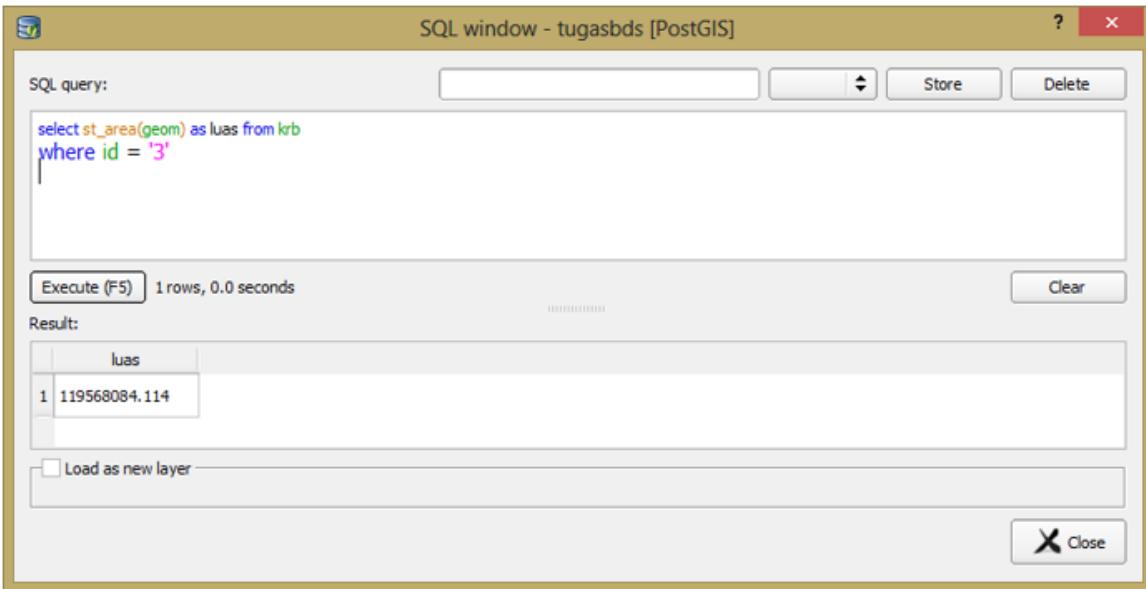
luas
1 9132834.09429

Load as new layer

X Close

- Mencari jumlah bangunan di area KRB III

```
select st_area(geom) as luas from krb
where id = '3'
```



Untuk latihan lanjutan, coba lakukan Query berikut:

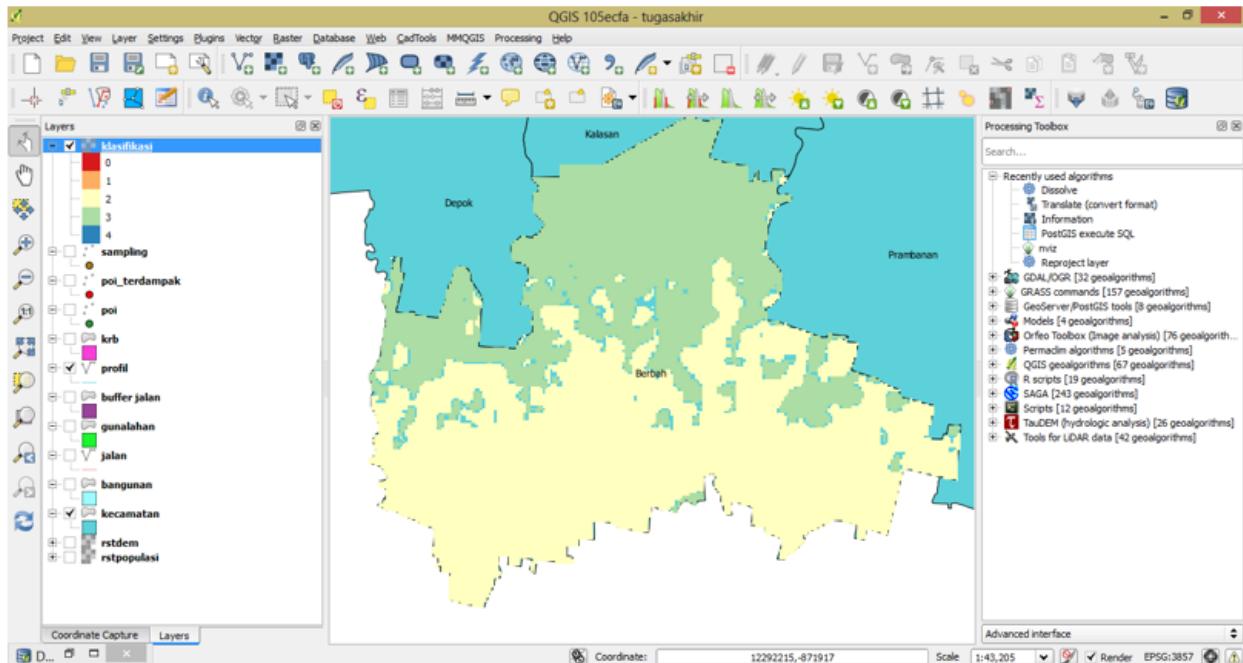
- Menampilkan jumlah titik penting yang berada di kecamatan Berbah
- Menampilkan Bangunan dengan luas lebih dari 1500 m<sup>2</sup>
- Menampilkan keliling Kecamatan Cangkringan
- Menampilkan panjang jalan kaliurang
- Menampilkan panjang total semua sungai

### Latihan: Query Spasial pada Raster

Berikut contoh analisis raster dengan PostGIS:

- Membuat tingkat kepadatan penduduk pada kecamatan Berbah

```
CREATE TABLE klasifikasi (rid SERIAL primary key, rast raster);
INSERT INTO klasifikasi (rast)
with Berbah as
    (Select geom from kecamatan where kec='Berbah'),
    klip as
        (Select ST_Clip(rast, 1, geom, true) as rast from rstdem, berbah where st_
        intersects(geom,rast))
select st_reclass(rast, 1, '0-50:1, 51-100:2, 101-150:3, 151-200:4', '4BUI', 0) from_
        klip;
SELECT AddRasterConstraints('klasifikasi'::name, 'rast'::name);
```



- Rencana Lokasi perumahan di Kecamatan Kalasan Dibuat kondisi sebagai berikut:
- Penggunaan lahan berupa sawah irigasi atau tegalan
- Berada pada daerah Kecamatan Kalasan
- Kemiringan lereng tidak lebih dari 0.1 radian Dibuat query sebagai berikut:

```

CREATE TABLE perumahan (rid SERIAL primary key, rast raster);
INSERT INTO perumahan(rast)

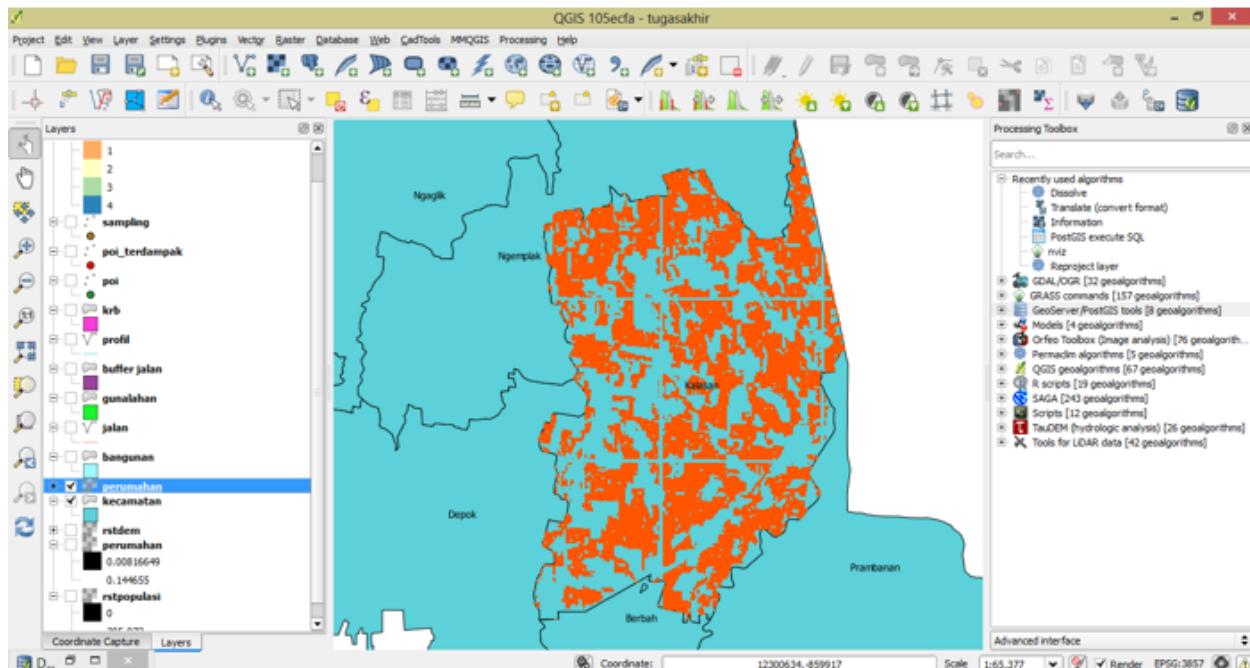
with kalasan as
    (select geom from kecamatan where kec='Kalasan'),
    sawah as
        (select geom from gunalahan where guna_lahan='SAWAH IRIGASI' or guna_
        lahan='TEGALAN'),
    potong1 as
        (select ST_Clip(rast, 1, geom, true) as rast from rstdem, kalasan_
        where st_intersects(geom,rast)),
    lereng as
        (select ST_Slope(rast, 1, '32BF') as rast from potong1),
    kelas_lereng as
        (select st_reclass(rast, 1, '0-0.09:1', '4BUI', 0) as rast from_
        lereng)

SELECT ST_Clip(rast, 1, geom, true) from kelas_lereng, sawah where st_-
intersects(geom,rast);

SELECT AddRasterConstraints('perumahan)::name, 'rast)::name);

```

hasilnya:



## PENGANTAR INTEROPERABILITAS DATA SPASIAL

Proses bisnis Sistem Informasi Geospasial saat ini tidak terbatas pada aktivitas akuisisi data, penyimpanan, pengolahan dan visualisasi pada sebuah sistem tunggal. Kebutuhan tukar-guna data semakin besar pada sebuah ekosistem Enterprise GIS, sehingga diperlukan adanya mekanisme yang memungkinkan data spasial dengan segala propertinya. Pada bagian ini akan dibahas mengenai konsep interoperabilitas dan implementasinya menggunakan perangkat server data spasial.

Bagian ini membahas mengenai:

*Interoperabilitas dan Standar OGC* : Konsep interoperabilitas data spasial dengan standar OGC dan instalasi server data spasial menggunakan Geoserver

*Publikasi data spasial dengan OGC* : Bagaimana mengatur geoserver untuk publikasi data spasial sesuai dengan standar OGC, mengatur sistem proyeksi data, dan membuat tiling untuk data citra satelit

*Pengaturan dan Konsumsi data spasial melalui OGC* : Setelah melakukan publikasi data pada OGC, beberapa client berbeda (QGIS, LeafletJS, OpenLayers, Mapstore) digunakan untuk memanggil data tersebut

### 3.1 Interoperabilitas dan Standar OGC

Kebutuhan untuk berbagi pakai data spasial semakin meningkat seiring dengan berkembangnya kebutuhan akan data dan informasi geospasial itu sendiri. Dengan beragamnya format data dan vendor aplikasi yang digunakan di berbagai instansi, perlu adanya suatu standar yang dapat digunakan untuk menjamin agar data spasial yang dibagikan antar instansi dapat digunakan secara optimal. Pada bagian ini akan dibahas mengenai standar pertukaran data spasial untuk menjamin interoperabilitas data spasial

#### 3.1.1 Kebijakan Satu Peta dan Interoperabilitas Data Spasial

Wacana mengenai kebijakan satu peta telah dimulai sejak tahun 2010 di Indonesia. Meskipun Indonesia merupakan salah satu negara *early-adopters* untuk infrastruktur data spasial, baru pada tahun 2011 melalui Undang-Undang Informasi Geospasial (UUIG) wujud Infrastruktur Data Spasial yang digunakan untuk berbagi pakai data antar instansi ini mulai terlihat.

### Interoperabilitas Data Spasial

Berbagai instansi di Indonesia menggunakan data spasial pada berbagai level. Sebagai produser atau pemilik data spasial pada domain masing-masing, tiap instansi menggunakan format data dan perangkat lunak yang berbeda. Hal ini berakibat pada sulitnya berbagi pakai data antar instansi. Terlebih lagi, data spasial memiliki karakteristik yang berbeda dibandingkan dengan data biasa, seperti adanya sistem proyeksi, geometri, topologi yang mendefinisikan data tersebut. Pertukaran data pada berbagai vendor yang berbeda dengan format data masing-masing berpotensi mengakibatkan *loss of information*, dan dalam kasus data spasial, dapat menyebabkan pengambilan keputusan yang salah.

*Fair Principle (Findable, Accessible, Interoperable dan Reuseable)* digunakan untuk menjamin agar data spasial dapat digunakan oleh berbagai aktor yang berbeda. Prinsip FAIR memungkinkan format data dari berbagai instansi yang berbeda untuk dapat digunakan tanpa hilangnya informasi penting terkait data tersebut. Pada data spasial, implementasi FAIR principle merupakan kebutuhan utama pada Infrastruktur Data Spasial, dimana data geospasial antar wali data yang berbeda dapat saling digunakan satu dengan yang lain.

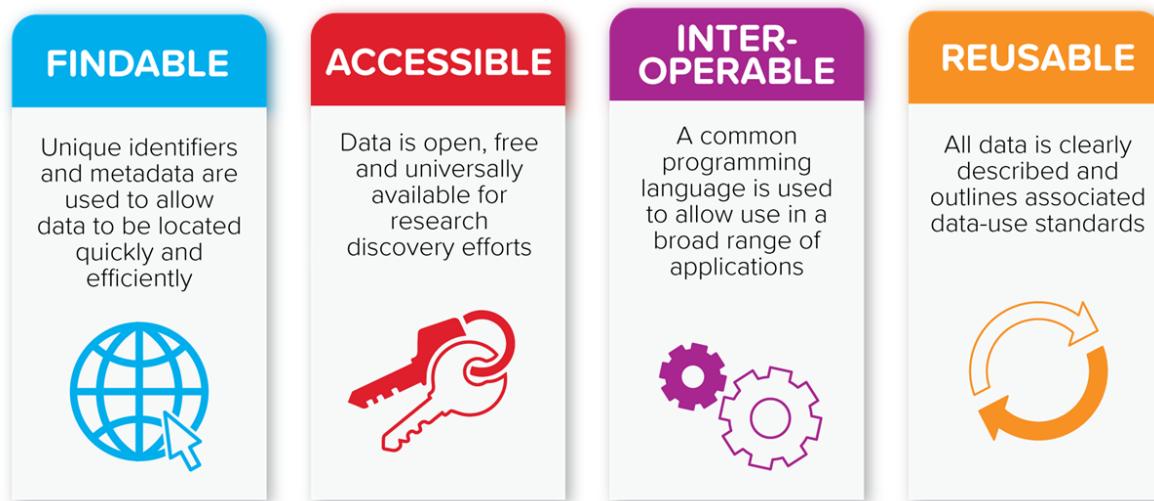


Fig. 3.1: Prinsip FAIR

### 3.1.2 Standar OGC

Standar OGC dikembangkan oleh para anggota untuk membuat informasi dan layanan geospasial memenuhi prinsip FAIR - Findable, Accessible, Interoperable, dan Reuseable. Standar digunakan oleh pengembang perangkat lunak untuk membangun antarmuka terbuka dan pengkodean ke dalam produk dan layanan masing-masing vendor. Standar adalah “produk” utama OGC dan telah dikembangkan oleh member OGC untuk mengatasi tantangan interoperabilitas tertentu, seperti menerbitkan konten peta di Web, bertukar data lokasi penting selama tanggap bencana & pemulihan, dan memungkinkan penggabungan informasi dari beragam konten Internet of Things (IoT).

### OGC untuk interoperabilitas

OGC mengikuti Spesifikasi OpenGIS yang mendefinisikan bagaimana data spasial disimpan dan direpresentasikan. Standar Simple Feature Specification ini menjadi acuan bagi OGC dan berbagai perangkat lunak yang mendefinisikan data spasial sehingga memungkinkan satu interpretasi yang sama pada fitur yang direpresentasikan oleh data spasial yang dimaksud. Simple Feature didefinisikan sebagai berikut:

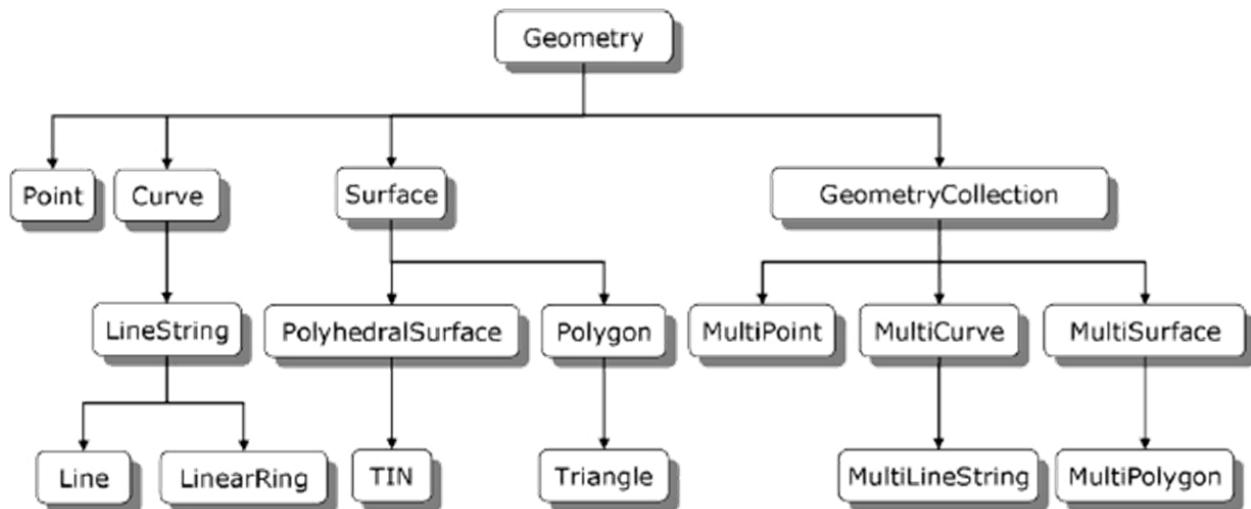


Fig. 3.2: Simple Feature Specification

Berbagai standar OGC dibuat untuk memungkinkan data-data spasial yang beragam dari berbagai domain untuk dapat digunakan mengikuti prinsip FAIR sesuai dengan format yang juga didefinisikan oleh OGC. Dengan tujuan untuk memungkinkan data spasial dapat disajikan melalui internet, standar ini juga menjamin prinsip FAIR melalui penggunaan metadata dan data dalam bentuk layanan (*services*). Layanan inilah yang digunakan untuk meng-onlinekan data spasial agar dapat digunakan oleh aktor yang berbeda agar dapat menggunakan data spasial dengan semua karakteristiknya.

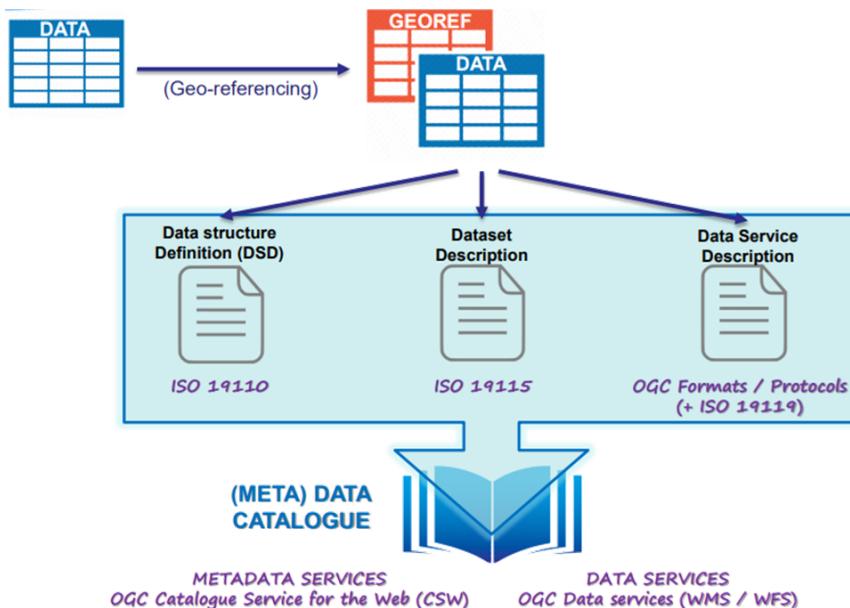


Fig. 3.3: OGC Data & Metadata Services

Terdapat puluhan standar OGC berbeda yang digunakan untuk berbagai keperluan. Standar untuk layanan dan format data diatur tersendiri. Demikian pula standar untuk memungkinkan Discovery atau pencarian data spasial pada portal untuk menjamin prinsip Findable dan Accessible. Berikut adalah standar terkait dengan Discovery, Container, serta layanan dan API.

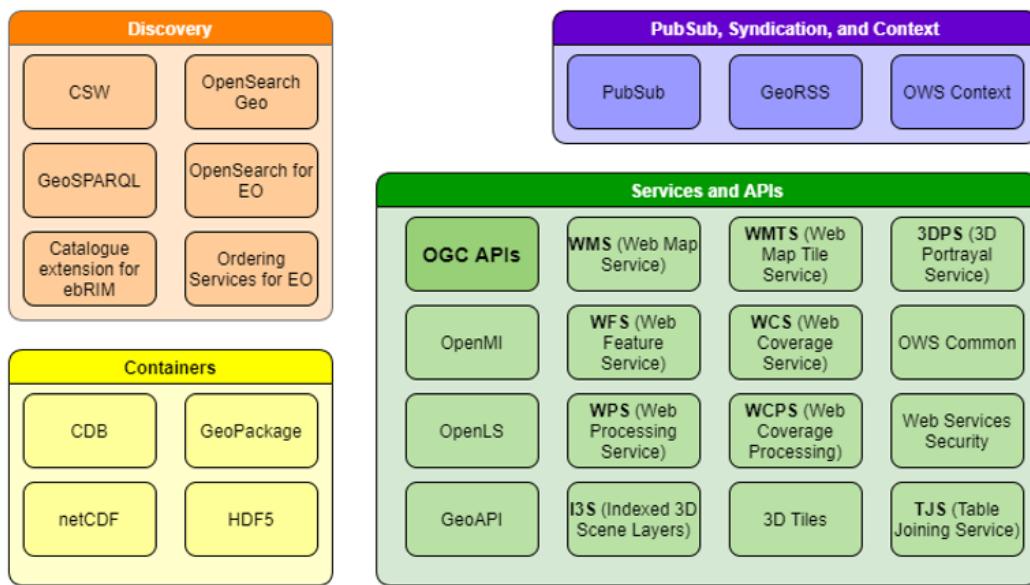


Fig. 3.4: Standar OGC untuk Discovery, Container, Service dan API

Berikut standar OGC untuk Model data dan Sensor:

Tiap standar tersebut memiliki fungsi dan fitur yang berbeda. Pada bagian selanjutnya akan dibahas mengenai WMS, WFS dan WCS yang merupakan standar yang umum dijumpai saat ini.

### WMS WFS dan WCS

Dari ratusan standar yang berbeda, beberapa standar digunakan lebih sering dibandingkan yang lain karena kebutuhan untuk representasi data menggunakan standar ini cukup besar. WMS digunakan untuk pertukaran data vektor dan raster, WFS untuk data vektor dan WCS untuk data raster. Ketiga standar ini memiliki spesifikasi yang berbeda untuk digunakan pada kebutuhan yang berbeda pula.

- **Web Map Service (WMS)** WMS merupakan standar yang dibuat untuk memungkinkan akses data spasial baik dalam format raster maupun vektor untuk disajikan sebagai Image. Pada data masukan berupa raster, WMS mengkonversi nilai pixel dan format data masukan menjadi format data yang lebih ringkas untuk digunakan melalui web (misalnya png atau jpg) sehingga nilai pixel aslinya hilang.

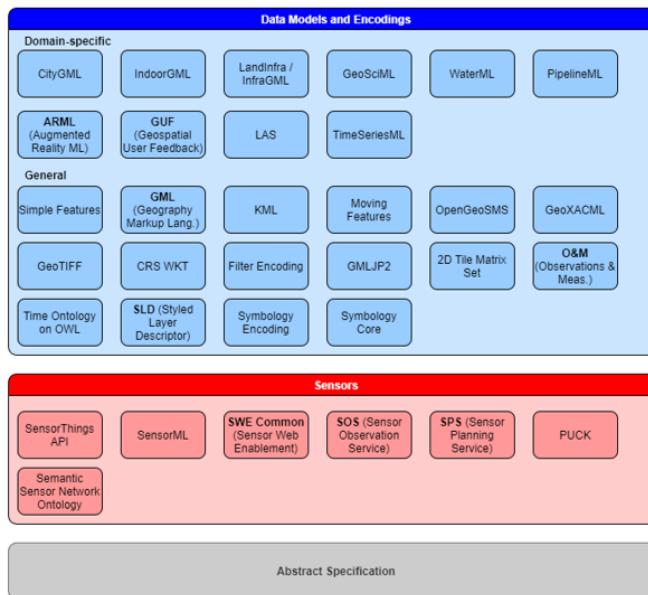
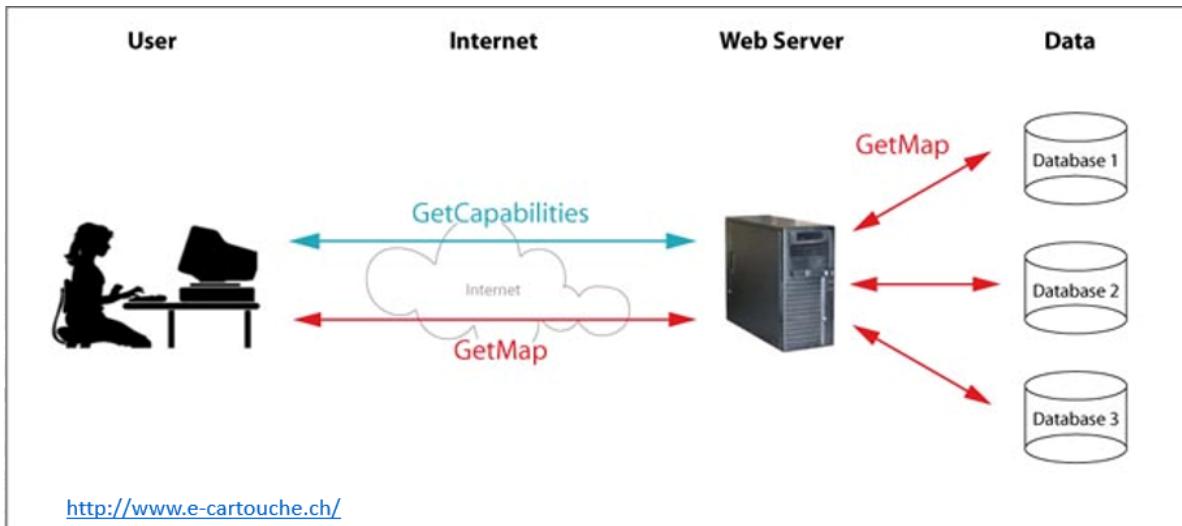
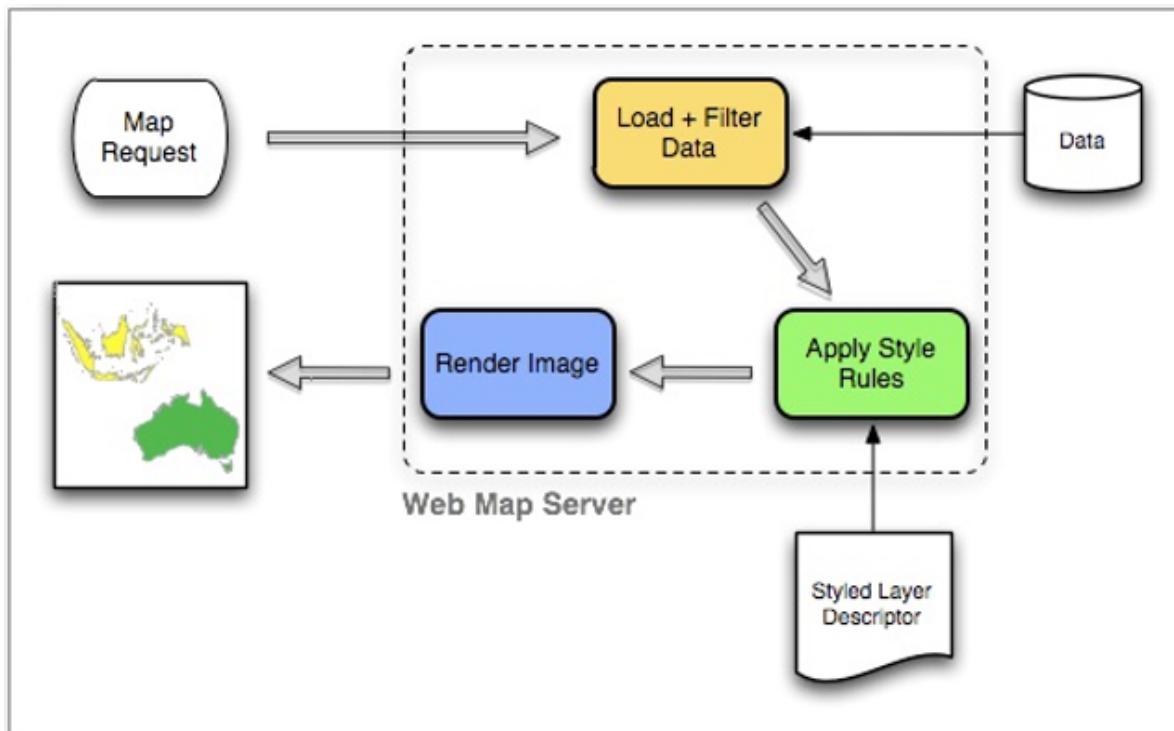


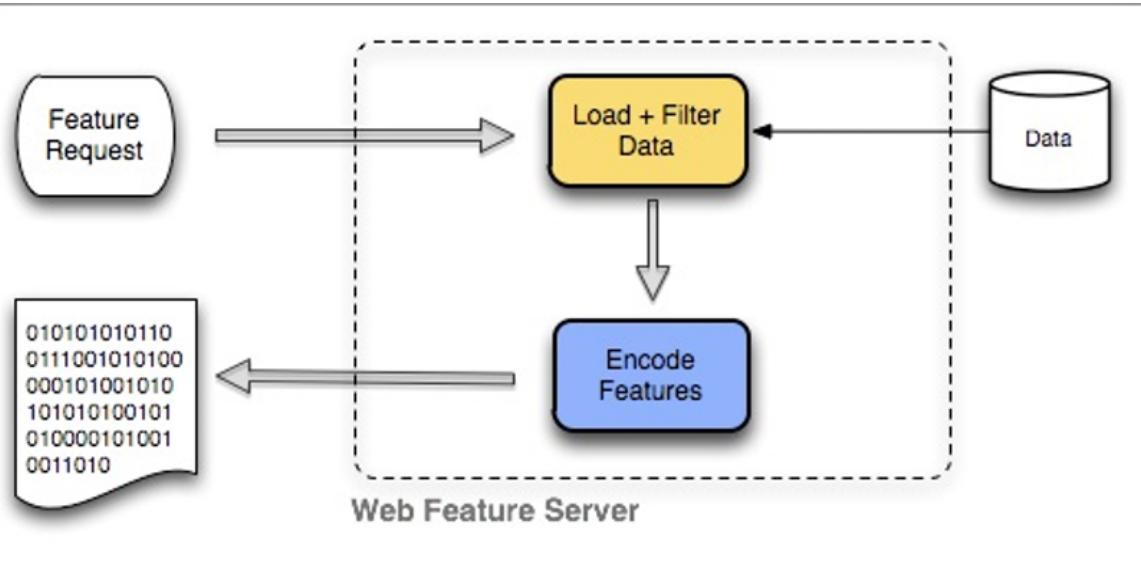
Fig. 3.5: Standar OGC untuk Model Data dan Sensor



Untuk data vector, WMS mengkonversi data tersebut menjadi raster untuk keperluan tampilan, dimana simbologi pada layer tampilan tersebut diperoleh berdasarkan styling yang dilakukan di sisi server.



- **Web Feature Service (WFS)** WFS digunakan untuk menyajikan data vector lengkap dengan semua propertiannya (atribut, geometri, sistem proyeksi, dst). WFS digunakan apabila client menginginkan untuk mendapatkan data vector secara penuh, misalnya untuk keperluan analisis spasial. Berbeda dengan WMS yang melakukan styling pada sisi server, WFS menyajikan data biner dalam bentuk vector kepada client, sehingga kebutuhan untuk melakukan simbolisasi diserahkan pada pengaturan client



- **Web Coverage Service (WCS)** WCS berbeda dengan WMS dalam hal penyajian informasi detil dari sebuah citra. Jika WMS menyederhanakan nilai pixel pada raster menjadi nilai RGB sesuai dengan tampilannya, maka WCS menjaga nilai pixel dan band dari raster masukan sehingga client dapat menggunakan data tersebut untuk analisis raster. Contoh penggunaan WCS adalah untuk menampilkan raster elevasi yang memungkinkan pengguna untuk melakukan analisis seperti slope atau aspect.

Berikut adalah contoh skenario penggunaan ketiga jenis layanan tersebut:

Sebagai contoh, dibuat sebuah WebGIS Desa yang memanfaatkan data dari berbagai sumber sebagai berikut:

- WMS Citra SPOT 6 desa dari Badan Informasi Geospasial sebagai latar belakang
- WMS Batas Desa dari BPS sebagai latar belakang
- WFS Jaringan Jalan dari PU untuk analisis buffer
- WFS bangunan desa dari server local untuk simbologi atribut
- WCS raster ketinggian (elevasi) DEMNAS dari BIG
- WCS raster cuaca terkini dari BIG

Dengan menggunakan layanan OGC, data dari berbagai sumber dapat digabungkan pada satu platform yang sama, sekali-gus memungkinkan analisis dan visualisasi dari data yang berbeda. Ini merupakan implementasi interoperabilitas pada Prinsip FAIR.

### 3.1.3 Moving Forward

Standar OGC mengatur pertukaran data spasial pada domain utama, khususnya terkait dengan berbagai format data baik raster maupun vektor. Seiring perkembangan teknologi, khususnya trend pengolahan data berbasis *cloud* dan Analysis Ready Data maupun perkembangan teknologi internet itu sendiri, terdapat beberapa standar lain yang saat ini berkembang untuk memungkinkan beragam aplikasi di luar standar yang disediakan oleh OGC saat ini. Beberapa diantaranya adalah seperti berikut:

#### Cloud Optimised GeoTIFF (COG)



Cloud Optimized GeoTIFF (COG) mirip dengan file GeoTIFF biasa tetapi dengan organisasi internal yang memungkinkan akses baca yang lebih efisien ke sebagian gambar, alur kerja yang paling umum untuk mengakses citra yang disimpan di infrastruktur cloud. Hal ini dilakukan dengan memanfaatkan kemampuan klien yang mengeluarkan permintaan rentang HTTP GET untuk meminta hanya bagian dari file yang mereka butuhkan. Teknologi yang sangat mirip yang memungkinkan Anda untuk langsung mulai menonton video streaming memungkinkan analisis pada GeoTIFF besar tanpa perlu mengunduh seluruh file. Sementara beberapa standar berbeda telah mengatasi masalah ini, COG berada pada posisi yang baik untuk menjadi standar komunitas dan industri. Dukungan untuk standar baru dengan cepat diintegrasikan ke dalam pustaka perangkat lunak sumber terbuka dan mempertahankan kompatibilitas ke belakang; sistem yang dapat melihat file TIFF biasa juga dapat melihat COG, meskipun sistem tersebut tidak memanfaatkan pembacaan rentang byte HTTP atau komponen geospasial. COG adalah GeoTIFFS adalah TIFF. Dalam waktu singkat sejak dikembangkan, format COG dengan cepat diadopsi oleh penyedia citra observasi Bumi karena ini merupakan cara untuk dengan mudah mengurangi biaya transfer jaringan dalam infrastruktur terdistribusi.

### Spatio Temporal Asset Catalog (STAC)



Spesifikasi SpatioTemporal Asset Catalog (STAC) bertujuan untuk menstandarkan bagaimana data geospasial diekspos secara online dan diquery. ‘Aset spasiotemporal’ adalah file apa pun yang mewakili informasi tentang bumi yang ditangkap dalam ruang dan waktu tertentu. Fokus awal utamanya adalah citra penginderaan jarak jauh (dari satelit, tetapi juga pesawat, drone, balon, dll.), tetapi intinya dirancang agar dapat diperluas ke SAR, FMV, Point-cloud, hiperspektral, LiDAR, dan data turunan seperti NDVI, Model Elevasi Digital, mozaik, dll.

Tujuan STAC adalah agar semua penyedia utama citra dan data observasi bumi lainnya mengekspos data mereka sebagai Katalog Aset SpatioTemporal, sehingga kode baru tidak perlu ditulis setiap kali REST API berbasis JSON baru keluar yang membuat datanya tersedia di dengan cara yang sedikit berbeda. Ini akan mengaktifkan komponen pustaka standar dalam banyak bahasa. STAC juga dapat diimplementasikan dengan cara yang sepenuhnya ‘statis’, memungkinkan penerbit data mengekspos data mereka hanya dengan menerbitkan file JSON yang ditautkan secara online.

Kedua standar tersebut mendukung penggunaan citra dan analisis citra pada cloud, tanpa perlu mengunduh satupun data pada komputer lokal. Rilis terbaru Landsat Collection 2 telah menggunakan standar COG dan STAC, dan menunjukkan arah penggunaan format data yang lebih terbuka untuk mendukung interoperabilitas data spasial.



Fig. 3.6: Landsat telah mengumumkan rilis untuk Collection 2 yang menggunakan STAC dan COG sebagai metode delivery data citra

## Standar Baru untuk Dunia Baru: OGC-API

Standar seperti WMS, WFS dan WCS dibuat berdasarkan standar dan teknologi internet yang berlaku pada awal tahun 2000an. Dengan perkembangan teknologi saat ini serta perubahan paradigma data spasial pada web, OGC mengusulkan serangkaian standar baru untuk menggantikan standar WMS, WFS, dst yang digunakan saat ini.

Berikut adalah beberapa standar OGC-API yang sedang diusulkan:

- OGC API - Common (OWS Common)
- OGC API - Features (former WFS)
- OGC API - Coverages (former WCS)
- OGC API - Maps (former WMS)
- OGC API - Tiles (former WMTS)
- OGC API - Processing (former WPS)
- OGC API - Records (former CSW)

OGC-API saat ini sedang dalam [tahap pengembangan aktif](#), dan prosedur migrasi dari standar lama (WMS, WFS, WCS, dst) sedang disiapkan. Beberapa aplikasi, seperti PyGeoAPI dikembangkan untuk keperluan ini. PyGeoAPI merupakan salah satu prototip aplikasi yang dikembangkan untuk mendukung Standar OGC API yang baru. Saat ini, PyGeoAPI telah diujicobakan dan berhasil digunakan untuk mendukung infrastruktur data spasial di beberapa negara.

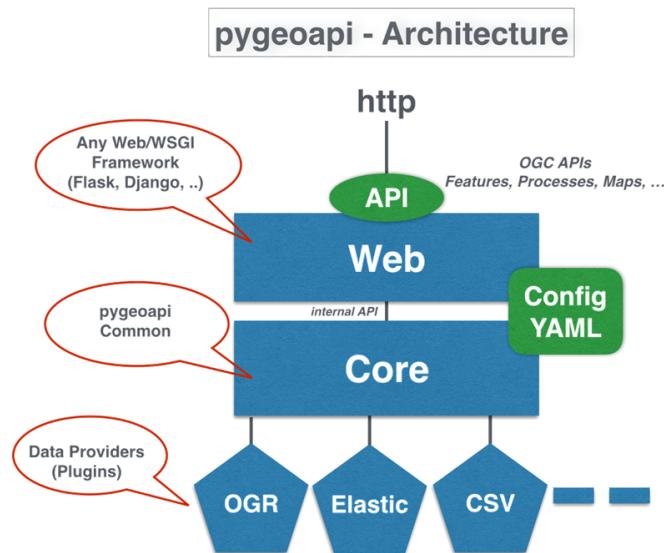


Fig. 3.7: Arsitektur PyGeoAPI

### 3.1.4 Server Data Spasial: Geoserver

Geoserver merupakan aplikasi Server Data Spasial yang mendukung publikasi berbagai macam jenis data spasial menjadi format yang didukung oleh OGC. Geoserver seringkali digunakan bersamaan dengan aplikasi lain sebagai bagian dari sebuah sistem informasi spasial, web GIS atau infrastruktur data spasial untuk memastikan data spasial dapat digunakan pada berbagai sistem yang berbeda dan menjamin interoperabilitas.

#### Latihan: Instalasi Geoserver

Pada bagian ini akan dibahas mengenai cara instalasi Geoserver: melalui aplikasi servlet seperti Tomcat, serta menggunakan paket aplikasi yang tersedia pada Docker.

- **Cara 1: menggunakan Docker** Pada bagian ini, akan digunakan Docker sebagai media instalasi Geoserver. Sebelumnya telah dibahas bagaimana Docker dapat mempermudah instalasi berbagai aplikasi. Berikut adalah panduannya:

1. Dengan cara pertama ini, dapat digunakan berbagai Image Geoserver yang tersedia pada Docker Hub. Untuk latihan ini akan digunakan image yang dibuat oleh Kartoza, sama seperti PostGIS pada materi sebelumnya.
2. Jalankan perintah berikut untuk menginstall Geoserver pada port 8080 menggunakan Docker:

```
docker run -d -p 8080:8080 --name geoserver kartoza/geoserver:latest
```

Maka Geoserver akan diunduh dan dijalankan pada Localhost:8080.

Cara ini akan mengunduh Geoserver berikut seluruh kelengkapannya (extensi standar maupun community, dokumentasi dan sampel data), sehingga ukuran unduhan akan cukup besar. Untuk instalasi Geoserver dengan ukuran yang lebih kecil untuk keperluan latihan ini, gunakan Cara Kedua di bawah ini.

- **Cara 2: Melalui Tomcat** Pada latihan ini, akan dibuat instalasi baru Geoserver dengan menggunakan Tomcat yang telah diinstall pada latihan berikutnya, sehingga terlebih dahulu perlu dipastikan bahwa instalasi Tomcat telah berhasil dan Tomcat berhasil dijalankan.

1. Unduh Geoserver dari website resminya: <http://geoserver.org/release/stable/> Untuk latihan ini, yang diunduh adalah build Geoserver dengan format Web Archive (WAR). Karena unduhan akan dilakukan pada mesin WSL, maka cukup copykan link yang mengarah pada Web Archive tersebut:



[Web Archive](#)

Web Archive (war) for servlet containers.

dalam hal ini, linknya adalah:

<http://sourceforge.net/projects/geoserver/files/GeoServer/2.18.1/geoserver-2.18.1-war.zip>

2. Buka WSL, kemudian pada folder dimana kita memiliki akses untuk menulis, jalankan perintah berikut

```
wget http://sourceforge.net/projects/geoserver/files/GeoServer/2.18.1/geoserver-2.18.1-war.zip
```

```
danylaksono@DANYLAKSONO:~$ wget http://sourceforge.net/projects/geoserver/files/GeoServer/2.18.1/geoserver-2.18.1-war.zip
--2020-12-04 06:53:23-- http://sourceforge.net/projects/geoserver/files/GeoServer/2.18.1/geoserver-2.18.1-war.zip
Resolving sourceForge.net (sourceforge.net)... 216.105.38.13
Connecting to sourceForge.net (sourceforge.net)[216.105.38.13]:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://sourceforge.net/projects/geoserver/files/GeoServer/2.18.1/geoserver-2.18.1-war.zip [following]
--2020-12-04 06:53:24-- https://sourceforge.net/projects/geoserver/files/GeoServer/2.18.1/geoserver-2.18.1-war.zip/
Reusing existing connection to sourceForge.net:443.
HTTP request sent, awaiting response... 302 Found
Location: https://sourceforge.net/projects/geoserver/files/GeoServer/2.18.1/geoserver-2.18.1-war.zip/download [following]
--2020-12-04 06:53:26-- https://sourceforge.net/projects/geoserver/files/GeoServer/2.18.1/geoserver-2.18.1-war.zip/download
Reusing existing connection to sourceForge.net:443.
HTTP request sent, awaiting response... 302 Found
Location: https://downloads.sourceforge.net/project/geoserver/GeoServer/2.18.1/geoserver-2.18.1-war.zip?r=&ts=1607039608&use_mirror=udomain [following]
--2020-12-04 06:53:26-- https://downloads.sourceforge.net/project/geoserver/GeoServer/2.18.1/geoserver-2.18.1-war.zip?r=&ts=1607039608&use_mirror=udomain
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 216.105.38.13
Connecting to downloads.sourceforge.net (downloads.sourceforge.net)[216.105.38.13]:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://udomain.dl.sourceforge.net/project/geoserver/GeoServer/2.18.1/geoserver-2.18.1-war.zip [following]
--2020-12-04 06:53:28-- https://udomain.dl.sourceforge.net/project/geoserver/GeoServer/2.18.1/geoserver-2.18.1-war.zip
Resolving udomain.dl.sourceforge.net (udomain.dl.sourceforge.net)... 203.135.147.10
Connecting to udomain.dl.sourceforge.net (udomain.dl.sourceforge.net)[203.135.147.10]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9950746 (95M) [application/octet-stream]
Saving to: "geoserver-2.18.1-war.zip"

geoserver-2.18.1-war.zip          20%[=====] 19.32M 2.14MB/s   eta 33s
] 19.32M 2.14MB/s   eta 33s
```

perintah tersebut akan mengunduh file zip dengan nama: geoserver-2.18.1-war.zip (sesuaikan nama berkas jika menggunakan versi yang berbeda)

- Lakukan ekstraksi file zip hasil unduhan pada folder aktif Tomcat. Pada Ubuntu, Tomcat menggunakan folder /opt/tomcat/latest/webapps (sebagaimana pengaturan softlink yang telah dibuat pada saat instalasi) untuk mempublikasikan aplikasi WAR. Gunakan perintah berikut untuk mengekstrak zip ke dalam folder tersebut:

Install unzip terlebih dahulu: sudo apt install unzip, kemudian:

```
sudo unzip -o geoserver-2.18.1-war.zip -d "/opt/tomcat/latest/webapps/"
```

- Jika belum, aktifkan Tomcat menggunakan perintah:

```
sudo systemctl start tomcat
```

Tunggu sampai beberapa saat pada saat Tomcat melakukan ekstraksi file WAR tersebut.

- Buka browser pada Windows di alamat: localhost:8080/geoserver

Welcome

Welcome

This GeoServer belongs to [The Ancient Geographers](#).

This GeoServer instance is running version **2.18.1**. For more information please contact the [administrator](#).

**Service Capabilities**

- WCS**
  - 2.0.1
  - 1.1.0
  - 1.1.1
  - 1.1
  - 1.0.0
- WFS**
  - 1.0.0
  - 1.1.0
  - 2.0.0
- WMS**
  - 1.1.1
  - 1.3.0
- TMS**
  - 1.0.0
- WMS-C**
  - 1.1.1
- WMPS**
  - 1.0.0

Cara yang sama dapat digunakan untuk aplikasi berbasis Java Web Archive lain, seperti [Mapstore](#) atau [Geonetwork](#). Instalasi melalui WAR ini akan memberikan kita kontrol penuh pada aplikasi, seperti pengaturan SSL, CORS, dan lain sebagainya. Demikian pula, deployment pada Tomcat memungkinkan beberapa aplikasi sekaligus untuk dapat diinstall pada port yang sama (Port 8080), sehingga cara inilah yang akan digunakan dalam latihan ini.

Selanjutnya, Geoserver siap digunakan untuk mengunggah data. Berikut adalah antarmuka Geoserver dengan menggunakan username: `admin` dan password `geoserver`.

Welcome

Welcome

This GeoServer belongs to [The Ancient Geographers](#).

**Service Capabilities**

- WCS**
  - 2.0.1
  - 1.1.0
  - 1.1.1
  - 1.1
  - 1.0.0
- WFS**
  - 1.0.0
  - 1.1.0
  - 2.0.0
- WMS**
  - 1.1.1
  - 1.3.0
- TMS**
  - 1.0.0
- WMS-C**
  - 1.1.1
- WMPS**
  - 1.0.0

Pada bagian kanan adalah daftar layanan yang didukung oleh Geoserver. Semua data yang diunggah pada Geoserver akan menggunakan layanan tersebut agar dapat diakses dari berbagai client, seperti QGIS. Inilah pembahasan kita berikutnya: bagaimana melakukan publikasi data agar dapat diakses secara *interoperable* oleh client?

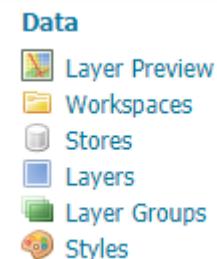
## 3.2 Publikasi Data Spasial dengan Geoserver

Web Map Server merupakan program komputer yang menghasilkan peta data yang direferensikan secara spasial dari data dan informasi geografis. Aplikasi ini, sebagaimana dibahas sebelumnya, adalah antarmuka sederhana yang menyajikan data dalam format standar untuk ditampilkan pada aplikasi berbasis web. Berbagai protokol melibatkan sintaks kueri untuk memposting meminta layer yang diinginkan dan memperbesar jendela ke server, yang menghasilkan peta sebagai image dalam format standar (GIF, PNG atau format lain) atau sebagai format vektor (misalnya KML, GML). Data ini kemudian dapat dikonsumsi oleh berbagai client. Pada bagian ini akan dibahas mengenai bagaimana melakukan publikasi data spasial dengan Geoserver.

### 3.2.1 Menghubungkan Geoserver dengan Data

Geoserver mengenali berbagai jenis data spasial. Adapun urutan untuk melakukan publikasi data pada Geoserver adalah sebagai berikut:

1. Buat **Workspace** baru (workspace ≈ nama project)
2. Buat **Store** baru sesuai jenis data (vector/raster/cascading). Pilih workspace yang telah dibuat
3. **Publish layer** dengan mengatur extent, proyeksi dan styling
4. Atur **Style** apabila diperlukan
5. Uji Layanan Layer dengan menggunakan **Layer Preview**



Langkah tersebut dapat diikuti sesuai urutan pada antarmuka Geoserver berikut:

Berikut adalah daftar datastore yang didukung oleh Geoserver pada instalasi defaultnya:

### New data source

Choose the type of data source you wish to configure

#### Vector Data Sources

- [Directory of spatial files \(shapefiles\)](#) - Takes a directory of shapefiles and exposes it as a data store
- [GeoPackage](#) - GeoPackage
- [PostGIS](#) - PostGIS Database
- [PostGIS \(JNDI\)](#) - PostGIS Database (JNDI)
- [Properties](#) - Allows access to Java Property files containing Feature information
- [Shapefile](#) - ESRI(tm) Shapefiles (\*.shp)
- [Web Feature Server \(NG\)](#) - Provides access to the Features published a Web Feature Service, and the ability to perform transactions on the server (when supported / allowed).

#### Raster Data Sources

- [ArcGrid](#) - ARC/INFO GRID Coverage Format
- [GeoPackage \(mosaic\)](#) - GeoPackage mosaic plugin
- [GeoTIFF](#) - Tagged Image File Format with Geographic information
- [ImageMosaic](#) - Image mosaicking plugin
- [WorldImage](#) - A raster file accompanied by a spatial data file

#### Other Data Sources

- [WMS](#) - Cascades a remote Web Map Service
- [WMTS](#) - Cascades a remote Web Map Tile Service

Dengan menggunakan ekstensi, daftar tersebut dapat berkembang menjadi lebih banyak. Silahkan mengacu pada halaman ekstensi Geoserver maupun [modul komunitas](#) untuk melihat opsi yang tersedia.

#### Latihan: Mengunggah Shapefile

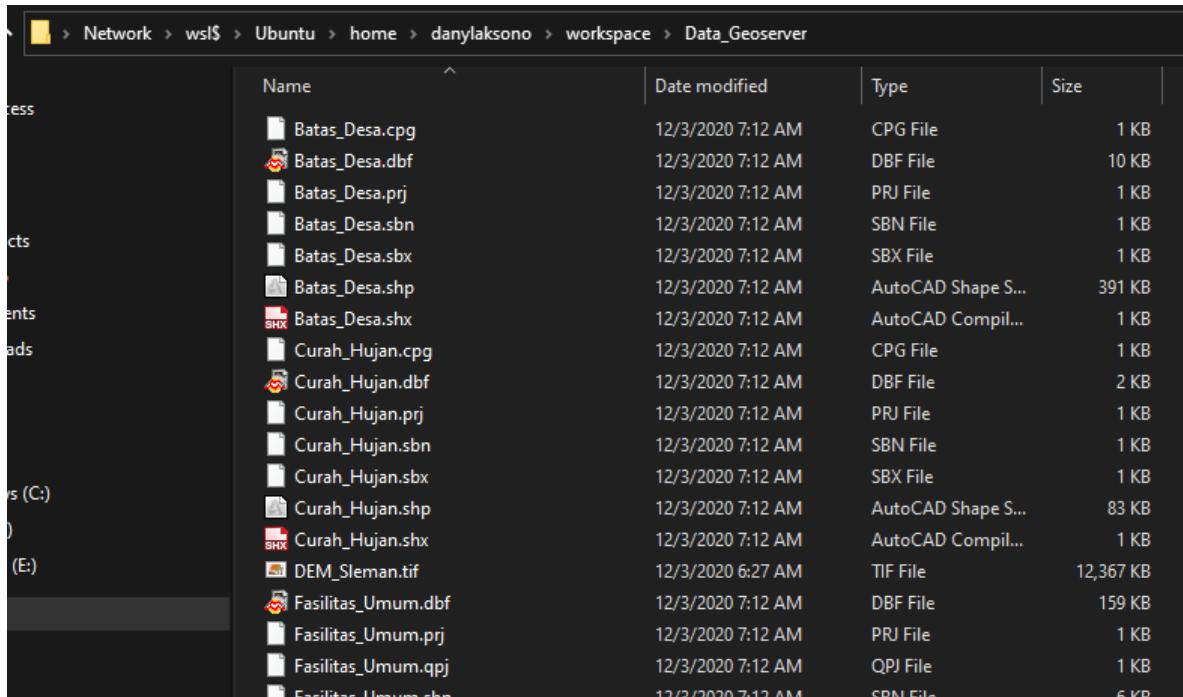
Geoserver membaca data yang disimpan pada Environment variable `GEOSERVER_DATA_DIR`, sehingga untuk melakukan ini, kita perlu mengcopy shapefile yang akan diunggah ke dalam folder tersebut. Lokasi folder ini dapat dilihat pada bagian data directory di server status:

The screenshot shows the GeoServer interface under the 'Server Status' tab. The left sidebar includes links for 'About & Status', 'Data', 'Services', and 'Settings'. The main content area displays various server statistics in a table format. The table has columns for 'Action' (which is 'Free locks' or 'Free memory' for some rows), 'Data directory' (set to '/opt/tomcat/apache-tomcat-9.0.40/webapps/geoserver/data'), 'Locks' (0), 'Connections' (4), 'Memory Usage' (157 MB / 910 MB), 'JVM Version' (Ubuntu: 11.0.9.1), 'Java Rendering Engine' (sun.java2d.marlin.DMarlinRenderingEngine), 'Available Fonts' (GeoServer can access 42 different fonts), 'Native JAI' (false), 'Native JAI ImageIO' (false), 'JAI Maximum Memory' (455 MB), and 'JAI Memory Usage' (0 KB).

Pada gambar di atas, folder Data Directory adalah /opt/tomcat/apache-tomcat-9.0.40/webapps/geoserver/data.

Masih dengan data latihan sebelumnya, kita akan gunakan layer-layer tersebut untuk mengunggah data pada Geoserver.

1. Dengan menggunakan koneksi WSL ke Windows Explorer, copykan data yang akan kita unggah ke folder di dalam WSL, misalnya `home` directory kita. Ingat, wsl dapat dibuka dari Windows Explorer menggunakan `\wsl$`.



2. Dari WSL, copykan data tersebut ke folder `GEO SERVER _DATA_DIR` yang kita catat sebelumnya. Gunakan perintah `cp` untuk mengcopy berkas ke folder yang dimaksud.

```
sudo cp -a Data_Geoserver/ "/opt/tomcat/apache-tomcat-9.0.40/webapps/geoserver/
˓→data/shpsleman/"
```

Pada Geoserver, kita dapat memanggil layer tersebut pada saat publikasi layer. Pembahasan mengenai publikasi layer adalah sebagai berikut.

### Latihan: Melakukan Publikasi dari Shapefile

Setelah kita memiliki shapefile pada server, berkas tersebut perlu dipublikasi agar dapat diakses menggunakan Standar OGC. Berikut langkah untuk melakukan publikasi data pada Geoserver:

1. Membuat Workspace. Workspace pada Geoserver dapat dianggap sebagai sebuah project yang didalamnya berisi beberapa layer sekaligus. Untuk membuat workspace, gunakan menu **Workspace** pada bagian kiri dari halaman Geoserver, kemudian Klik **Add New Workspace**.

### New Workspace

Configure a new workspace

**Basic Info**    **Security**

Name

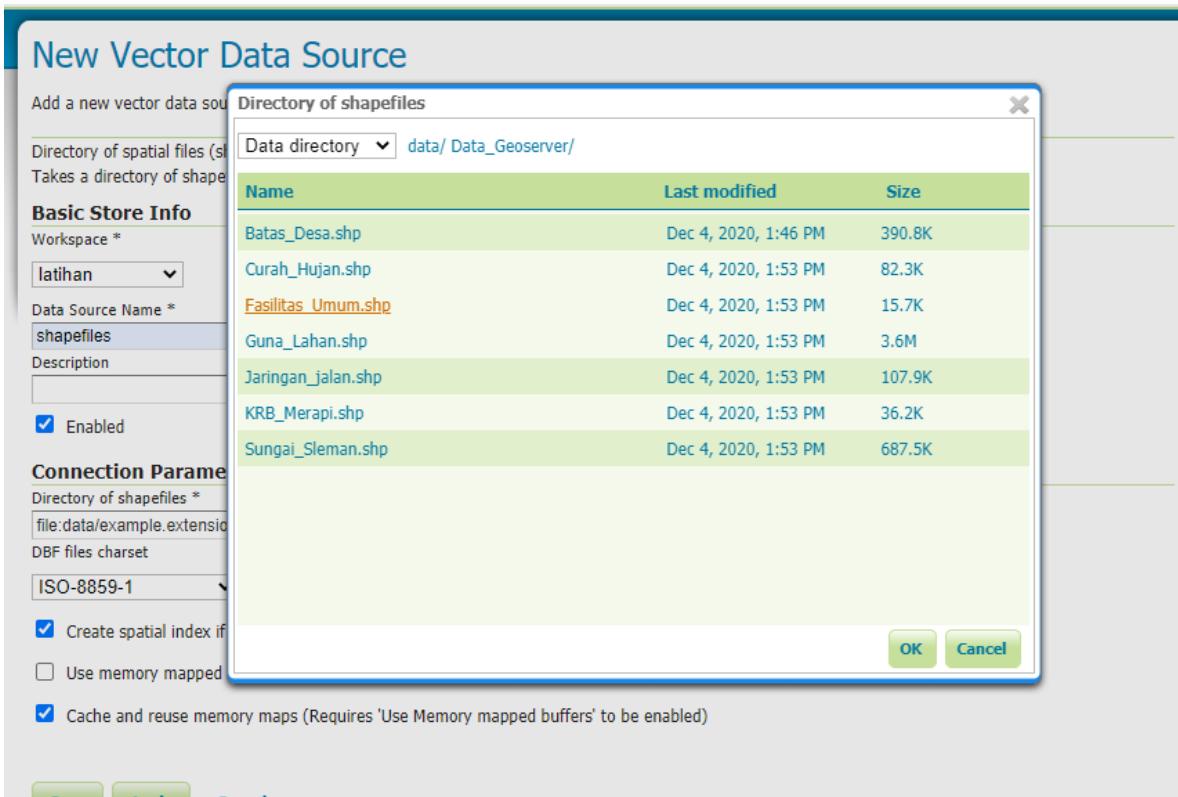
Namespace URI  
  
The namespace uri associated with this workspace

Default Workspace  
 Isolated Workspace

**Save**    **Cancel**

Berikan nama dan URI pada workspace tersebut. URI digunakan sebagai identitas pada saat layer yang tersimpan di dalam workspace tersebut dipanggil nantinya. Klik **Save** untuk menyimpan workspace

2. Berpindah ke langkah selanjutnya, kita akan menghubungkan Geoserver dengan data pada direktori. Gunakan menu **Stores** untuk menyimpan dan menghubungkan satu atau beberapa layer dengan menggunakan Geoserver. Klik pada **Add New Store** untuk menghubungkan sumber data baru. Karena kita akan menambahkan beberapa **Shapefile** sekaligus, gunakan pilihan **Directory of Spatial Files** (shapefiles) seperti berikut:



klik OK untuk konfirmasi penambahan Store

3. Geoserver akan secara otomatis mengarahkan kita ke halaman layer. Selanjutnya adalah menggunakan menu 'Publish' pada halaman tersebut untuk mempublikasikan data

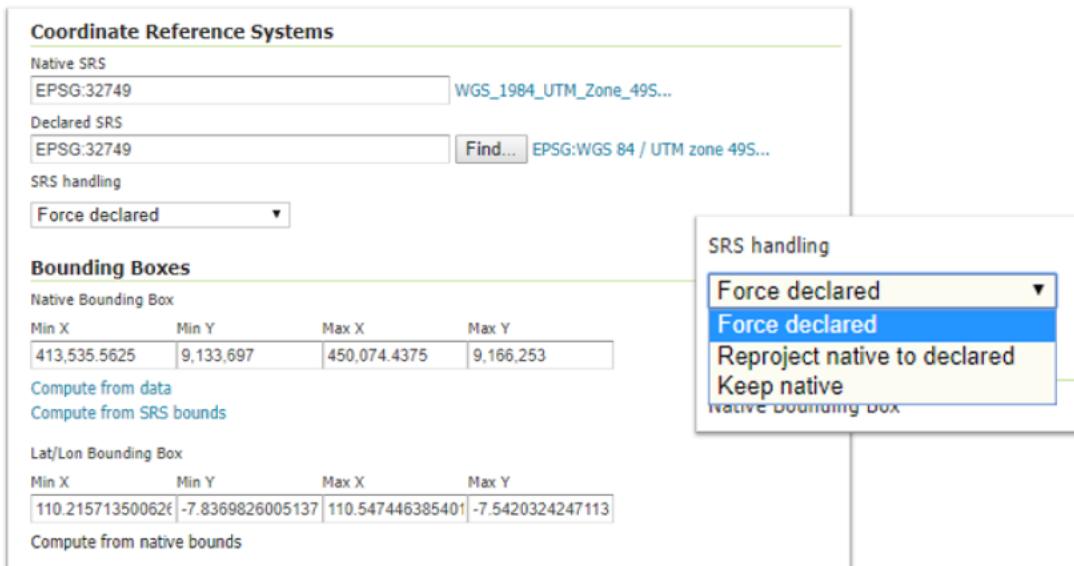
## New Layer

Add a new layer

You can create a new feature type by manually configuring the attribute names and types. [Create new feature type...](#)  
Here is a list of resources contained in the store 'shapefiles'. Click on the layer you wish to configure

<<	<	1	>	>>	Results 1 to 7 (out of 7 items)	Search
						Action
					Batas_Desa	Publish
					Curah_Hujan	Publish
					Fasilitas_Umum	Publish
					Guna_Lahan	Publish
					Jaringan_jalan	Publish
					KRB_Merapi	Publish
					Sungai_Sleman	Publish

4. Pada saat publikasi, hal yang harus diperhatikan adalah pengaturan **sistem proyeksi** dan **extent**. Pengaturan ini dapat dijumpai pada bagian akhir di halaman publikasi:



Terdapat tiga pengaturan penggunaan sistem proyeksi yang ada pada Geoserver:

- **Force Declared**, abaikan sistem proyeksi asli dari layer, dan berikan definisi sistem proyeksi baru pada layer tersebut. Ini dapat digunakan apabila sistem proyeksi asli dari layer hilang atau rusak, misalnya pada berkas dengan tipe *shapefile* (prj). Tool ini sama seperti *Define Projection* pada perangkat desktop
- **Reproject Native to Declared**, digunakan untuk melakukan transformasi sistem proyeksi, misalnya dari sistem proyeksi UTM-49S (EPSG:32749) menjadi sistem proyeksi geografis (EPSG:4326). Ini sama seperti tool *Reproject* pada perangkat lunak GIS desktop
- **Keep Native**, abaikan sistem proyeksi yang didefinisikan oleh Geoserver, dan gunakan sistem proyeksi asli yang terdapat pada data. Ini dapat dipilih, sebagai contoh, pada saat melakukan publikasi peta dengan sistem proyeksi lokal

5. Untuk menguji apakah layer sudah sesuai dengan keinginan, gunakan menu Layer Preview pada Geoserver

**Layer Preview**

List of all layers configured in GeoServer and provides previews in various formats for each.

Type	Title	Name	Common Formats	All Formats
	Batas Administrasi Desa DIY	geonode:admin_diy_pemendagri	OpenLayers GML KML	Select one Select one <b>WMS</b> AtomPub GIF GeoRSS <b>GeoTiff</b> GeoTiff 8-bits JPEG JPEG-PNG JPEG-PNG8 KML (compressed) KML (network link) KML (plain) OpenLayers OpenLayers 2 OpenLayers 3 PDF PNG PNG 8bit SVG
	Pola Ruang Kabupaten Bantul	geonode:pola_ruang_bantul	OpenLayers GML KML	
	Pola Ruang Kabupaten Gunungkidul	geonode:pola_ruang_gk	OpenLayers GML KML	
	Pola Ruang Kabupaten Kulon Progo	geonode:pola_ruang_kp	OpenLayers GML KML	
	Pola Ruang RDTR Kota Yogyakarta	geonode:pola_ruang_rdtr_kota_jogja	OpenLayers GML KML	

<< < 1 > >> Results 1 to 5 (out of 5 items)

Search

Pada bagian ini, dapat diperiksa apakah pemanggilan WMS dan WFS untuk layer yang dipublikasi sudah sesuai. Demikian pula, dapat diperiksa mengenai kesesuaian bounding box dan sistem proyeksi yang digunakan.

Setelah publikasi, pengaturan yang diperlukan adalah pembuatan style serta pengaturan keamanan apabila diperlukan. Bagian akhir dari bab ini membahas mengenai pengaturan style pada layer yang dapat digunakan untuk WMS pada Geoserver.

### Latihan: Menghubungkan Geoserver dan Basisdata PostGIS

PostGIS dapat dihubungkan melalui antarmuka Geoserver. Berikut adalah tampilan antarmuka dan parameter koneksinya:

## New Vector Data Source

Add a new vector data source

PostGIS  
PostGIS Database

### Basic Store Info

Workspace \*

cite ▾

Data Source Name \*

Description

Enabled

### Connection Parameters

host \*

localhost

port \*

5432

database

schema

public

user \*

passwd

\*\*\*\*\*

Namespace \*

<http://www.opengeospatial.net/cite>

Sama seperti pada QGIS, koneksi ke PostGIS dari Geoserver juga menggunakan parameter seperti Host, Username, Basisdata dan Password.

Setelah terkoneksi sebagai store, layer pada store tersebut juga dapat ditambahkan pada Geoserver melalui Publikasi sebagaimana pada layer yang dipublikasi menggunakan Shapefile. Penggunaan PostGIS dalam hal ini memiliki beberapa keuntungan, diantaranya adalah:

- Keamanan data yang lebih baik, karena akses pada data dibatasi oleh Geoserver dan PostGIS sekaligus
- Manajemen data yang fleksibel. Dengan menggunakan PostGIS, akses terhadap data dapat diatur menggunakan aplikasi framework di luar Geoserver, seperti menggunakan Laravel atau NodeJS.
- [SQL View](#), merupakan cara untuk memanggil layer menggunakan parameter. Sebagai contoh, kita dapat memanggil hanya layer titik yang dipilih oleh pengguna melalui bounding box, misalnya.

### Latihan: Publikasi Data melalui REST-API Geoserver

Pada remote server, seringkali kita tidak dapat menggunakan antarmuka yang memungkinkan untuk publikasi data secara langsung. Boleh jadi kita tidak dapat menggunakan console pada server untuk mengcopy data pada folder Data Directory seperti yang kita lakukan sebelumnya. Dengan demikian, interaksi pada Geoserver juga menjadi terbatas. Untuk itu, Geoserver menyediakan REST API yang dapat digunakan untuk berinteraksi dengan fungsi-fungsi Geoserver melalui antar muka API standard. API ini dapat diakses melalui <http://alamatgeoserver:8080/geoserver/rest/workspaces>.

Berikut adalah langkah untuk mengunggah file pada server:

1. Membuat Workspace baru dengan nama latihan (lewati jika Workspace sudah tersedia)

```
curl -v -u admin:password -XPOST -H "Content-type: text/xml" -d "<workspace>
  <name>latihan</name></workspace>" http://alamatgeoserver:8080/geoserver/
  **rest**/workspaces
```

2. Mengunggah file `jalan.zip` sebagai layer baru pada workspace `latihan` dengan nama datastore `jalan`

```
curl -v -u admin:geoserver -XPUT -H "Content-type: application/zip" --data-binary
  @jalan.zip http://54.254.9.31:8080/geoserver/rest/workspaces/latihan/datastores/
  jalan/file.shp
```

Dengan cara seperti ini, kita dapat mengunggah layer dengan format shapefile sebagai layer pada Geoserver meskipun kita tidak memiliki akses pada server tersebut secara langsung.

### 3.2.2 Styling Publikasi dengan SLD

Protokol WMS yang digunakan pada Geoserver memerlukan styling sebelum disajikan pada client. Format yang digunakan oleh Geoserver adalah SLD (*Style Layer Descriptor*). Antarmuka untuk pengaturan SLD adalah sebagai berikut:

**New style**

Type a new style definition, or use an existing one as a template, or upload a ready made style from your file system. The editor can provide syntax highlighting and automatic formatting. Click on the "validate" button to verify the style is a valid style document.

**Data**

**Style Data**

Name:

Workspace:

Format:

**Style Content**

Generate a default style:

Copy from existing style:

Upload a style file:  No file chosen

**Legend**

Legend:

Preview legend:

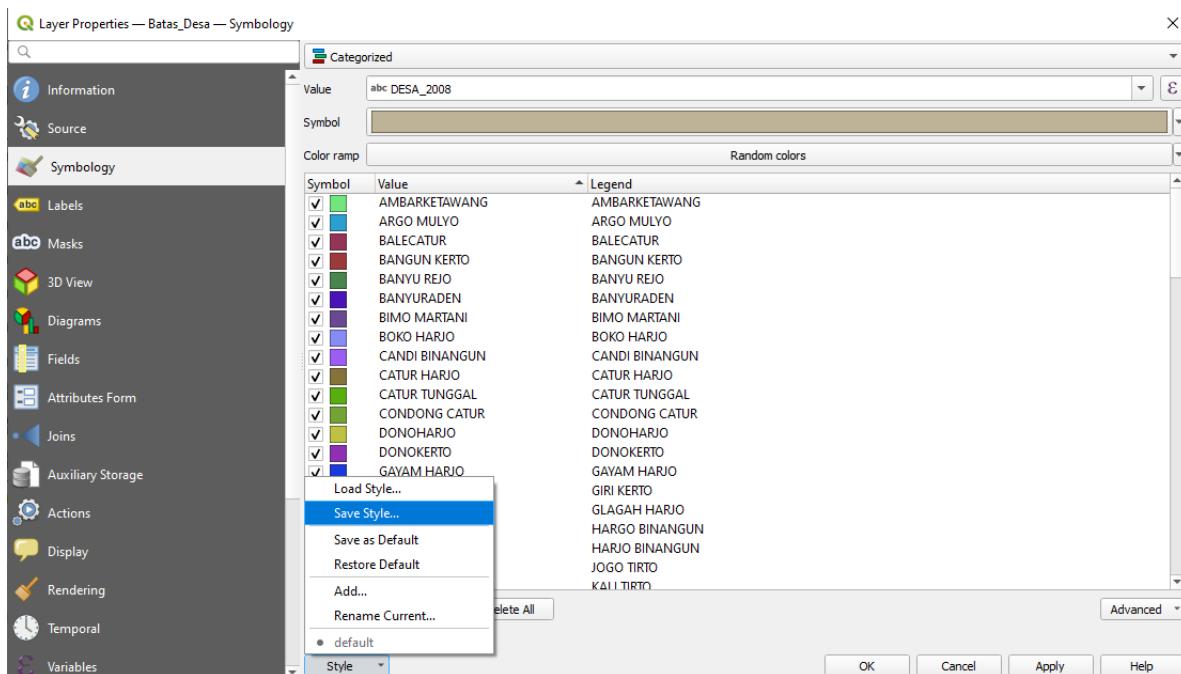
**Formatting Tools**

Font:  Height:

**Action Buttons**

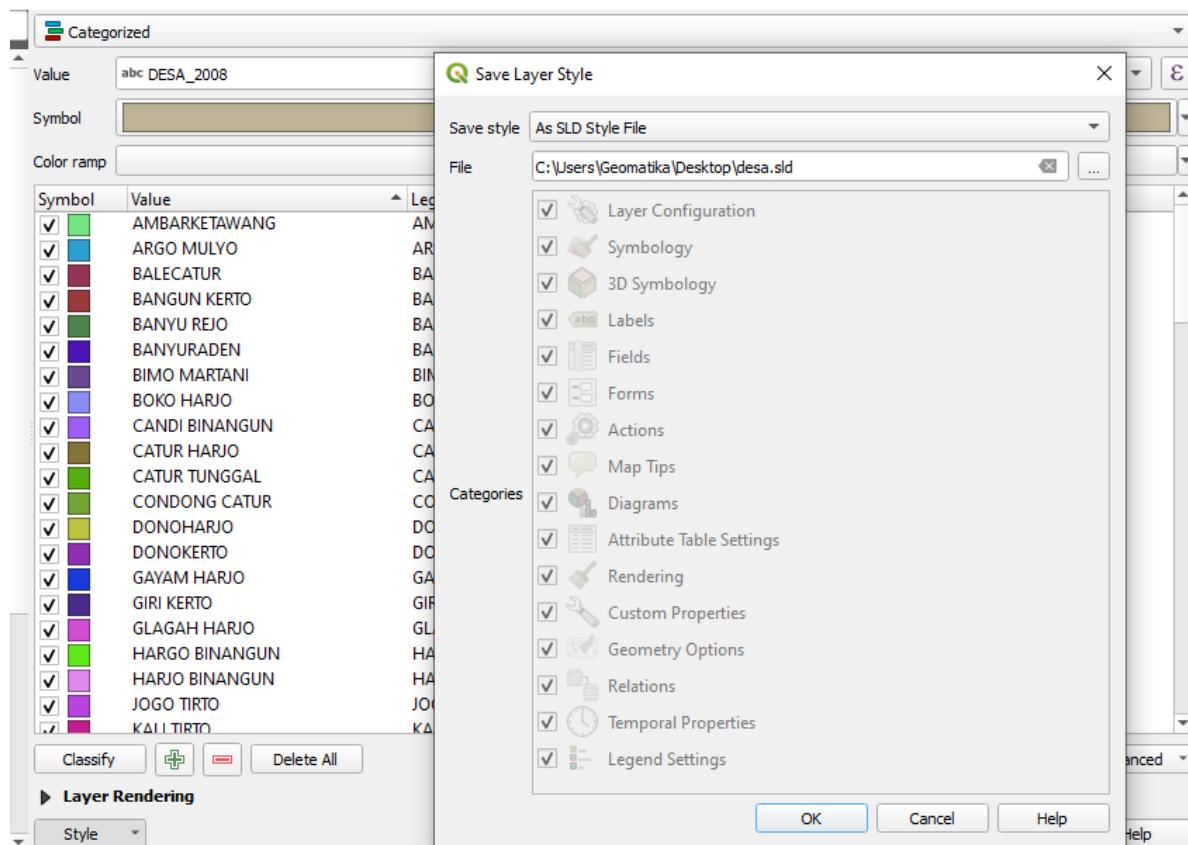
SLD menggunakan format XML untuk menyimpan style. Untuk memudahkan styling, dapat digunakan QGIS untuk menyimpan simbolosasi yang diinginkan. Berikut adalah menu untuk melakukan *export Style* yang dapat dilakukan pada QGIS:

1. Pada Properti layer di QGIS, buka simbologi, kemudian buat simbol sesuai dengan style yang diinginkan



Klik pada Style > Save Style untuk membuka kotak dialog penyimpanan style tersebut.

- Simpan style tersebut sebagai SLD. File inilah yang nanti akan diunggah pada antarmuka SLD di Geoserver seperti di atas



Hasil publikasi dapat dikonsumsi oleh perangkat client seperti QGIS atau LeafletJS melalui alamat publikasi masing-masing layer sesuai standar OGC. Pembahasan ini akan diberikan pada bab selanjutnya.

### 3.3 Konsumsi Data dengan Standar OGC

Dengan publikasi pada Geoserver, layer yang tersimpan tersebut akan dapat dipanggil dari berbagai macam client. Seperti tujuan awal penggunaan server data spasial, terdapat beragam client yang mendukung standar internasional OGC, sehingga dengan menggunakan standar tersebut, berbagai layanan dapat dikombinasikan dengan memperhatikan prinsip-prinsip FAIR. Pada bagian ini akan dibahas mengenai bagaimana menggunakan perangkat FOSS untuk mengakses layanan OGC yang disajikan dari Geoserver.

### 3.3.1 Menggunakan Perangkat FOSS untuk mengakses data OGC

Berbagai perangkat FOSS dapat saling berkomunikasi dengan bantuan dukungan standar OGC. Untuk latihan ini, QGIS akan digunakan sebagai perangkat client untuk memanggil data dari Geoserver. Anda juga dipersilahkan untuk mencoba perangkat lunak lain, seperti uDIG atau gvSIG yang memiliki kemampuan serupa.

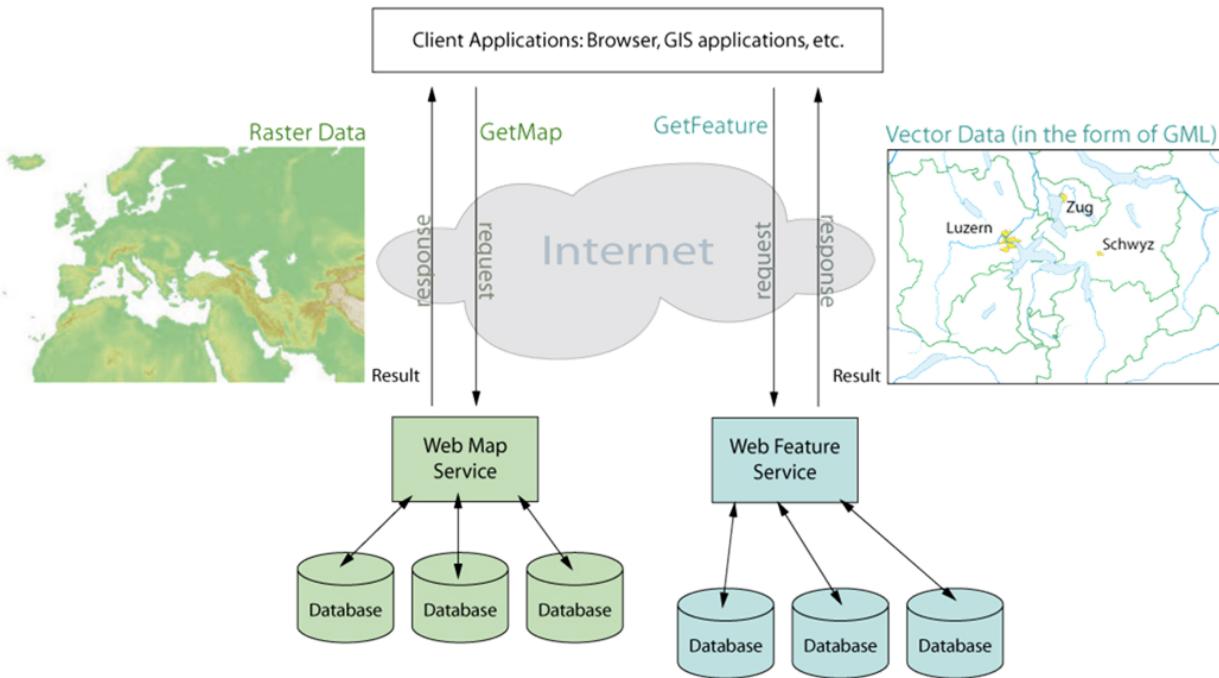


Fig. 3.8: Konsumsi WMS dan WFS dari Geoserver

Standar-stanadar OGC menggunakan rangkaian parameter pada URL yang digunakan untuk memanggil berbagai layanan sesuai fungsinya, seperti WMS, WFS maupun WCS. Dari parameter inilah aplikasi client dapat digunakan untuk memanggil layanan yang dibutuhkannya. Perangkat lunak seperti QGIS dapat memanggil beragam layanan yang menggunakan standar OGC ini.

#### Memanggil WMS, WFS dan WCS dengan QGIS

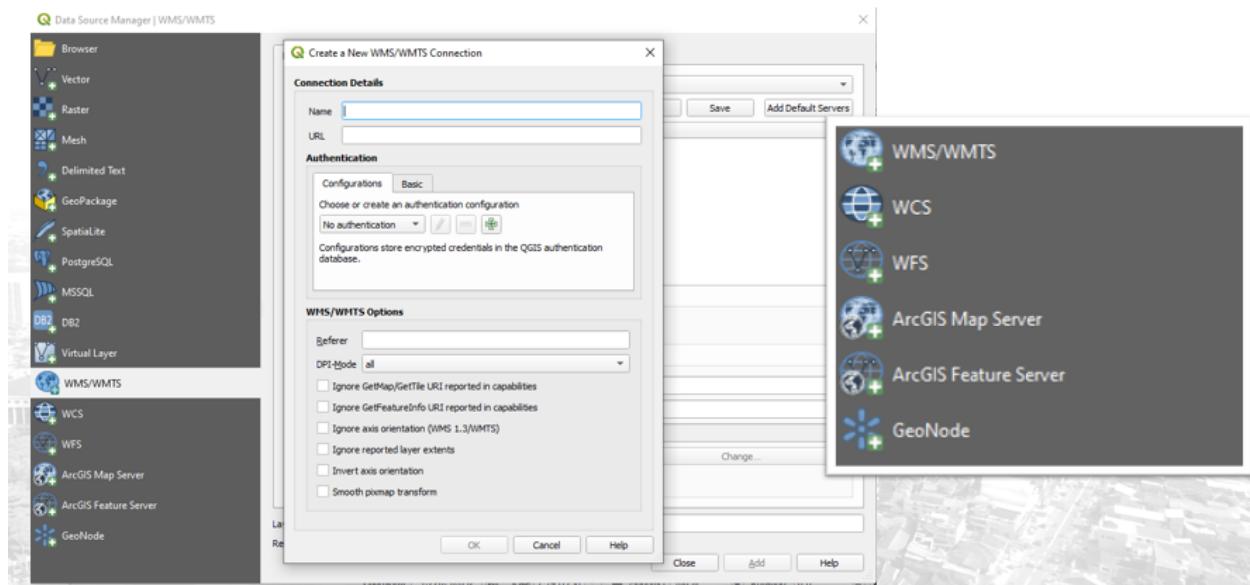
Apabila kita perhatikan, link publikasi WMS dari Geoserver adalah seperti berikut. Link ini dapat dipanggil dari Layer Preview, kemudian memilih WMS atau WFS:

```
**http://localhost:8080/geoserver/Sleman/wms?**
service=WMS
&version=1.1.0
&request=GetMap
&layers=Sleman%3ABatas_Des
&bbox=413535.5625%2C9133697.0%2C450074.4375%2C9166253.0
&width=768&height=684
&srs=EPSG%3A32749
&format=image%2Fpng
```

Sedangkan berikut adalah link publikasi untuk WFS:

```
**http://localhost:8080/geoserver/Sleman/ows?**
service=WFS
&version=1.0.0
&request=GetFeature
&typeName=Sleman%3ABatas_Desa
&maxFeatures=50
&outputFormat=application%2Fgml%2Bxml%3B%20version%3D3.2
```

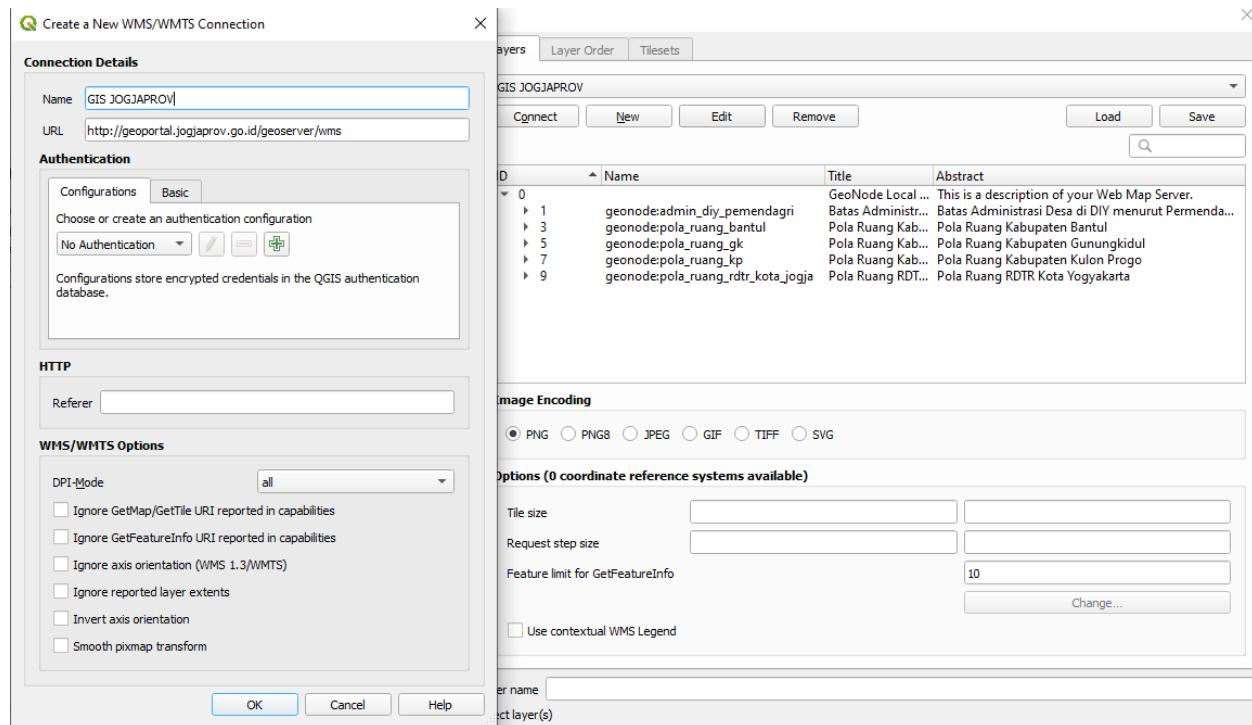
Baris-baris di atas menunjukkan parameter yang dapat diakses dari client, dan disebut sebagai key-value pair (KVP), dimana tiap kunci berisi parameter yang dapat diatur untuk memanggil WMS yang dimaksud (misalnya, format PNG atau JPG). Apabila kita menggunakan QGIS, parameter inilah yang akan dipanggil. QGIS telah menyediakan fungsi yang cukup lengkap untuk memanggil data dari Geoserver. Berikut caranya:



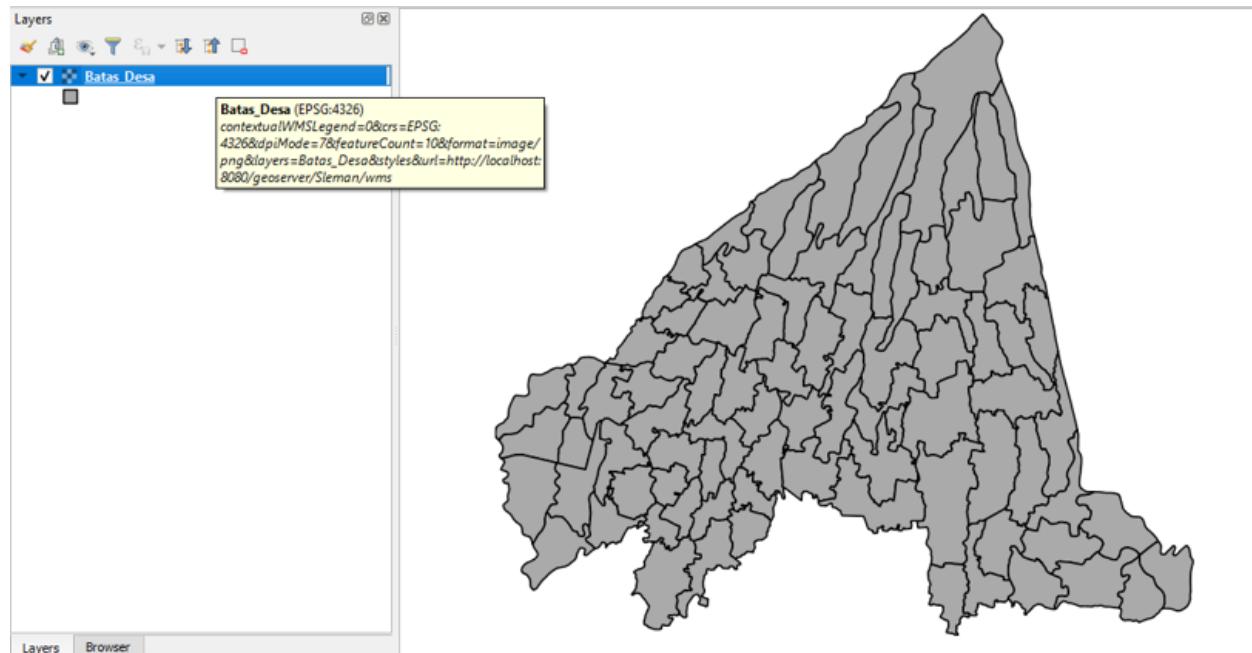
Tiap menu digunakan untuk memanggil layanan OGC dari jenis yang berbeda (WMS/WMTS, WFS, dan WCS). Parameter yang diperlukan untuk memanggil layer ini adalah *alamat host Geoserver*. Alamat ini adalah link bertanda tebal pada contoh KVP di atas. Misalnya untuk WMS:

```
http://localhost:8080/geoserver/Sleman/wms
```

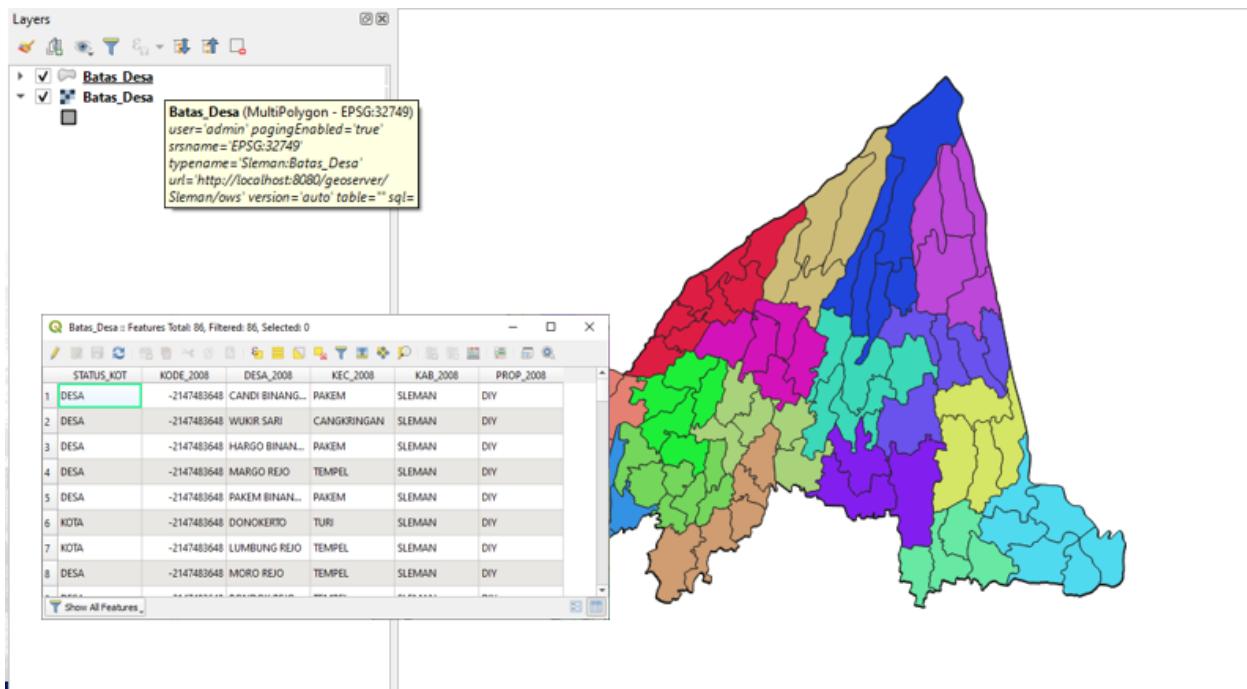
QGIS selanjutnya akan memanggil seluruh layer pada Geoserver tersebut melalui WMS GetCapabilities. Berikut adalah layer yang dipanggil dari Geoserver Geoportal Propinsi Jogja melalui WMS GetCapabilitiesnya:



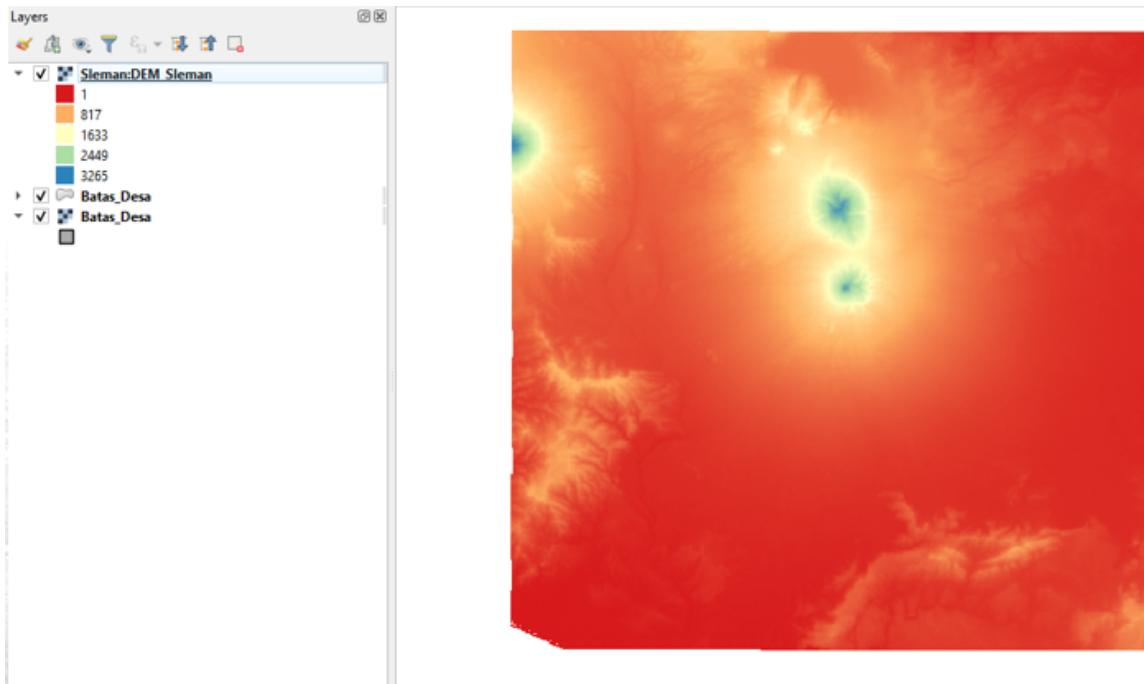
Apabila kita panggil layer tersebut pada QGIS, kita dapat melihat perbedaan antara WMS, WFS, dan WCS seperti berikut:



WMS dipanggil sebagai Image atau raster, sehingga kita tidak dapat melakukan simbologi pada layer tersebut. Data atribut pada WMS dapat diakses dengan menggunakan tool *Feature Info* pada QGIS, tetapi atribut ini hanya dapat ditampilkan, dan bukan untuk dianalisis.



WFS dipanggil sebagai vektor, sehingga kita dapat melakukan simbologi, analisis spasial, dan perlakuan lain yang dapat kita lakukan pada sebuah layer vektor biasa. Seluruh atribut dan Geometri dari layer ini dapat dipanggil.



Sama seperti WMS, WCS memanggil raster dari Geoserver. Perbedaannya terletak pada nilai raster yang diteruskan ke QGIS. Pada WCS, seluruh nilai raster sama seperti data aslinya, sehingga kita dapat menggunakan simbologi pada sisi client seperti gambar di atas.

### Memanggil data publikasi OGC dengan Javascript Library (LeafletJS)

Pada LeafletJS (dan library web map lain seperti OpenLayers, GMap API, Mapbox dst), layer secara umum terbagi menjadi dua macam, yaitu 1) Tilelayer ('Slippy Map') yang dapat diperoleh dalam bentuk layanan (service) dari penyedia tile, dan 2) Feature Layer yang dapat ditambahkan sendiri pada Leaflet. Tilelayer cenderung bersifat statis, tidak interaktif dan disajikan dalam bentuk 'tile'. Tilelayer sering digunakan sebagai background atau basemap untuk peta yang akan dibuat pada sebuah webmap library seperti Leaflet. Pada Tilelayer ini tidak dapat diterapkan konsep interaktivitas, misalnya memunculkan popup berdasarkan atribut (dengan beberapa pengecualian, misalnya pada Vector Tile).

Adapun Feature Layer merupakan layer yang dapat ditambahkan pada LeafletJS Map bersumber dari data yang kita miliki. Contoh Feature layer yang banyak dijumpai adalah Marker. Pada Marker kita dapat memberikan atribut dan menerapkan interaktivitas, misalnya dengan memunculkan popup, melakukan drag and drop, dan seterusnya. Selain marker, kita juga dapat menambahkan polygon dan polyline dengan cara yang hampir sama. Feature Layer yang lain adalah GeoJSON, yang memungkinkan kita untuk menambahkan layer data spasial lengkap dengan atribut dan fungsi interaktifnya. Pada bagian ini akan dibahas bagaimana menggunakan GeoJSON untuk menampilkan data spasial (shapefile) menjadi peta yang interaktif pada LeafletJS melalui Geoserver.

Untuk memanggil data dari Geoserver menggunakan LeafletJS, dapat digunakan script seperti berikut:

```
var layerjalan = L.tileLayer.wms('http://localhost:8080/geoserver/wms?', {
    layers: 'latihan:jalan'
}).addTo(map);
```

Contoh script LeafletJS tersedia pada <https://github.com/danylaksono/sample-leaflet.js>.

Sebagai latihan, modifikasi script tersebut untuk memanggil layer KRB yang sudah diunggah pada Geoserver WSL pada alamat <http://localhost:8080/geoserver> sebagai WMS dan WFS, kemudian bandingkan perbedaan antara keduanya.