

# Temmy Alex

Programming Enthusiast

Focus in Web Development

Currently work in KoinWorks as Backend Engineer



**Temmy Alex**

8 years experience as Web  
Developer

## Education Background



**UNIVERSITAS  
BUDI LUHUR**

**2011-2015**

**Bachelor Degree**

Information System

# CI/CD with GitLab

- ☐ Introduction CI/CD
- ☐ Using Gitlab CI/CD

# CI/CD with GitLab



**Introduction CI/CD**



Using GitLab CI/CD

# Apa itu Automation Server?

Sebelum adanya automation server proses deployment aplikasi dilakukan secara manual, proses deployment yang dilakukan secara manual ini dapat menimbulkan kesalahan dan proses yang tidak konsisten (seperti menjalankan unit test atau download dependency)

# Service Automation Server

Berikut beberapa service Automation Server yang dapat digunakan

1. Jenkins (<https://www.jenkins.io/>)
2. Gitlab CI/CD (<https://docs.gitlab.com/ee/ci/>)
3. Github Action (<https://github.com/features/actions>)
4. Circle CI (<https://circleci.com/>)

Dan tentunya masih banyak service automation lainnya selain daftar diatas

# CI/CD with GitLab



**Introduction CI/CD**



Using GitLab CI/CD



# Apa itu CI/CD?

Continuous integration (CI) merupakan metode pengintegrasian kode dalam repositori dan melakukan pengujian secara otomatis, tentunya proses CI ini berjalan melalui perintah commit

Kemudian Continuous Delivery (CD) merupakan proses lanjutan dari proses CI yang dimana proses ini berfokus pada proses build (proses perilisan kode)

# GitLab CI/CD

Merupakan salah satu tools yang disediakan oleh gitlab yang tentu nya terdapat fitur penggunaan ci/cd namun tools ini hanya support untuk server berbasis Linux (CentOS, Debian, Ubuntu)



# Study Case

1. Buatlah repository pada gitlab
2. Dan push project api nodejs ke dalam gitlab
3. Dan buatlah proses pipeline ci/cd pada gitlab

# Study Case Explanation

1. Buat repository pada gitlab dan push project api menggunakan git seperti pada gambar dibawah ini

```

• → apitodo git:(master) x git add .
• → apitodo git:(master) x git commit -m "config ci/cd"
  [master a41cba6] config ci/cd
    3 files changed, 9 insertions(+), 13 deletions(-)
• → apitodo git:(master) git push origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 541 bytes | 541.00 KiB/s, done.
Total 5 (delta 3), reused 0 (delta 0)
To https://gitlab.com/temmyalexhimsself/ci_cd_todoapi.git
  7e7b33b..a41cba6  master -> master
  
```

# Study Case Explanation

2. Kemudian tambahkan Dockerfile yang berada dalam project

```

Dockerfile M X
Dockerfile > ...
1 FROM node:lts-alpine3.17
2
3 WORKDIR /apitodo/src/app
4
5 COPY package*.json ./
6
7 COPY . .
8
9 EXPOSE 8090
10
11 CMD [ "node", "app.js" ]

```

# Study Case Explanation

3. Kemudian build Dockerfile menggunakan perintah docker build -t apitodo .

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

● → apitodo git:(master) x docker build -t apitodo .
Sending build context to Docker daemon 29.88MB
Step 1/6 : FROM node:lts-alpine3.17
--> f520ad35ba68
Step 2/6 : WORKDIR /apitodo/src/app
--> Running in d5578e4295b6
Removing intermediate container d5578e4295b6
--> 0299b8bb03b3
Step 3/6 : COPY package*.json ./
--> 5d654b658e8c
Step 4/6 : COPY . .
--> a94d7946d629
Step 5/6 : EXPOSE 8090
--> Running in f8f9b50a3bc9
Removing intermediate container f8f9b50a3bc9
--> 56b30acf48f3
Step 6/6 : CMD [ "node", "app.js" ]
--> Running in 5e684d32d506
Removing intermediate container 5e684d32d506
--> dbaalca0b4f8
Successfully built dbaalca0b4f8
Successfully tagged apitodo:latest

```

# Study Case Explanation

3. Kemudian pastikan docker images sudah berhasil di build menggunakan perintah **docker images apitodo**

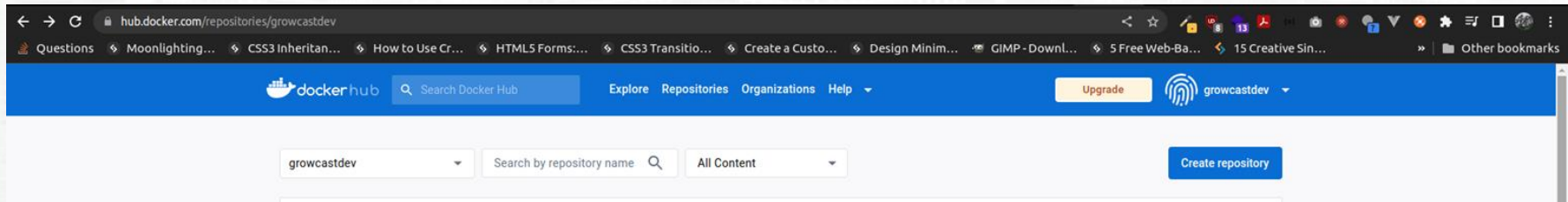
```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

● → apitodo git:(master) x docker images apitodo
REPOSITORY    TAG       IMAGE ID      CREATED        SIZE
apitodo       latest   dbaa1ca0b4f8  2 minutes ago  201MB
  
```

# Study Case Explanation

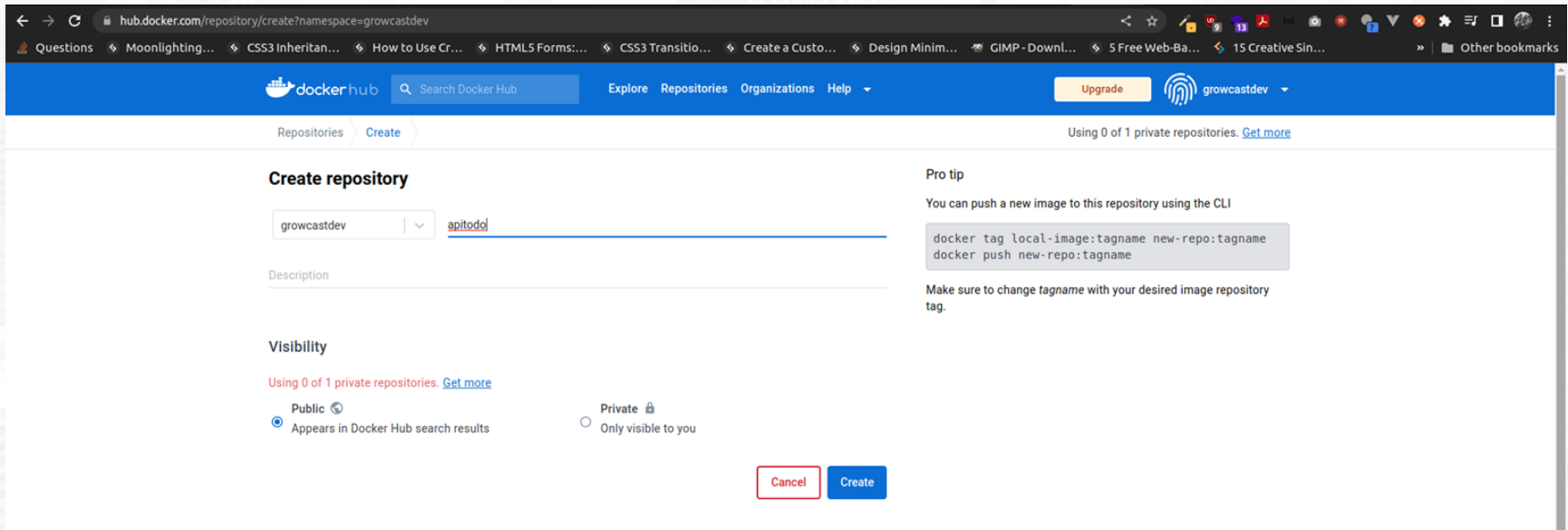
4. Buatlah akun pada dockerhub untuk persiapan push docker di local ke repo dockerhub dengan klik button **Create Repository**





# Study Case Explanation

5. Kemudian buat repository dengan nama apitodo



hub.docker.com/repository/create?namespace=growcastdev

Questions Moonlighting... CSS3 Inheritan... How to Use Cr... HTML5 Forms... CSS3 Transio... Create a Cust... Design Minim... GIMP - Downl... 5 Free Web-Ba... 15 Creative Sin... Other bookmarks

dockerhub Search Docker Hub Explore Repositories Organizations Help Upgrade growcastdev

Repositories Create Using 0 of 1 private repositories. [Get more](#)



### Create repository

growcastdev

Description

### Visibility

Using 0 of 1 private repositories. [Get more](#)

☒ Public  Appears in Docker Hub search results
 ☐ Private  Only visible to you

Pro tip

You can push a new image to this repository using the CLI

```
docker tag local-image:tagname new-repo:tagname
docker push new-repo:tagname
```

Make sure to change tagname with your desired image repository tag.

Cancel Create

# Study Case Explanation

- Kemudian ubah tag docker di local sesuai dengan nama pada repo dockerhub menggunakan perintah **docker tag apitodo (sesuai dengan nama repo masing-masing)**



```

• → apitodo git:(master) x docker images apitodo
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
apitodo        latest    dbaalca0b4f8   10 minutes ago 201MB
• → apitodo git:(master) x docker tag apitodo growcastdev/apitodo:latest
  
```

# Study Case Explanation

7. Kemudian login menggunakan akun docker (sesuaikan dengan akun pada dockerhub)

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
```

● → **apitodo** **git:(master)** ✕ `docker login --username=growcastdev`  
Password:  
WARNING! Your password will be stored unencrypted in /home/dell/.docker/config.json.  
Configure a credential helper to remove this warning. See  
<https://docs.docker.com/engine/reference/commandline/login/#credentials-store>  
  
Login Succeeded

# Study Case Explanation

- Kemudian push docker yang berada di local ke repo dockerhub yang sudah dibuat

```


PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL


• → apitodo git:(master) x docker push growcastdev/apitodo:latest
The push refers to repository [docker.io/growcastdev/apitodo]
d03a56540b4f: Pushed
58f15fc24ddf: Pushed
46101ef95cce: Pushed
43db81ae3a73: Mounted from growcastdev/apitodo_cicd_sample
cc9a0a9b7a43: Mounted from growcastdev/apitodo_cicd_sample
2e65656323bb: Mounted from growcastdev/apitodo_cicd_sample
7cd52847ad77: Mounted from growcastdev/apitodo_cicd_sample
latest: digest: sha256:26cd540d3b1854abeddb4ba90d3bd590fae8e9245baa3f04adde2b02e6acfd size: 1785


```

# Study Case Explanation

- Kemudian pastikan repo pada dockerhub berhasil diupdate sesuai dengan versi terbaru


**growcastdev / apitodo**

**Description**  
*This repository does not have a description* 

 Last pushed: 2 minutes ago

**Docker commands** [Public View](#)  
 To push a new tag to this repository,  

```
docker push growcastdev/apitodo:tagname
```

**Tags**  IMAGE INSIGHTS INACTIVE [Activate](#)

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
 latest		Image	---	2 minutes ago

[See all](#)
[Go to Advanced Image Management](#)

**Automated Builds**  
 Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.  
 Available with Pro, Team and Business subscriptions. [Read more about automated builds.](#)  
[Upgrade](#)

# Study Case Explanation


10. Kemudian masuk ke menu **Settings > CI/CD** dan disana terdapat link dan registrasi token yang dapat digunakan untuk mengaktifkan gitlab-runner dan dapat kita lihat pada bagian berikut gitlab runners belum aktif

## Project runners


These runners are assigned to this project.

### Set up a project runner for a project

1. [Install GitLab Runner and ensure it's running.](#)
2. Register the runner with this URL:

<https://gitlab.com/> 

And this registration token:

`GR1348941Ri_oyrB1bBj3syvozdrE` 

Reset registration token


Show runner installation instructions

## Assigned project runners

 #21835759 (eyT21sn16) 

apitodo\_cicd

[build](#) [deploy](#)

  [Remove runner](#)



# Study Case Explanation

11. Untuk mengaktifkan fitur gitlab runner kalian perlu melakukan install gitlab-runner dengan mengikuti step by step pada link dibawah ini

<https://docs.gitlab.com/runner/install/windows.html>

# Study Case Explanation

- Kemudian jika proses instalasi selesai jalankan perintah **gitlab-runner register** serta isikan link dan registrasi token yang berasal dari menu **Settings > CI/CD**

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
• → apitodo git:(master) x gitlab-runner register
Runtime platform arch=amd64 os=linux pid=396519 revision=d540b510 version=15.9.1
WARNING: Running in user-mode.
WARNING: The user-mode requires you to manually start builds processing:
WARNING: $ gitlab-runner run
WARNING: Use sudo for system-mode:
WARNING: $ sudo gitlab-runner...

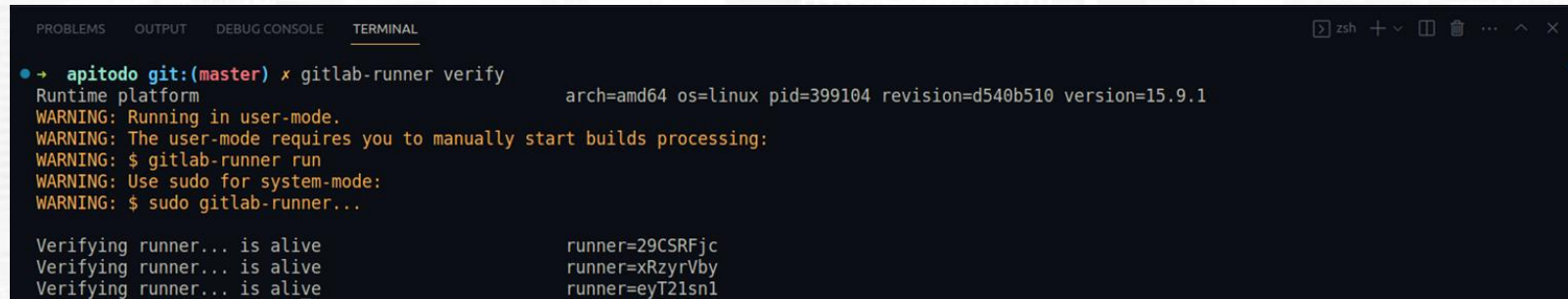
Enter the GitLab instance URL (for example, https://gitlab.com/):
https://gitlab.com/
Enter the registration token:
GR1348941Ri_oyrB1bJ3syvozdrE
Enter a description for the runner:
[dell-Vostro-3400]: apitodo_cicd
Enter tags for the runner (comma-separated):
build, deploy
Enter optional maintenance note for the runner:
shell
WARNING: Support for registration tokens and runner parameters in the 'register' command has been deprecated in GitLab Runner 15.6 and will be replaced with support for authentication tokens. For more information, see https://gitlab.com/gitlab-org/gitlab/-/issues/380872
Registering runner... succeeded runner=GR1348941Ri_oyrB1
Enter an executor: parallels, ssh, virtualbox, docker+machine, instance, custom, docker, docker-ssh, shell, docker-ssh+machine, kubernetes:
shell
Runner registered successfully. Feel free to start it, but if it's running already the config should be automatically reloaded!

Configuration (with the authentication token) was saved in "/home/dell/.gitlab-runner/config.toml"

```

# Study Case Explanation

13. Kemudian jalankan perintah **gitlab-runner verify** untuk mengaktifkan gitlab runner



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
• → apitodo git:(master) * gitlab-runner verify
Runtime platform                                arch=amd64 os=linux pid=399104 revision=d540b510 version=15.9.1
WARNING: Running in user-mode.
WARNING: The user-mode requires you to manually start builds processing:
WARNING: $ gitlab-runner run
WARNING: Use sudo for system-mode:
WARNING: $ sudo gitlab-runner...

Verifying runner... is alive                    runner=29CSRFjc
Verifying runner... is alive                    runner=xRzyrVby
Verifying runner... is alive                    runner=eyT2lsn1
```

# Study Case Explanation

```

.gitlab-ci.yml M X
.gitlab-ci.yml
1 image: docker:latest
2
3 variables:
4   REPOSITORY_URL: growcastdev/apitodo
5   IMAGE_TAG: latest
6
7 services:
8   - docker:dind
9
10 before_script:
11   - docker login --username=$DH_USERNAME --password=$DH_PASSWORD
12
13 stages:
14   - build
15   - deploy
16
17 build:
18   stage: build
19   script:
20     - echo "Building image..."
21     - docker build
22     -t apitodo:latest
23     --build-arg PORT=$PORT
24     --no-cache
25     - echo "Tagging image..."
26     - docker tag apitodo:latest $REPOSITORY_URL:latest
27     - echo "Pushing image..."
28     - docker push $REPOSITORY_URL:latest
29   only:
30     - master
31
32 deploy:
33   stage: deploy
34   script:
35     - docker stop apitodo || true && docker rm apitodo || true
36     - docker container run -dt -p $PORT:$PORT --name apitodo $REPOSITORY_URL:latest
37     - docker image prune -f
38   only:
39     - master

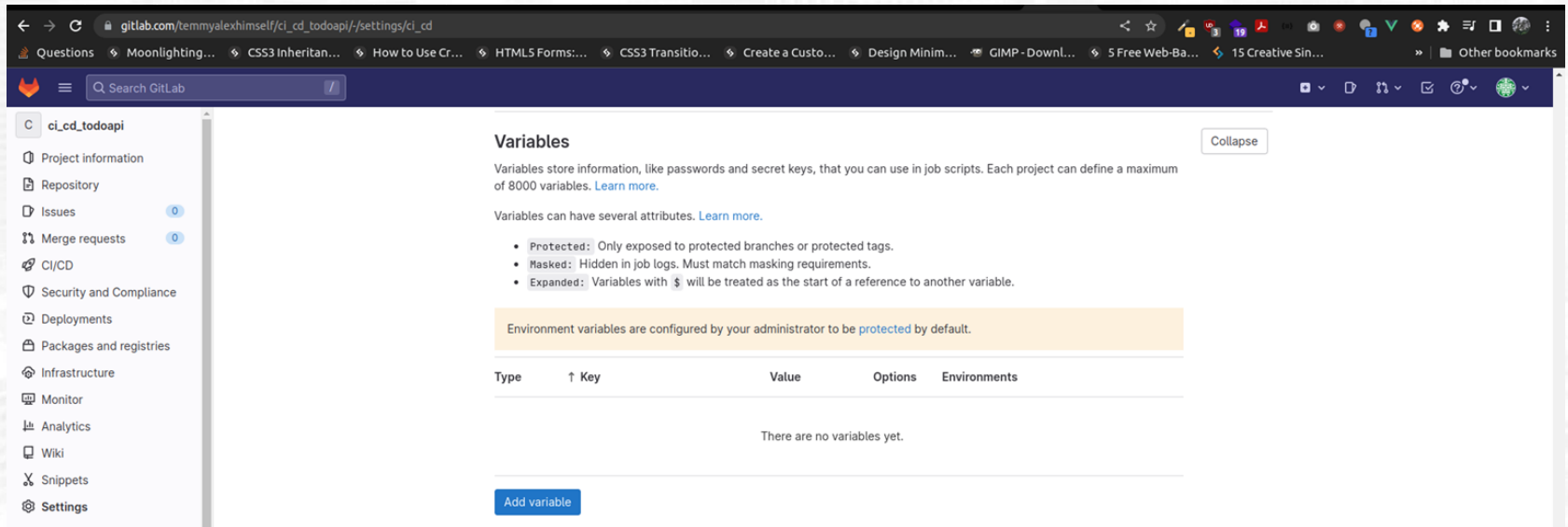
```

14. Kemudian tambahkan serta setup file

**.gitlab-ci.yml (extension menggunakan .yml)**

# Study Case Explanation

15. Kemudian tambahkan variable untuk dijalankan pada **.gitlab-ci.yml**



The screenshot shows the GitLab web interface. The left sidebar contains a navigation menu with options like Project information, Repository, Issues, Merge requests, CI/CD, Security and Compliance, Deployments, Packages and registries, Infrastructure, Monitor, Analytics, Wiki, Snippets, and Settings. The main content area is titled 'Variables' and includes a 'Collapse' button. Below the title, there is a description of variables and a list of attributes: Protected, Masked, and Expanded. A yellow box indicates that environment variables are configured by the administrator to be protected by default. Below this, there is a table with columns for Type, Key, Value, Options, and Environments. The table is currently empty, showing 'There are no variables yet.' At the bottom of the table, there is a blue button labeled 'Add variable'.

**Variables** Collapse

Variables store information, like passwords and secret keys, that you can use in job scripts. Each project can define a maximum of 8000 variables. [Learn more.](#)

Variables can have several attributes. [Learn more.](#)

- Protected:** Only exposed to protected branches or protected tags.
- Masked:** Hidden in job logs. Must match masking requirements.
- Expanded:** Variables with `$` will be treated as the start of a reference to another variable.

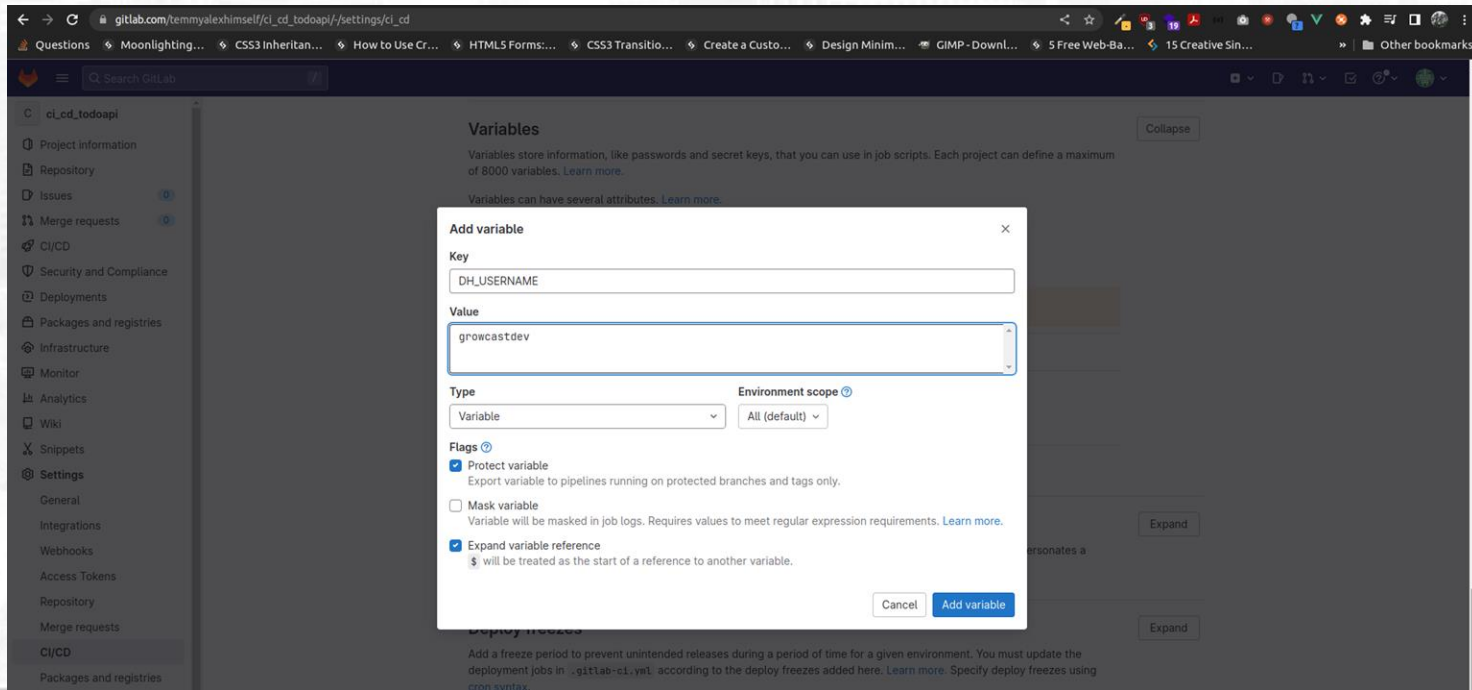
Environment variables are configured by your administrator to be **protected** by default.

Type	↑ Key	Value	Options	Environments
There are no variables yet.				

[Add variable](#)

# Study Case Explanation

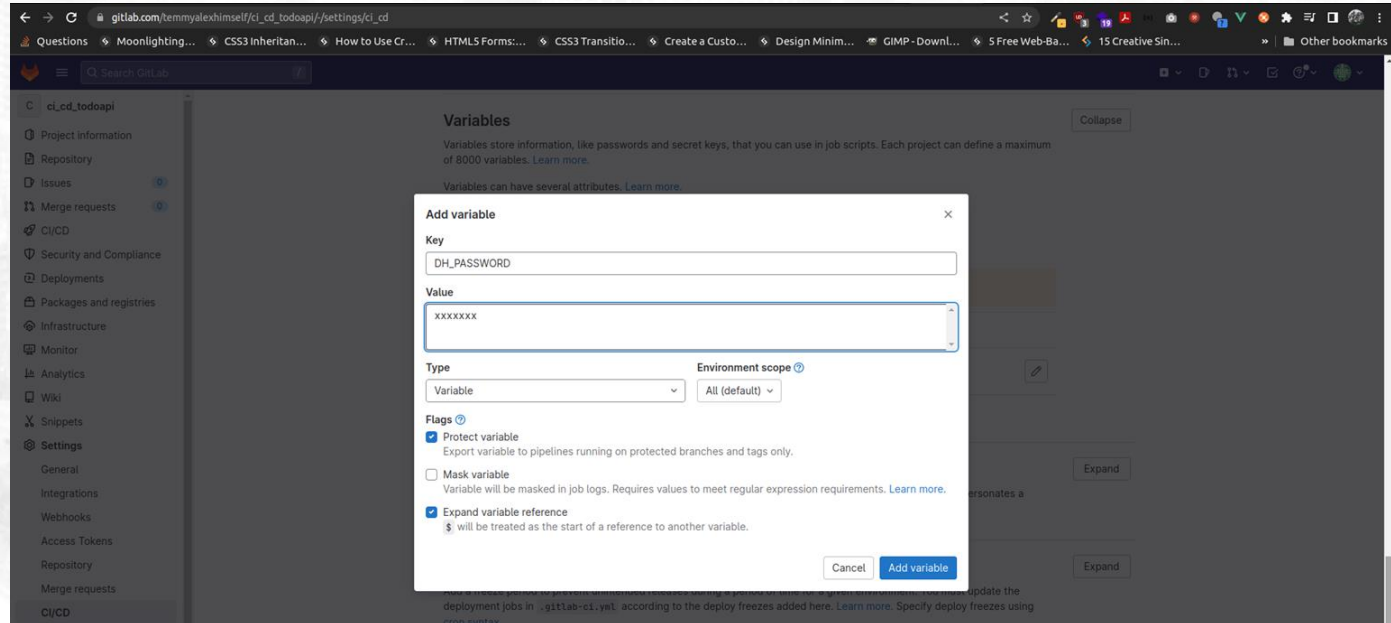
- Kemudian tambahkan variable **DH\_USERNAME** isikan value sesuai dengan username dockerhub





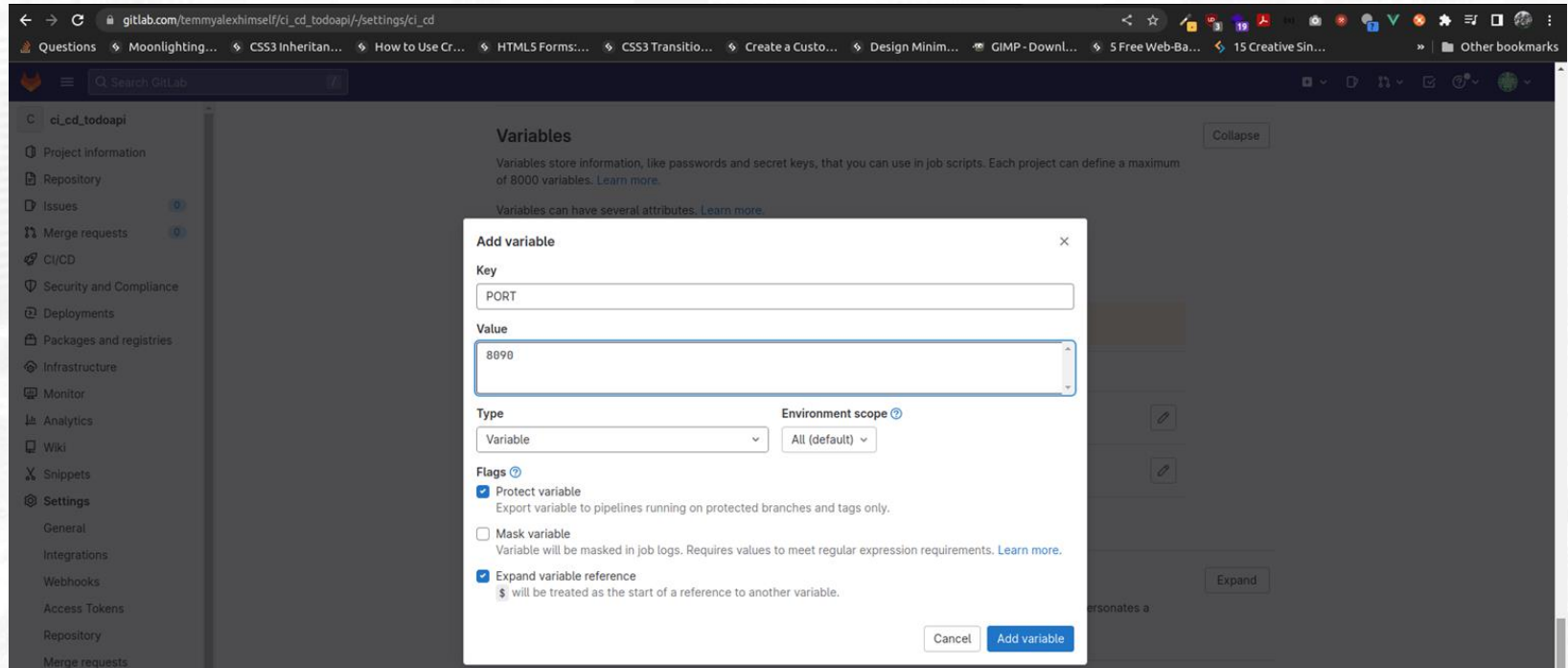
# Study Case Explanation

17. Kemudian tambahkan variable **DH\_PASSWORD** isikan value sesuai dengan password dockerhub



# Study Case Explanation

18. Kemudian tambahkan variable **PORT**



The screenshot shows the GitLab CI/CD settings page for a project named 'ci\_cd\_todoapi'. The 'Variables' section is active, displaying a modal dialog for adding a new variable. The dialog contains the following fields and options:

- Key:** A text input field containing the value 'PORT'.
- Value:** A text input field containing the value '8090'.
- Type:** A dropdown menu set to 'Variable'.
- Environment scope:** A dropdown menu set to 'All (default)'.
- Flags:**
  - ☒ **Protect variable**: Export variable to pipelines running on protected branches and tags only.
  - ☐ **Mask variable**: Variable will be masked in job logs. Requires values to meet regular expression requirements. [Learn more.](#)
  - ☒ **Expand variable reference**: \$ will be treated as the start of a reference to another variable.

At the bottom of the dialog, there are two buttons: 'Cancel' and 'Add variable'.

# Study Case Explanation

## 19. Variable berhasil ditambahkan

### Variables













[Collapse](#)

Variables store information, like passwords and secret keys, that you can use in job scripts. Each project can define a maximum of 8000 variables. [Learn more.](#)

Variables can have several attributes. [Learn more.](#)

- **Protected:** Only exposed to protected branches or protected tags.
- **Masked:** Hidden in job logs. Must match masking requirements.
- **Expanded:** Variables with `$` will be treated as the start of a reference to another variable.

Environment variables are configured by your administrator to be **protected** by default.

Type	↑ Key	Value	Options	Environments	
Variable	DH_PASSWORD 	***** 	Protected, Expanded	All (default) 	
Variable	DH_USERNAME 	***** 	Protected, Expanded	All (default) 	
Variable	PORT 	***** 	Protected, Expanded	All (default) 	

[Add variable](#)
[Reveal values](#)

# Study Case Explanation

20. Kemudian push ulang menggunakan git

```

• → apitodo git:(master) x git add .
• → apitodo git:(master) x git commit -m "config ci/cd"
[master a41cba6] config ci/cd
 3 files changed, 9 insertions(+), 13 deletions(-)
• → apitodo git:(master) git push origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 541 bytes | 541.00 KiB/s, done.
Total 5 (delta 3), reused 0 (delta 0)
To https://gitlab.com/temmyalexhimself/ci_cd_todoapi.git
 7e7b33b..a41cba6  master -> master

```

# Study Case Explanation

21. Dan pipeline berhasil dijalankan


Temmy Alex > ci\_cd\_todoapi > Pipelines

All 1 Finished Branches Tags

Clear runner caches CI lint Run pipeline

Filter pipelines  Show Pipeline ID ▾

Status	Pipeline	Triggerer	Stages
<p>passed</p> <p>00:01:45</p> <p>1 hour ago</p>	<p>gitlab config</p> <p>#805873467</p> <p>latest</p>	<p>master</p> <p>efb1a95f</p>	<p>Stages</p> <p>✓ ✓</p>

Download icon: 

# CI/CD with GitLab



**Introduction CI/CD**



**Using GitLab CI/CD**

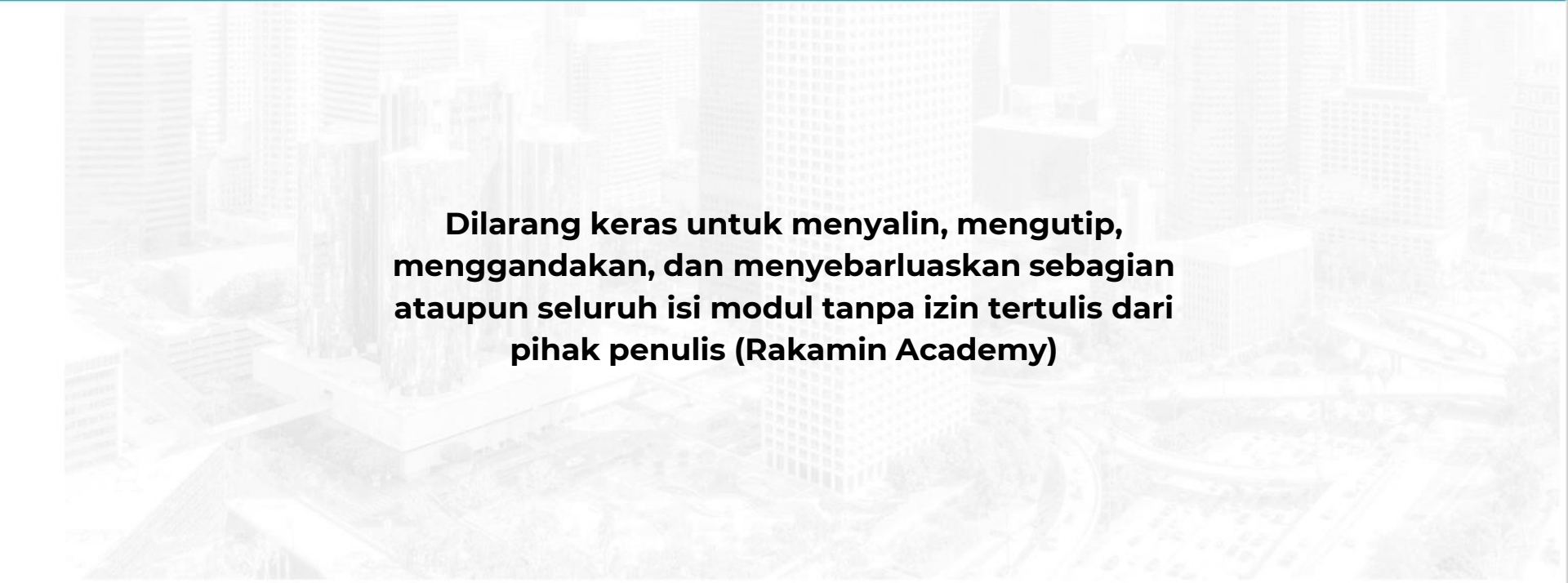


# Reference material

1. [Install Docker](#)
2. [Basic Using Docker](#)
3. [Docker Commands](#)
4. [Docker Manipulation Image](#)
5. [Programmer Zaman Now](#)

**Terima kasih!**

# Copyright Rakamin Academy

A faded, light-colored background image of a city skyline with various skyscrapers and buildings.

**Dilarang keras untuk menyalin, mengutip,  
menggandakan, dan menyebarkan sebagian  
ataupun seluruh isi modul tanpa izin tertulis dari  
pihak penulis (Rakamin Academy)**