Komputer utama

composer create project laravel/laravel olivia 🡪 di cmd

membuat repository baru di github dengan langakah”

cara menarik perubah route hello agar masuk ke dalam komputer kita juga, karena sebelumnya yang melakukan perubahan di komputer teman.

git pull origin main

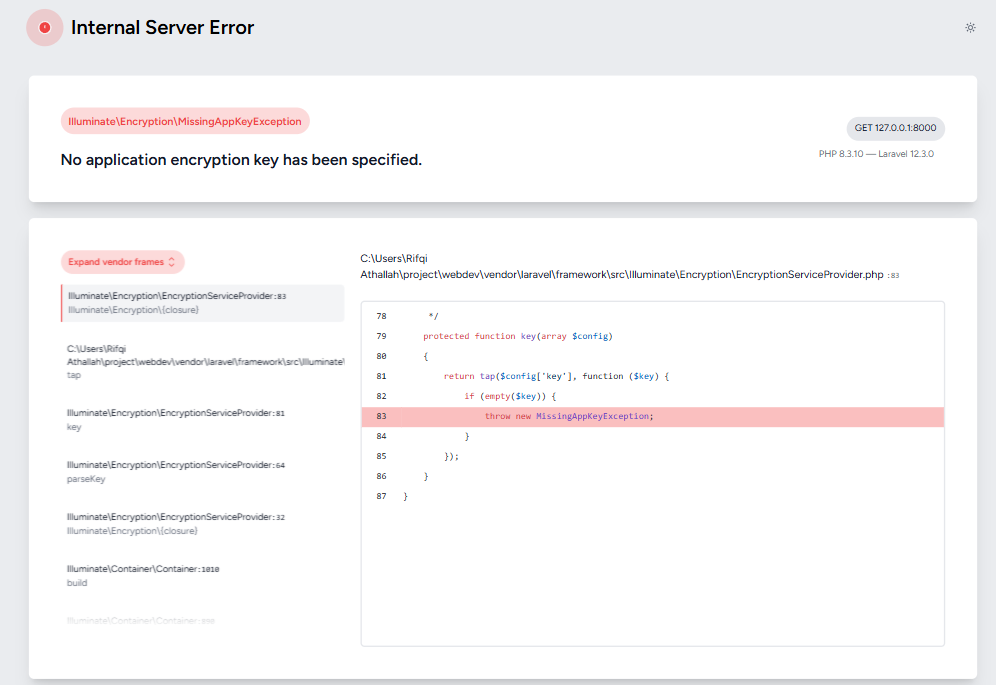


komputer temanmu

buat folder project dimana aja, setelah itu klik kanan buka git bash masuk ke dalam git bash, lalu masukan perintah berikut ini.

1. git clone https://github.com/rifkythalah/webdev.git
2. ls webdev/
3. code .
4. composer install (untuk install library vendor laravel)
5. copy isi .env.example
6. buat file baru .env dan copy semua isi dari .env.example
7. sesuikan database pada .env dan mysql nya.
8. Npm install
9. php artisan serve

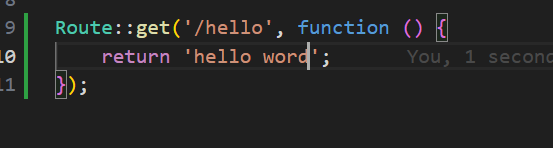
terdapat eror seperti ini



Kembali ke cmd dan tuliskan perintah berikut ini

php artisan key:generate

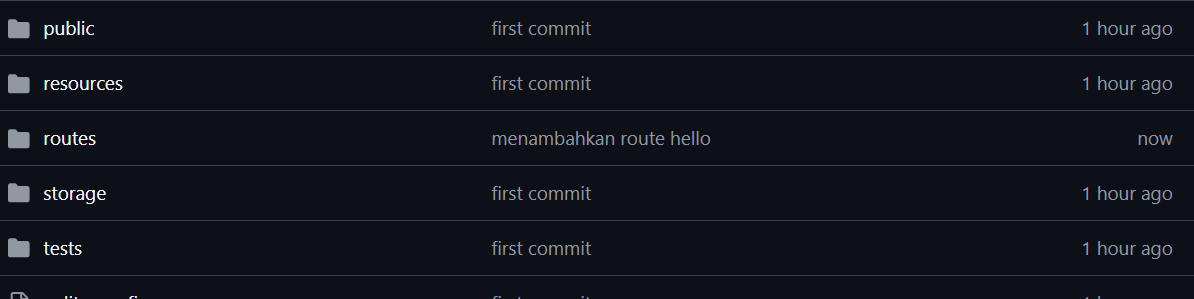
php artisan migrate

jika ingin mencoba melakukan perubahan, contoh 

git add .

git commit -m ’menambahkan route baru’

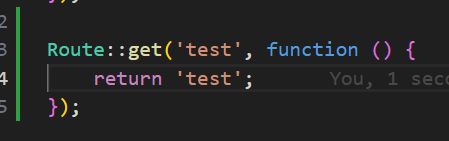
git push origin main



**Masalah conflict**

Komputer utama

Semisal jika komputer teman dan komputer mu saling mengerjakan ke duanya melakukan update an tanpa tahu temenmu lagi mengerjakan halaman yang sama, semisal contoh perubahan yang ada di komputermu.



git add .

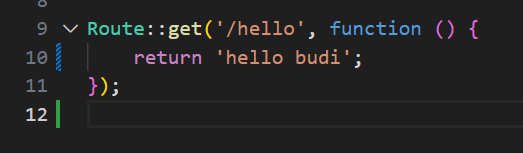
git commit -m ’menambahkan test’

git push origin main 🡪 disini berhasil melakukan perubahan

langkah selanjutnya

git pull origin main

Komputer temenmu



Dan jika disini komputer temenmu ingin melakukan perubahan juga tidak bisa

git pull origin main 🡪 tidak akan berhasil

git add .

git commit -m ‘hello’

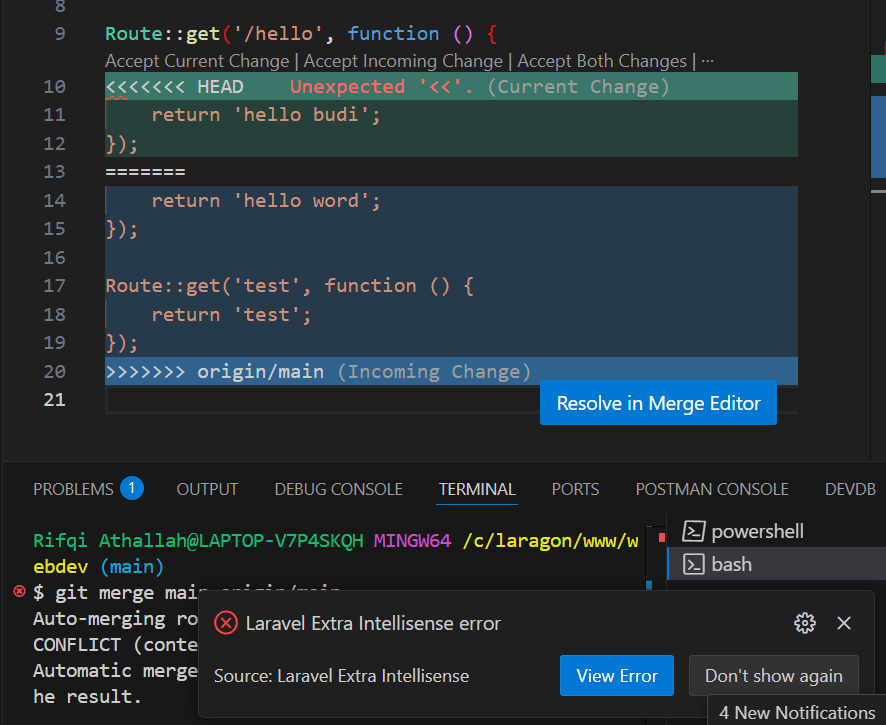
git push origin main 🡪 ini tidak akan berhasil karena akan conflict

langkah atasi nya tarik dulu repo sebelumnya.

git pull origin main 🡪 jika eror langkah dibawahnya

jika masih terjadinya conflict bisa melakukan perintah

* git fetch origin main
* git merge main origin/main



Klik accept cureent change, 🡪 git push origin main

Kalo tidak bisa langsung lakukan ini dahulu

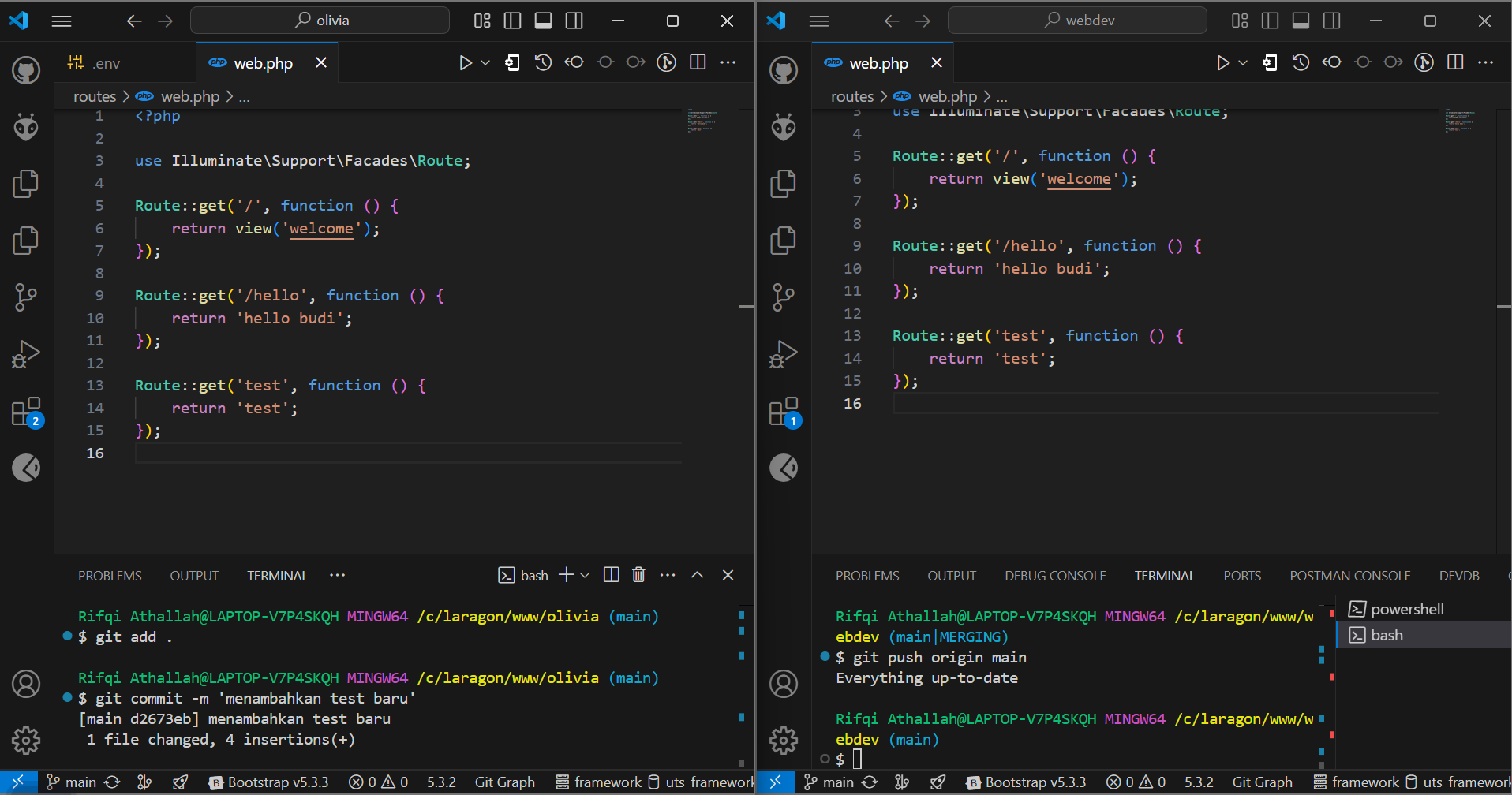
git add.

git commit -m ‘merubah’

git push origin main

accept cureent change = melakukan perubahan punya komputer kita

accept incoming change = melakukan perubahan punya komputer lain.



Hasil akhirnya seperti contoh gambar diatas.

**SARAN**

disini akan gagal karena ketika ingin mengpush ke repo, sebelumnya di komputer utama sedang melakukan push update terbaru.

Tarik dulu repo sebelumnya dengan (git pull origin main), lalu melakukan update an terbaru nya.

Solusi seperti ini bisa dihindari jika kamu dan temenmu saling komunikasi dan gantian untuk push commit perubahan di repository.

Dan sebelum melakukan push update terbaru cek dulu di repo update an apa yang terbaru, jika ada kalian bisa melakukan (git pull origin main)

APP\_URL=https://10.41.99.226

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=juaraolivia\_db

DB\_USERNAME=laraveluser

DB\_PASSWORD=Pa$$w0rdLaravel!

http://10.41.99.226/

ssh ninja5@10.41.99.226

mysql -u laraveluser -p

USE juaraolivia\_db;

show databases;

SHOW TABLES;

DESCRIBE pemerintah\_pusat\_dinas;

**Buat Melihat kesalahan di git bash atau terminal**

tail -f storage/logs/laravel.log

**Password Vm 5**

[Ninja5.123@#](mailto:Ninja5.123@)

Var/www/juaraolivia

.env

MAIL\_MAILER=smtp

MAIL\_HOST=smtp.gmail.com

MAIL\_PORT=587

MAIL\_USERNAME=femasalfaridzi17@gmail.com

MAIL\_PASSWORD=mqjpsglffhtftcxw

MAIL\_ENCRYPTION=tls

MAIL\_FROM\_ADDRESS=femasalfaridzi17@gmail.com

MAIL\_FROM\_NAME="LAPOR PAL"

Password akun user

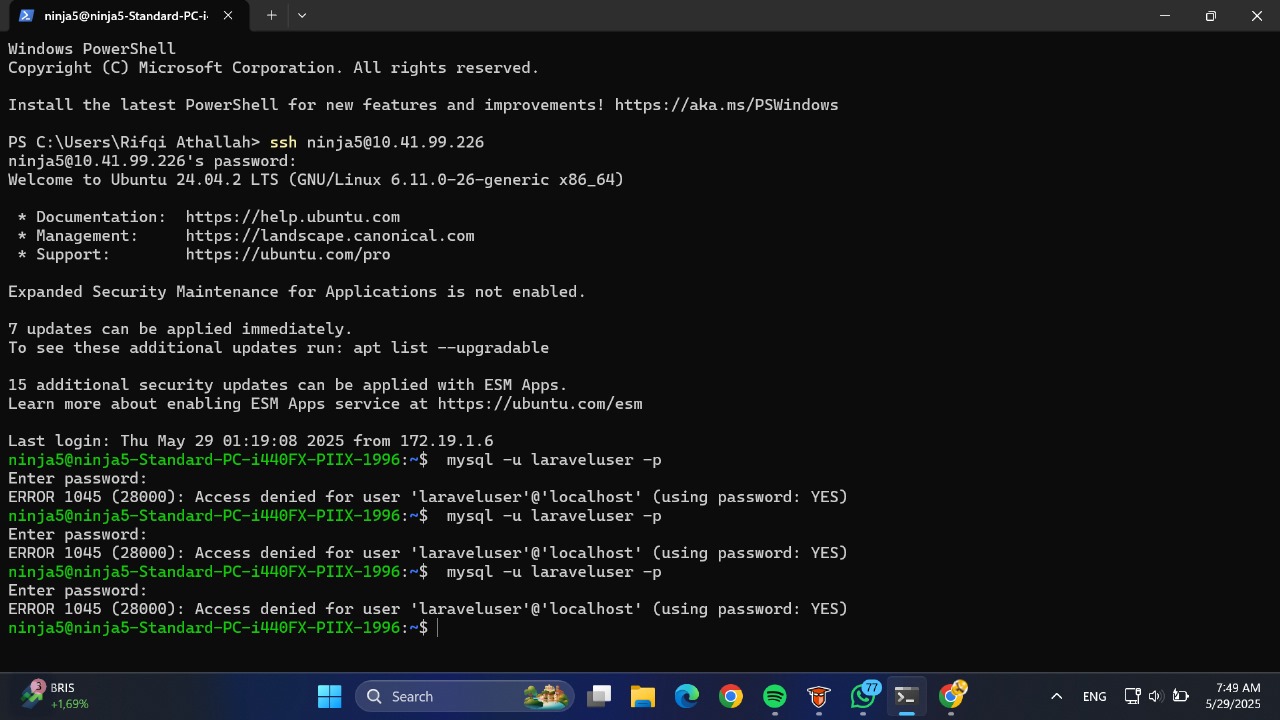
dinas = Pusat.123

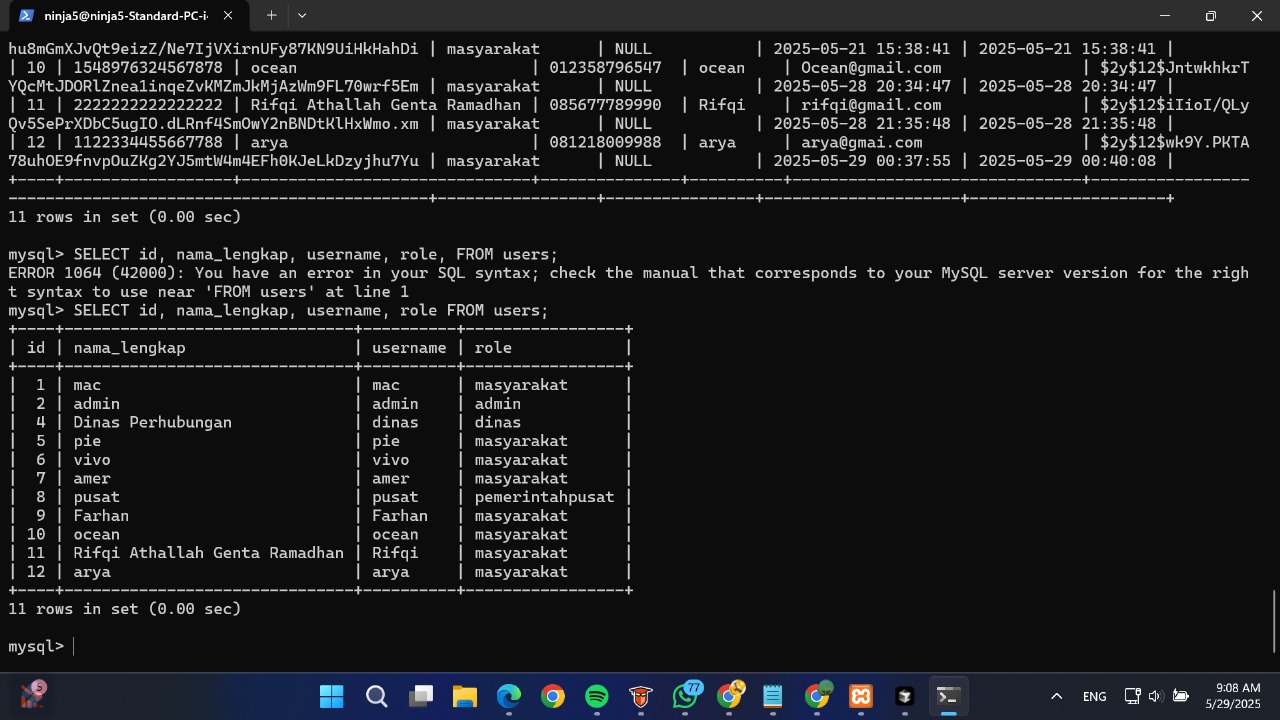
admin = Admin.123

pusat = Pusat.123

-7.977968742058564, 112.63387872431868 = pusat

-7.983458346506709, 112.63259214150533 = dinas





php artisan schedule:work

php artisan schedule:run

php artisan app:escalate-laporan-menunggu

php artisan app:escalate-laporan-tidak-terselesaikan

http://127.0.0.1:8000/escalate-laporan

<http://127.0.0.1:8000/escalate-unfinished>

2. Bagaimana jika sudah di-hosting di server (VPS/Cloud/Shared Hosting)?

A. VPS/Cloud Hosting (Akses root/SSH)

Solusi Standar:

Gunakan cronjob Linux untuk menjalankan Laravel Scheduler secara otomatis.

Langkah:

Login ke server via SSH.

Jalankan perintah : crontab -e

\* \* \* \* \* cd /path/to/your/project && php artisan schedule:run >> /dev/null 2>&1

Ganti /path/to/your/project dengan path folder Laravel Anda.

Ini akan menjalankan scheduler setiap menit secara otomatis, tanpa perlu terminal terbuka.

B. Shared Hosting (tanpa akses root/SSH)

Biasanya ada menu Cron Jobs di cPanel/Plesk.

Tambahkan cron job baru:

\* \* \* \* \* /usr/local/bin/php /home/username/path/to/your/project/artisan schedule:run >> /dev/null 2>&1

Sesuaikan path PHP dan path project sesuai hosting Anda.

4. Kesimpulan

Jalankan scheduler via cronjob di server (bukan manual di terminal).

Setelah di-setup, website akan berjalan otomatis dan fitur eskalasi akan selalu aktif, meski Anda logout dari SSH/terminal.

User di perangkat manapun tetap bisa pakai website tanpa masalah.

\* \* \* \* \* cd /var/www/html/your-laravel-project && php artisan schedule:run >> /dev/null 2>&1

==========================================================================================

**Tutorial Lengkap: Setup Auto-Start Laravel Scheduler di VM (Linux/Ubuntu)**

**Pendahuluan**

Tutorial ini akan membantu Anda mengatur Laravel Scheduler agar berjalan otomatis di VM Linux/Ubuntu, sehingga laporan yang tidak direspon atau tidak terselesaikan akan otomatis masuk ke pusat tanpa perlu intervensi manual.

**Metode 1: Menggunakan Systemd Service (Direkomendasikan)**

**Langkah 1: Persiapan**

1. **Login ke VM via SSH atau Console**

ssh username@ip-address-vm

1. **Pastikan PHP dan Composer terinstal**

php -v

composer -v

1. **Cek lokasi project Laravel**

# Biasanya di /var/www/html/ atau /home/username/

ls -la /var/www/html/

**Langkah 2: Buat Service File**

1. **Buat file service baru**

sudo nano /etc/systemd/system/laravel-scheduler.service

1. **Copy dan paste konfigurasi berikut**

[Unit]

Description=Laravel Scheduler Service

After=network.target mysql.service

Wants=mysql.service

[Service]

Type=simple

User=www-data

Group=www-data

WorkingDirectory=/var/www/html/JuaraOliviaFinal

ExecStart=/usr/bin/php artisan schedule:work

Restart=always

RestartSec=10

StandardOutput=journal

StandardError=journal

[Install]

WantedBy=multi-user.target

1. Simpan file (Ctrl+X, Y, Enter)

**Langkah 3: Set Permission dan Reload**

1. **Set permission yang benar**

sudo chown www-data:www-data /var/www/html/JuaraOliviaBackEnd

sudo chmod -R 755 /var/www/html/JuaraOliviaBackEnd

sudo chmod -R 775 /var/www/html/JuaraOliviaBackEnd /storage

sudo chmod -R 775 /var/www/html/JuaraOliviaBackEnd /bootstrap/cache

**Reload systemd daemon**

sudo systemctl daemon-reload

**Langkah 4: Enable dan Start Service**

* **Enable service (auto-start saat boot)**

sudo systemctl enable laravel-scheduler

* **Start service**

sudo systemctl start laravel-scheduler

* **Cek status service**

sudo systemctl status laravel-scheduler

**Langkah 5: Testing dan Monitoring**

1. **Cek apakah service berjalan**

ps aux | grep "php artisan schedule:work"

1. **Lihat log real-time**

sudo journalctl -u laravel-scheduler -f

1. **Cek log Laravel**

tail -f /var/www/html/JuaraOliviaBackEnd/storage/logs/laravel.log

**Metode 2: Menggunakan Cron Job (Alternatif)**

**Langkah 1: Buat Script Bash**

1. **Buat file script**

sudo nano /usr/local/bin/laravel-scheduler.sh

1. **Isi dengan script berikut**

#!/bin/bash

cd /var/www/html/JuaraOliviaFinal

php artisan schedule:run >> /var/log/laravel-scheduler.log 2>&1

1. **Set permission**

sudo chmod +x /usr/local/bin/laravel-scheduler.sh

**Langkah 2: Setup Cron Job**

1. **Buka crontab untuk user www-data**

sudo crontab -u www-data -e

1. **Tambahkan baris berikut**

\* \* \* \* \* /usr/local/bin/laravel-scheduler.sh

3. Simpan dan keluar (Ctrl+X, Y, Enter)

**Langkah 3: Pastikan Cron Service Berjalan**

1. **Cek status cron service**

sudo systemctl status cron

1. **Jika belum running, start dan enable**

sudo systemctl start cron

sudo systemctl enable cron

**Metode 3: Menggunakan Supervisor (Advanced)**

**Langkah 1: Install Supervisor**

sudo apt update

sudo apt install supervisor

**Langkah 2: Buat Konfigurasi Supervisor**

1. **Buat file konfigurasi**

sudo nano /etc/supervisor/conf.d/laravel-scheduler.conf

1. **Isi dengan konfigurasi berikut**

[program:laravel-scheduler]

command=php artisan schedule:work

directory=/var/www/html/JuaraOliviaFinal

user=www-data

autostart=true

autorestart=true

redirect\_stderr=true

stdout\_logfile=/var/log/laravel-scheduler.log

logfile\_maxbytes=50MB

logfile\_backups=10

**Langkah 3: Reload dan Start Supervisor**

sudo supervisorctl reread

sudo supervisorctl update

sudo supervisorctl start laravel-scheduler

**Troubleshooting dan Monitoring**

**1. Cek Status Service**

* # Untuk Systemd Service

sudo systemctl status laravel-scheduler

* # Untuk Supervisor

sudo supervisorctl status laravel-scheduler

* # Untuk Cron

sudo systemctl status cron

2. **Lihat Log**

* # Log Systemd

sudo journalctl -u laravel-scheduler -f

# Log Laravel

* tail -f /var/www/html/JuaraOliviaFinal/storage/logs/laravel.log

# Log Cron (jika pakai cron)

* tail -f /var/log/laravel-scheduler.log

**3. Test Command Manual**

cd /var/www/html/JuaraOliviaFinal

php artisan app:escalate-laporan-menunggu

php artisan app:escalate-laporan-tidak-terselesaikan

**4. Cek Process**

# Cek apakah process berjalan

ps aux | grep "php artisan schedule:work"

# Cek memory dan CPU usage

Htop

**Verifikasi Setup**

**Test 1: Buat Laporan Baru**

1. **Buat laporan baru di dinas**
2. **Tunggu 2 menit** (sesuai setting waktu tindakan)
3. **Cek apakah laporan muncul di pusat**

**Test 2: Cek Database**

# Login ke MySQL

mysql -u root -p

# Cek tabel laporan

USE nama\_database;

SELECT \* FROM laporan\_pengaduan WHERE escalated\_to\_pusat = 1;

**Test 3: Cek Log Laravel**

tail -f /var/www/html/JuaraOliviaBackend/storage/logs/laravel.log

**Perintah Penting**

**Management Service (Systemd)**

* # Start service

sudo systemctl start laravel-scheduler

* # Stop service

sudo systemctl stop laravel-scheduler

* # Restart service

sudo systemctl restart laravel-scheduler

* # Enable auto-start

sudo systemctl enable laravel-scheduler

* # Disable auto-start

sudo systemctl disable laravel-scheduler

* # Cek status

sudo systemctl status laravel-scheduler

**Management Supervisor**

* # Start

sudo supervisorctl start laravel-scheduler

* # Stop

sudo supervisorctl stop laravel-scheduler

* # Restart

sudo supervisorctl restart laravel-scheduler

* # Cek status

sudo supervisorctl status laravel-scheduler

**Keamanan dan Best Practices**

**1. Set Permission yang Benar**

# Set ownership

sudo chown -R www-data:www-data /var/www/html/JuaraOliviaBackend

# Set permission

sudo chmod -R 755 /var/www/html/JuaraOliviaBackend

sudo chmod -R 775 /var/www/html/JuaraOliviaBackend/storage

sudo chmod -R 775 /var/www/html/JuaraOliviaBackend/bootstrap/cache

**2. Backup dan Monitoring**

# Backup database secara berkala

mysqldump -u root -p nama\_database > backup\_$(date +%Y%m%d\_%H%M%S).sql

# Monitor disk space

df -h

# Monitor memory usage

free -h

**3. Log Rotation**

# Edit logrotate

sudo nano /etc/logrotate.d/laravel-scheduler

# Tambahkan konfigurasi

/var/log/laravel-scheduler.log {

daily

missingok

rotate 7

compress

notifempty

create 644 www-data www-data

}

**Kesimpulan**

Setelah setup ini selesai:

1. **Laravel Scheduler akan berjalan otomatis** setiap menit
2. **Laporan yang tidak direspon** akan otomatis masuk ke pusat dengan status "Belum Respon"
3. **Laporan yang tidak terselesaikan** akan otomatis masuk ke pusat dengan status "Tidak Terselesaikan"
4. **Service akan auto-restart** jika terjadi error atau crash
5. **Log akan tersimpan** untuk monitoring dan debugging

**Pilih metode yang sesuai dengan kebutuhan Anda:**

* **Systemd Service**: Paling direkomendasikan, mudah di-manage
* **Cron Job**: Sederhana, tapi kurang robust
* **Supervisor**: Advanced, fitur monitoring lebih lengkap