

Tugas:

1. Buat 1 aplikasi sederhana
 - Contoh: aplikasi Node.js / Python / Go yang hanya menampilkan **Hello World**.
 - Aplikasi harus bisa dijalankan menggunakan Dockerfile.
 - Push kode aplikasi ke Git repository GitLab
2. Buat satu proses CI/CD di GitLab: build hingga push image. Push image ke **staging** dilakukan otomatis, sedangkan push ke **production** dilakukan secara manual.

1. Buat aplikasi node.js simple

index.js

```
console.log("Hello World");
```

Test : node index.js

2. Buat package.json

Supaya rapi & profesional (CI/CD biasanya expect ini):

npm init -y

Nanti akan muncul kek gini:

```
index.js Dockerfile package.json
{
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1",
    "start": "node index.js"
  }
}
```

Notes : aku tambahin sendiri : "start": "node index.js"

3. Buat Dockerfile

Buat file **Dockerfile**

```
FROM node:20-alpine
```

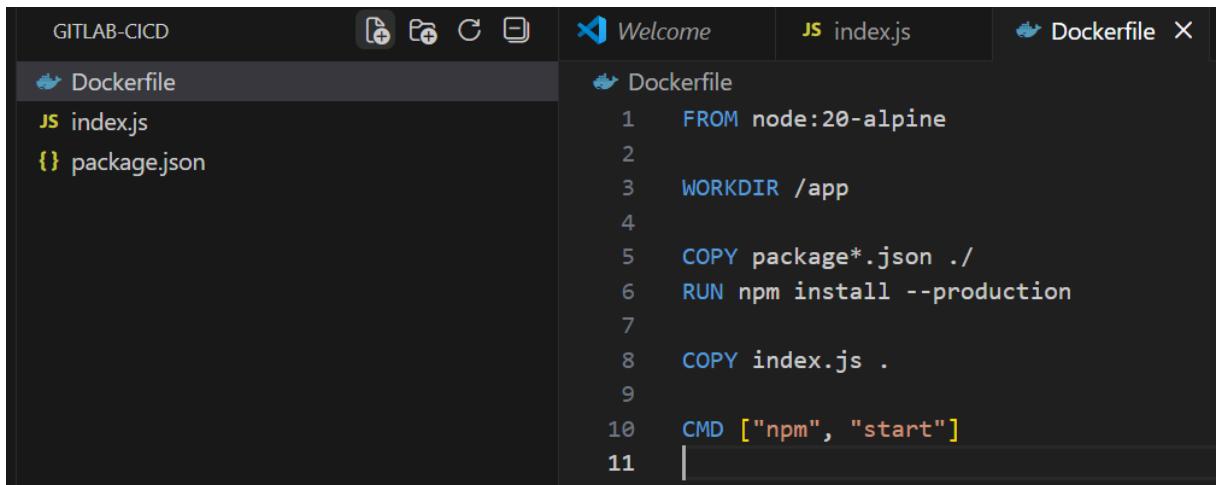
```
WORKDIR /app
```

```
COPY package*.json ./
```

```
RUN npm install --production
```

```
COPY index.js .
```

```
CMD ["npm", "start"]
```



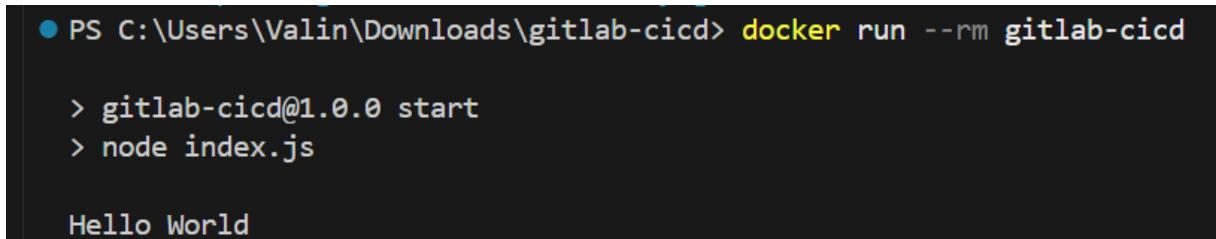
The screenshot shows a code editor interface with three tabs: 'Dockerfile', 'index.js', and 'package.json'. The 'Dockerfile' tab contains the following code:

```
1 FROM node:20-alpine
2
3 WORKDIR /app
4
5 COPY package*.json ./ 
6 RUN npm install --production
7
8 COPY index.js .
9
10 CMD ["npm", "start"]
11 |
```

4. Tes docker local

```
docker build -t gitlab-cicd .
```

```
docker run --rm gitlab-cicd
```



The screenshot shows a terminal window with the following output:

```
● PS C:\Users\Valin\Downloads\gitlab-cicd> docker run --rm gitlab-cicd
> gitlab-cicd@1.0.0 start
> node index.js
Hello World
```

5. Push ke gitlab

Pake SSH atau HTTPS

6. Gitlab CI/CD

Buat file **.gitlab-ci.yml**

stages:

- build

```
- test
- deploy

# --- Build stage ---
build:
  stage: build
  script:
    - echo "Building project..."
  artifacts:
    paths:
      - build/
  when: on_success

# --- Test stage ---
test:
  stage: test
  script:
    - echo "Running tests..."
  when: on_success

# --- Deploy to staging (otomatis) ---
deploy-staging:
  stage: deploy
  script:
    - echo "Deploy to STAGING success"
  environment:
    name: staging
  when: on_success
  only:
    - main
    - develop # Bisa diubah sesuai branch staging kamu

# --- Deploy to production (manual) ---
deploy-production:
  stage: deploy
  script:
    - echo "Deploy to PRODUCTION success"
  environment:
    name: production
  when: manual
  only:
    - main # Hanya muncul di main branch
```

```
allow_failure: false
```

dan... push ke Gitlab

Hasil Akhir

- App Node.js (done)
- Dockerfile (done)
- Image ke GitLab Container Registry (done)
- Staging auto deploy (done)
- Production manual deploy (done)

1. Bukti Image ke GitLab Container Registry (done)

The screenshot shows the GitLab Container registry interface. The left sidebar has a 'Container registry' option selected. The main area displays a single repository named 'gitlab-cicd'. It shows 1 tag. The status bar at the bottom indicates 'Container scanning for registry: Off'.

2. Bukti Staging Auto Deploy

The screenshot shows the GitLab Pipelines interface. The left sidebar has a 'Pipelines' option selected. The main area shows a pipeline with three stages: 'Plan', 'Code', and 'Build', all in the 'Running' status. The pipeline ID is #2290330023. The stages are represented by green checkmarks. The status bar at the bottom indicates 'Running'.

Lihat job:

build-image → ✓

deploy-staging → ✓ (jalan otomatis)

deploy-production → ⚡ (belum jalan)

3. Bukti Production Manual Deploy

The screenshot shows the GitLab Pipelines interface for a project named 'gitlab-cicd'. On the left, there's a sidebar with options like Issues, Merge requests, Manage, Plan, Code, Build, Pipelines, Jobs, Pipeline editor, Pipeline schedules, Artifacts, Secure, Deploy, and Operate. The 'Pipelines' option is selected. In the main area, there's a search bar and a status summary: 'All 3' pipelines, 'Finished' (0), 'Branches' (main), and 'Tags' (latest). Below this is a 'Pipeline' section with a single pipeline card. The pipeline has three stages: 'build', 'test', and 'deploy-production'. The 'deploy-production' stage is currently inactive, as indicated by a tooltip: '#2290330023 update gitlab-ci.yml' and 'main ➔ 4bb735c7 [] latest [] branch []'. There are also buttons for 'Run' and 'Cancel'.

Klik deploy-production :

This screenshot shows the same GitLab Pipelines interface as the previous one, but the 'deploy-production' stage is now active. A large green checkmark icon is displayed above the stage name, and the status message 'Passed' is shown. The pipeline card also includes details about the commit (4bb735c7) and the fact that it was created 6 minutes ago by 'valinfernanda'.