



# Automasi Deployment dan Testing dengan Trigger Repository Menggunakan GitHub Actions dan Docker Compose

## Anggota:

- Wahyu Dwi Laksana Putri – 23515020111004
- Bintang Ula Nur Maghfiroh – 235150207111002
- Shatara Belva Maritza – 235150207111010



# Pendahuluan

Latar belakang:

- Proses deployment manual seringkali lambat, rawan kesalahan, dan tidak efisien.
- Adanya kebutuhan otomasi agar setiap perubahan kode langsung teruji dan ter-deploy tanpa intervensi manusia.

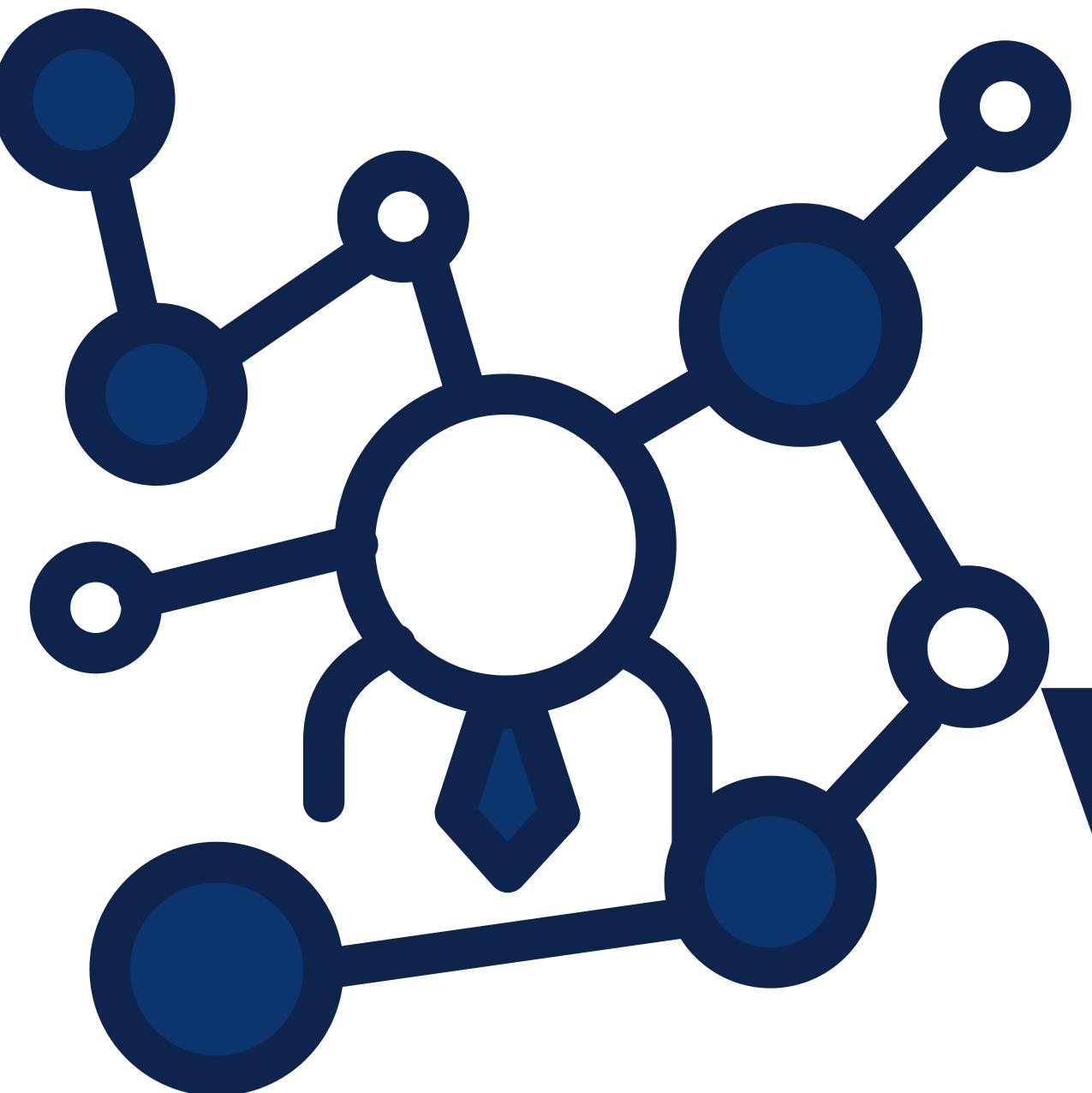
Solusi: Menggunakan GitHub Actions dan Docker Compose sebagai dasar penyediaan automasi dan layanan aplikasi.



# Tentang Website BloodWellness

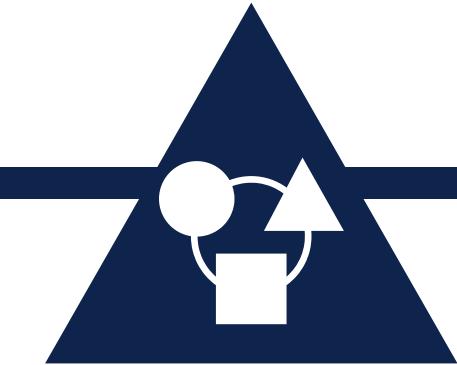
BloodWellness adalah aplikasi web berbasis kalkulator kesehatan harian dan meal planner. Aplikasi ini ditujukan untuk membantu pengguna dalam:

- Menghitung kebutuhan kalori harian berdasarkan data pribadi seperti tinggi badan, berat badan, usia, lemak dan tingkat aktivitas.
- Merancang rencana makan yang sesuai dengan kebutuhan nutrisi pengguna.



# Rumusan Masalah

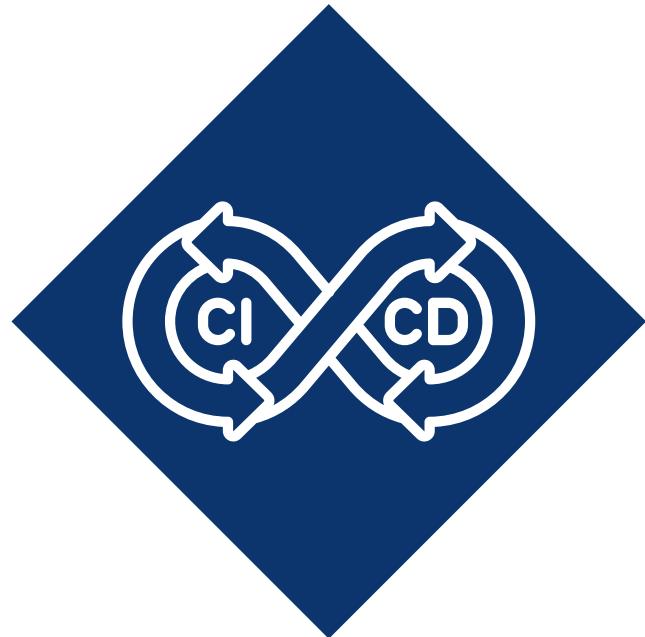
**Bagaimana mengotomatisasi proses build, test, dan deployment aplikasi agar tidak memerlukan intervensi manual?**



**Bagaimana memastikan bahwa setiap perubahan kode langsung diuji dan diterapkan ke server?**

**Bagaimana menyederhanakan penyediaan layanan multi-komponen seperti backend, database, dan frontend?**

# TUJUAN



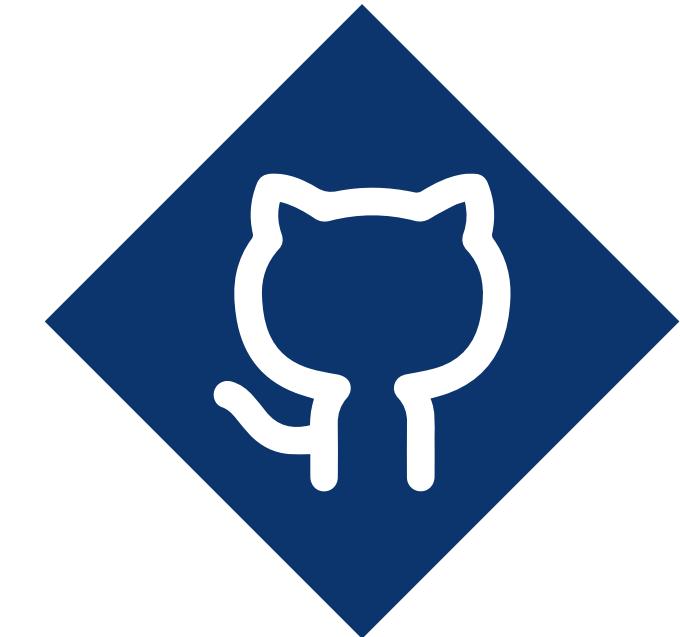
## Efisiensi Integrasi

Membangun **pipeline CI/CD** yang terotomatisasi sepenuhnya, yang memproses build, testing, dan deployment secara langsung setelah ada aktivitas di repository GitHub.



## Container

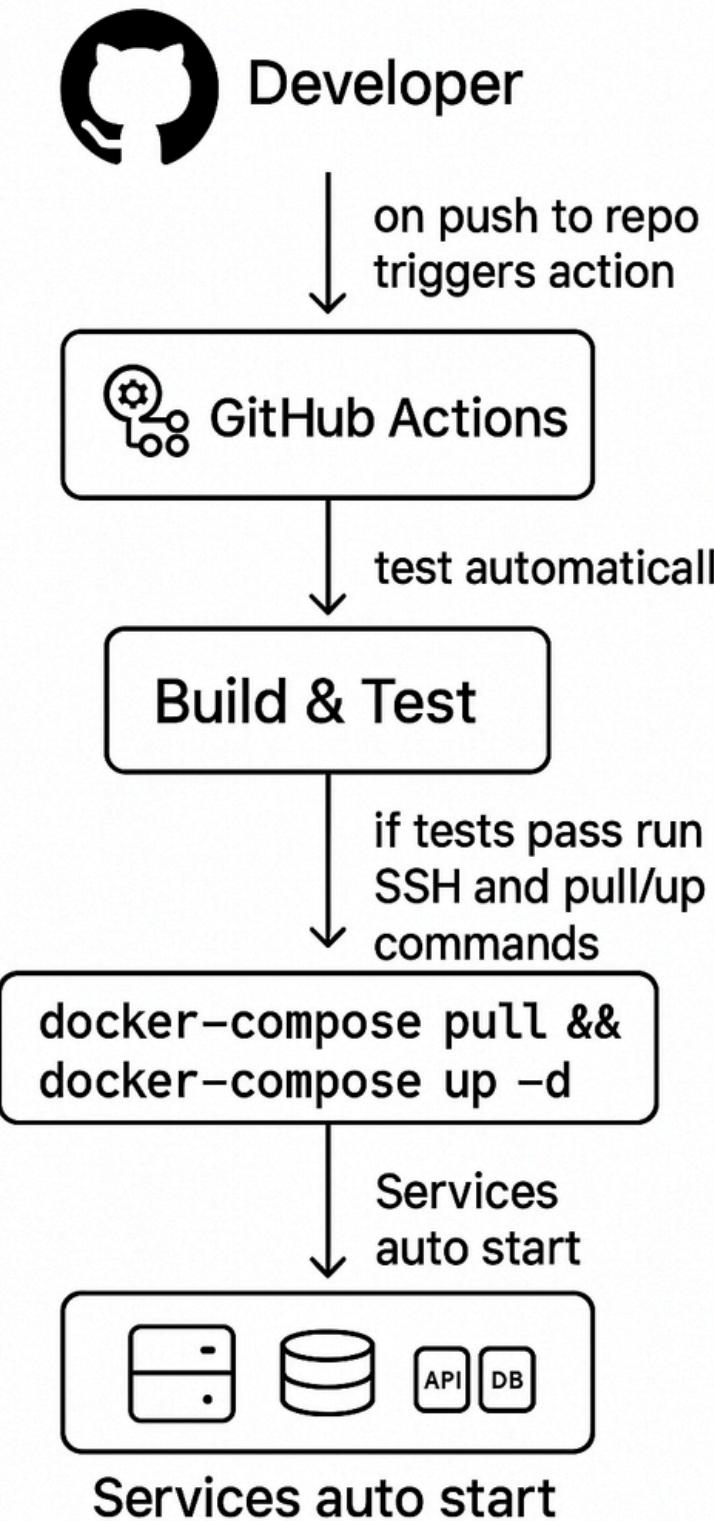
Menyediakan layanan aplikasi berbasis container menggunakan **Docker** dan **Docker Compose** agar proses provisioning dan manajemen layanan menjadi lebih konsisten dan portabel.



## Automatisasi CI/CD

Membangun sistem deployment otomatis berbasis aktivitas di repository menggunakan workflow **GitHub Actions** yang dapat berjalan tanpa intervensi manual.

# Desain Solusi



## Alur Otomasi CI/CD dengan GitHub Actions

1. Push Kode
  - Developer melakukan push kode terbaru ke GitHub Repository (ikon GitHub).
2. Trigger Workflow
  - Setiap push otomatis memicu workflow GitHub Actions.
3. Build dan Test
  - GitHub Actions menjalankan tahap build (membuat image Docker)
  - Dilanjutkan dengan test (unit/integration tests) secara otomatis
4. Deploy ke Server
  - Jika semua test berhasil (tests pass), GitHub Actions tersambung ke server (VPS) melalui SSH
  - Menjalankan perintah:
    - git pull untuk menarik kode terbaru dari repository
    - docker-compose down dan docker-compose up -d --build untuk menjalankan ulang container dengan image terbaru
    - php artisan migrate, cache:clear, dan config:cache untuk memastikan database dan konfigurasi aplikasi sudah diperbarui
5. Layanan Berjalan Otomatis
  - Semua service (API, database, dsb.) otomatis start
  - Hasil deploy dapat dipantau dan diakses langsung

# Rencana Penggunaan GitHub Actions

- **Workflow file:** .github/workflows/ci-cd.yml
- **Trigger:** on: push ke branch main
- **Job 1 – Build & Test :** Checkout kode, build Docker image, jalankan unit/integration tests
- **Job 2 – Deploy** (hanya jika Job 1 sukses) : Setup SSH (via secret SSH\_PRIVATE\_KEY)SSH ke server
- **Secrets yang dibutuhkan :** SSH\_PRIVATE\_KEY, SERVER\_HOST, SERVER\_USER



# Pembagian Tugas

Anggota 1

Install dan konfigurasi Docker + Docker Compose di server tujuan.

Menyiapkan file docker-compose.yml berisi service seperti backend, frontend, dan database.

Menentukan struktur arsitektur sistem (misalnya seperti yang digambarkan di slide arsitektur).

Pastikan semua container berjalan dan saling terhubung dengan benar.

Anggota 2

Menulis file .github/workflows/deploy.yml di GitHub.

Menentukan trigger (misal: saat push ke branch main).

Buat job: build → test → deploy ke server via SSH.

Pastikan GitHub Actions bisa mengakses server (via SSH key atau token).

Anggota 3

Membuat dokumentasi proses: setup server, CI/CD, dan cara kerja sistem.

Menyusun diagram alur workflow (dari push sampai aplikasi jalan).

Menyusun dan melakukan testing: apakah build & deploy otomatis bekerja saat kode di-push.

Dokumentasikan hasil testing, error log, dan perbaikannya (jika ada).

# PROGRESS

```
ASUS@WAHYU MINGW64 ~/Downloads (main)
$ ssh -i SSH-KEY-Jenkins.pem ubuntu@3.86.104.59
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1024-aws x86_64)

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

System information as of Mon Jun  2 10:00:03 UTC 2025

System load: 0.2          Processes:      110
Usage of /: 76.5% of 6.71GB Users logged in:   0
Memory usage: 47%         IPv4 address for enx0: 172.31.82.120
Swap usage:  0%          

Expanded Security Maintenance for Applications is not enabled.

40 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

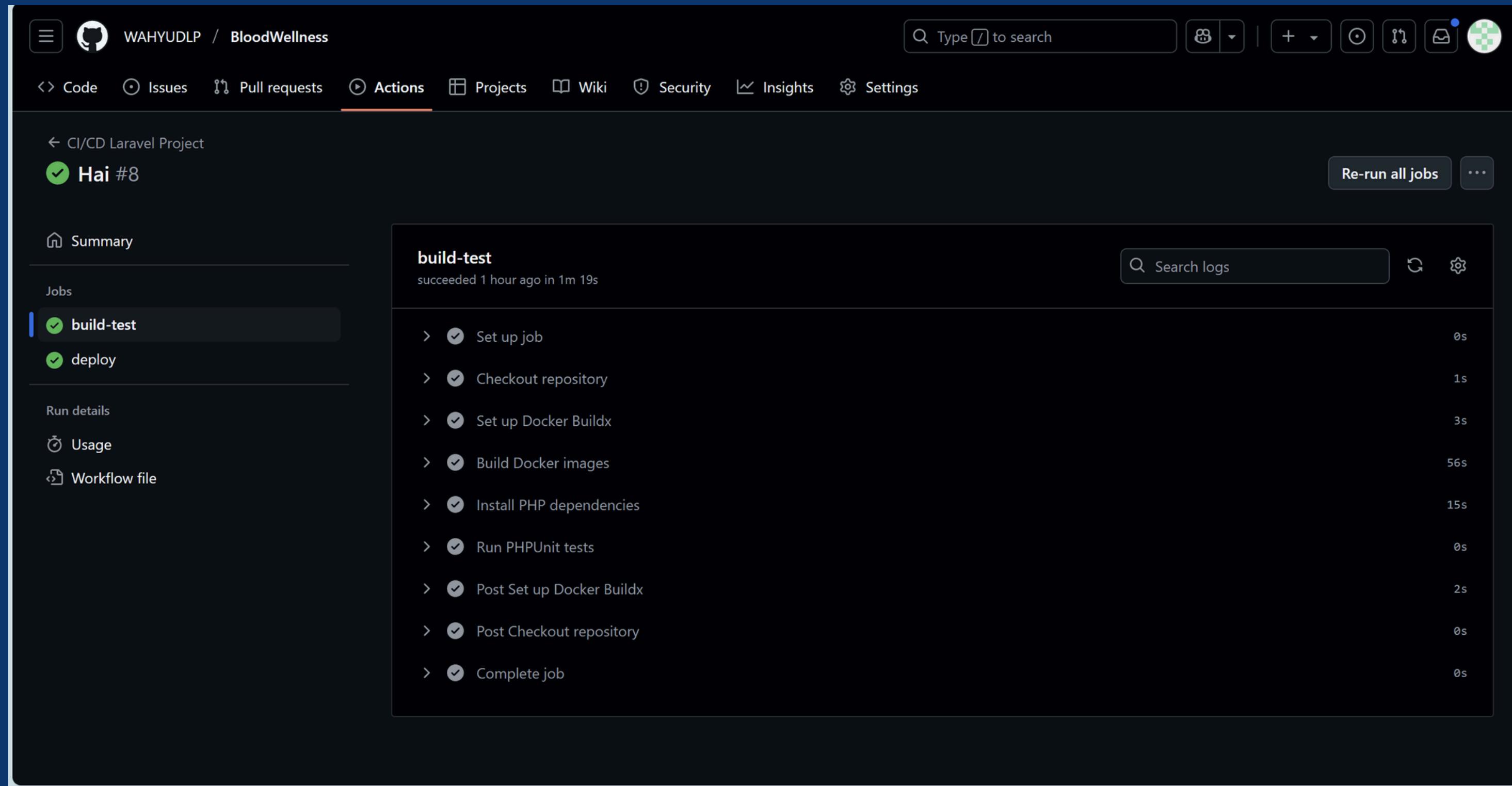
*** System restart required ***
Last login: Fri May 30 10:04:43 2025 from 182.253.58.100
```

Inbound rules (3)			
Protocol	Port range	Source	Description
TCP	80	0.0.0.0/0	-
TCP	8080	0.0.0.0/0	-
TCP	22	0.0.0.0/0	-

Instance summary <small>Info</small>		
Instance ID	Public IPv4 address	Private IPv4 addresses
<a href="#">i-030fcf4de6f723cae</a>	<a href="#">3.86.104.59   open address</a>	<a href="#">172.31.82.120</a>
IPv6 address	Instance state	Public DNS
-	<span>Running</span>	<a href="#">ec2-3-86-104-59.compute-1.amazonaws.com   open address</a>
Hostname type	Private IP DNS name (IPv4 only)	Elastic IP addresses
IP name: ip-172-31-82-120.ec2.internal	<a href="#">ip-172-31-82-120.ec2.internal</a>	-
Answer private resource DNS name	Instance type	AWS Compute Optimizer findings
IPv4 (A)	t2.micro	
Auto-assigned IP address	VPC ID	
<a href="#">3.86.104.59 [Public IP]</a>	<a href="#">vpc</a>	

# PROGRESS



The screenshot shows a GitHub Actions workflow run for a project named "Hai #8". The workflow consists of two jobs: "build-test" and "deploy". The "build-test" job has succeeded 1 hour ago in 1m 19s. It contains the following steps:

- > ✓ Set up job 0s
- > ✓ Checkout repository 1s
- > ✓ Set up Docker Buildx 3s
- > ✓ Build Docker images 56s
- > ✓ Install PHP dependencies 15s
- > ✓ Run PHPUnit tests 0s
- > ✓ Post Set up Docker Buildx 2s
- > ✓ Post Checkout repository 0s
- > ✓ Complete job 0s

There is also a "deploy" job listed in the sidebar.

<https://github.com/WAHYUDLP/BloodWellness>

# PROGRESS

```
ubuntu@ip-172-31-82-120:~/Bloodwellness$ git log -1
commit 268ad64a1f41a61bd0371197f16a47a95c9c2e06 (HEAD -> main, origin/main, origin/HEAD)
Author: WAHYUDLP <149256633+WAHYUDLP@users.noreply.github.com>
Date:   Mon Jun 2 17:39:44 2025 +0700
```

Hai

```
ubuntu@ip-172-31-82-120:~/Bloodwellness$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
a988f77cb52b	bloodwellness-web	"docker-php-entrypoi..."	2 hours ago	Up 33 minutes	0.0.0.0:8080->80/tcp, [::]:8080->80/tcp	bloodwellness-web-1
88432c9f26d9	mysql:5.7	"docker-entrypoint.s..."	2 hours ago	Up 2 hours	0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp	bloodwellness-db-1

```
ubuntu@ip-172-31-82-120:~/Bloodwellness$ |
```

```
ubuntu@ip-172-31-82-120:~/BloodWellness$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
a988f77cb52b	bloodwellness-web	"docker-php-entrypoi..."	About an hour ago	Up 11 minutes	0.0.0.0:8080->80/tcp, [::]:8080->80/tcp	bloodwellness-web-1
88432c9f26d9	mysql:5.7	"docker-entrypoint.s..."	About an hour ago	Up About an hour	0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp	bloodwellness-db-1

```
ubuntu@ip-172-31-82-120:~/BloodWellness$
```

# PROGRESS

The screenshot shows a web browser window with the URL [3.86.104.59:8080](http://3.86.104.59:8080). The page is titled "BloodWellness". The header includes a navigation bar with links to "Beranda" (underlined), "Kalkulator Kalori", "Perencana Makanan", "Profil", and "Keluar". The main content features a large background image of various fruits and vegetables like avocados, lemons, and berries. A central logo consists of a stylized flower inside a red circle. Below the logo, the text reads "BloodWellness" and "Temukan pola makan sehat sesuai tipe darah Anda!". A descriptive paragraph follows: "Dengan fitur hitung kalori dan meal planner otomatis, BloodWellness membantu Anda merencanakan menu harian lengkap dengan resep yang lezat dan bergizi." At the bottom left, it says "© 2025 BloodWellness". The browser's address bar shows "Not secure" and the IP address.

# PROGRESS

The screenshot shows a web browser window with the URL [18.208.180.241:8080/login](http://18.208.180.241:8080/login). The page is titled "Not secure". The main content area features a dark background with various fruits and leaves, including a lemon, a lime, an avocado, and a grapefruit. On the left, there is a logo for "BloodWellness" and text in Indonesian: "Temukan pola makan sehat sesuai tipe darah Anda!" (Discover a healthy eating pattern suitable for your blood type) and "Dengan fitur hitung kalori dan meal planner otomatis, BloodWellness membantu Anda merencanakan menu harian lengkap dengan resep yang lezat dan bergizi." (With calorie calculation and automatic meal planner features, BloodWellness helps you plan a full daily menu with delicious and nutritious recipes). The top navigation bar includes links for "Beranda", "Kalkulator Kalori", "Perencana Makanan", and "Profil". On the right, there are "Masuk" and "Daftar" buttons. A modal window titled "AYO MULAI SEKARANG di PAL!" contains fields for "Email" and "Kata Sandi", a "Forgot My Password" link, a "Remember Me" checkbox, a "MASUK" button, an "Or" link, and a "Sign in with Google" button. Below the modal, a link says "Belum punya akun? DAFTAR DI SINI". The browser's address bar shows the URL again, and the toolbar includes links for Google, Gmail, Translate, YouTube, and several local applications.

# Masalah



# Masalah dan Solusi

8 workflow runs			
	Event	Status	Branch
✓ Hai	CI/CD Laravel Project #8: Commit <a href="#">268ad64</a> pushed by WAHYUDLP	<span>main</span>	1 hour ago 2m 31s
✓ Iniii	CI/CD Laravel Project #7: Commit <a href="#">d9904cf</a> pushed by WAHYUDLP	<span>main</span>	1 hour ago 1m 21s
✗ Ini	CI/CD Laravel Project #6: Commit <a href="#">b84d241</a> pushed by WAHYUDLP	<span>main</span>	1 hour ago 1m 35s
✗ Up	CI/CD Laravel Project #5: Commit <a href="#">7f662c1</a> pushed by WAHYUDLP	<span>main</span>	2 hours ago 1m 1s
✗ ter	CI/CD Laravel Project #4: Commit <a href="#">d4e2aa5</a> pushed by WAHYUDLP	<span>main</span>	2 hours ago 1m 36s
✗ Terbaru	CI/CD Laravel Project #3: Commit <a href="#">34fe01a</a> pushed by WAHYUDLP	<span>main</span>	2 hours ago 1m 7s
✗ Terbaru	CI/CD Laravel Project #2: Commit <a href="#">938b182</a> pushed by WAHYUDLP	<span>main</span>	2 hours ago 1m 0s
✗ Update	CI/CD Laravel Project #1: Commit <a href="#">b5da287</a> pushed by WAHYUDLP	<span>main</span>	2 hours ago 13s

1. Terjadi gagal pada saat build dan test pada file **Dockerfile** dikarenakan belum menjalankan composer install di dalam container. Oleh karena itu perlu menambahkan:

```
RUN composer install --no-dev --optimize-autoloader
```

2. Pada **Dockerfile** Composer gagal karena ekstensi ext-bcmath tidak tersedia. Oleh karena itu perlu menambahkan:

```
RUN apt-get update && apt-get install -y \
    libzip-dev unzip curl git \
    && docker-php-ext-install pdo pdo_mysql zip bcmath
```

# Masalah

<http://3.86.104.59:8080/> tampilan website tidak muncul, ketika di cek mengalami 500 internal server error.

Laravel tidak bisa membuka atau menulis ke

- /var/www/html/storage/logs/laravel.log
- /var/www/html/storage/framework/views/...

karena **permission (izin akses)**

# Solusi

## Cara memperbaiki

1. Masuk ke container: docker exec -it bloodwellness-web-1 bash
2. Menjalankan 2 perintah untuk memberikan izin akses:
  - chown -R www-data:www-data /var/www/html/storage /var/www/html/bootstrap/cache
  - chmod -R 775 /var/www/html/storage /var/www/html/bootstrap/cache
3. Keluar dari container dan lakukan restart: docker restart bloodwellness-web-1
4. Website bisa di akses

```
ubuntu@ip-172-31-82-120:~$ docker exec -it bloodwellness-web-1 bash
root@a988f77cb52b:/var/www/html# chown -R www-data:www-data /var/www/html/storage /var/www/html/bootstrap/cache
root@a988f77cb52b:/var/www/html# chmod -R 775 /var/www/html/storage /var/www/html/bootstrap/cache
root@a988f77cb52b:/var/www/html# exit
exit
ubuntu@ip-172-31-82-120:~$ docker restart bloodwellness-web-1
```

## Penjelasan:

- www-data adalah user web server Apache/Nginx dalam container PHP biasanya.
- chown mengubah pemilik folder/file ke www-data.
- chmod 775 memberikan izin baca-tulis-eksekusi untuk pemilik dan grup.



Teknik Informatika II Fakultas Ilmu Komputer

# Thank You

For your attention and cooperation



Penyediaan Automasi dan Layanan

