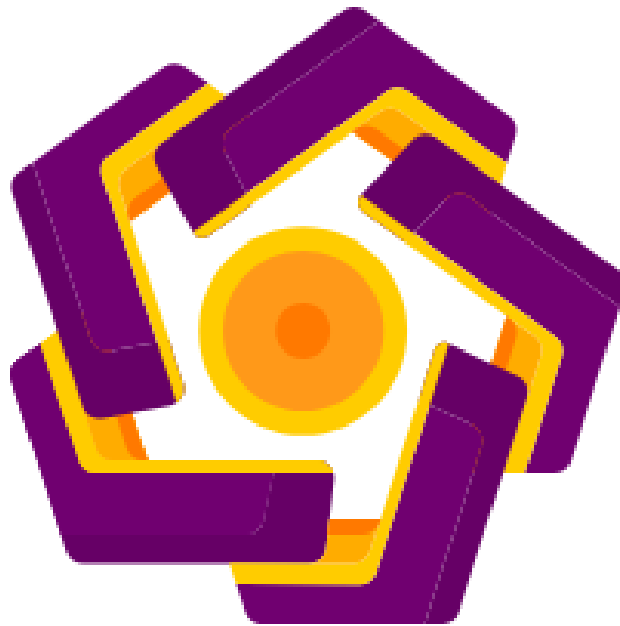


LAPORAN PRAKTIKUM MATA KULIAH DEVELOPMENT OPERATION



Nama : Frumentios David Ivan Satria
NIM : 23.01.5085
Kelas : D3 Teknik Informatika 03
Praktikum ke- : 8

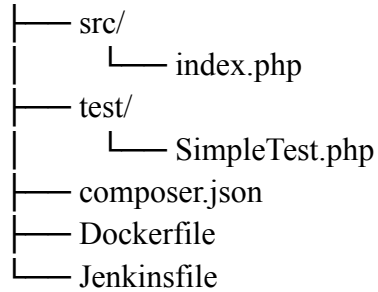
Pengampu :
Hastari Utama, S.Kom., M.Cs.

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
SLEMAN 2025**

LANGKAH-LANGKAH TUGAS

1. Buatlah struktur folder seperti dibawah :

laprak8/



2. Isi *index.php* dengan script dibawah ini :

```
index.php x
src > index.php
1 <?php
2 echo "<h1>Aplikasi PHP Sederhana Berhasil Dijalankan!</h1>";
3 echo "<p>Pipeline CI/CD dengan Jenkins dan Docker sukses.</p>";
4 echo "Hostname: " . gethostname();
```

A screenshot of a code editor showing the content of 'index.php'. The code is a PHP script that outputs an HTML header, a paragraph, and the system hostname.

3. Isi *composer.json* dengan script dibawah ini :

```
Jenkinsfile Dockerfile {} composer.json 1 x
home > david > kuliah > devops > laprak8 > {} composer.json > ...
1
2 "name": "my-user/laprak8",
3 "description": "A simple PHP project for Jenkins CI/CD.",
4 "autoload": {
5     "psr-4": {
6         "App\\": "src/"
7     }
8 },
9 "require-dev": {
10     "phpunit/phpunit": "^9.5"
11 },
12 "scripts": {
13     "test": "vendor/bin/phpunit test"
14 }
15
```

A screenshot of a code editor showing the content of 'composer.json'. The JSON file defines project metadata, PSR-4 autoloading, development dependencies (PHPUnit), and a test script.

4. Isi *SimpleTest.php* dengan script dibawah ini :

```
SimpleTest.php x
test > SimpleTest.php
1 <?php
2 use PHPUnit\Framework\TestCase;
3
4 class SimpleTest extends TestCase
5 {
6     public function testSomethingIsTrue()
7     {
8         $this->assertTrue(true);
9     }
10 }
```

A screenshot of a code editor showing the content of 'SimpleTest.php'. The code is a PHP class that extends PHPUnit's TestCase and contains a single test method 'testSomethingIsTrue'.

5. Isi *Dockerfile* dengan script dibawah ini :

```
Dockerfile x
home > david > kuliah > devops > laprak8 > Dockerfile > ...
1 FROM php:8.1-apache
2 COPY src/ /var/www/html/
3 EXPOSE 80
```

6. Isi *Jenkinsfile* dengan script dibawah ini :

```
Jenkinsfile x Dockerfile {} composer.json 1
home > david > kuliah > devops > laprak8 > Jenkinsfile
1 pipeline {
2     agent any
3     stages {
4         stage('Install Dependencies') {
5             agent {
6                 docker {
7                     image 'composer:2.5'
8                 }
9             }
10            steps {
11                sh 'composer install --no-interaction --no-progress'
12            }
13        }
14        stage('Run Unit Test') {
15            agent {
16                docker {
17                    image 'composer:2.5'
18                }
19            }
20            steps {
21                echo "Memulai diagnostik direktori sebelum menjalankan tes..."
22                sh 'echo "--- LOKASI SAAT INI (PWD) ---"'
23                sh 'pwd'
24                sh 'echo "--- DAFTAR FILE DI WORKSPACE (ls -la) ---"'
25                sh 'ls -la'
26                sh 'echo "--- MENCoba MENJALANKAN TES ---"'
27                sh 'composer test'
28            }
29        }
30        stage('Build and Deploy') {
31            steps {
32                script {
33                    def appImage = docker.build('laprak8-image:latest')
34                    sh 'docker stop laprak8-live-app || true'
35                    sh 'docker rm laprak8-live-app || true'
36                    appImage.run('-d -p 8000:80 --name laprak8-live-app')
37                }
38            }
39        }
40    }
41}
```

7. Push semua script tadi ke github, milik saya dapat dicek di link berikut :

https://github.com/c0llosum/devops_laprak8_5085.git

```
github.com/c0l0sseum/devops_laprak8_5085
Linux Lite Terminal
File Edit View Terminal Tabs Help
Initialized empty Git repository in /home/david/kuliah/devops/laprak8/.git/
david ~ > kuliah > devops > laprak8 > git add .
david ~ > kuliah > devops > laprak8 > git commit -m "LAPORAN PRAKTIKUM DEVOPS 8 (23.01.5085)"
[master (root-commit) 2d6381f] LAPORAN PRAKTIKUM DEVOPS 8 (23.01.5085)
5 files changed, 84 insertions(+)
create mode 100644 Dockerfile
create mode 100644 Jenkinsfile
create mode 100644 composer.json
create mode 100644 src/index.php
create mode 100644 test/SimpleTest.php
david ~ > kuliah > devops > laprak8 > git remote add origin https://github.com/c0l0sseum/devops_laprak8_5085.git
david ~ > kuliah > devops > laprak8 > git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 1.44 KiB | 491.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/c0l0sseum/devops_laprak8_5085.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
david ~ > kuliah > devops > laprak8 >
```

8. Setelah diupload ke Github kita perlu mengganti script Dockerfile pada direktori local kita (bukan di github) menjadi seperti dibawah ini :

```
Jenkinsfile Dockerfile
home > david > kuliah > devops > laprak8 > Dockerfile > ...
1 # Mulai dari image Jenkins resmi yang stabil
2 FROM jenkins/jenkins:its-jdk11
3
4 # Ganti ke user root untuk menginstal paket
5 USER root
6
7 # Instal Docker CLI di dalam image
8 RUN apt-get update && apt-get install -y curl gnupg
9 RUN install -m 0755 -d /etc/apt/keyrings
10 RUN curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --dearmor -o /etc/apt/keyrings/docker.gpg
11 RUN chmod a+r /etc/apt/keyrings/docker.gpg
12 RUN echo \
13     "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/debian \
14     $(grep 'VERSION_CODENAME=' /etc/os-release | cut -d= -f2) stable" | \
15     tee /etc/apt/sources.list.d/docker.list > /dev/null
16 RUN apt-get update && apt-get install -y docker-ce-cli
17
18 # Ganti kembali ke user jenkins
19 USER jenkins
```

9. Kita akan membuat dengan image terlebih dahulu karena device saya harus custom image supaya bisa jalan dengan command *docker build -t my-jenkins* . (catatan: harus masuk cd ke folder yang berisi Dockerfile yang sudah diubah tadi baru jalankan perintahnya)

10. Jalankan script ini pada terminal docker desktop :

```
docker run -d -p 8080:8080 -p 50000:50000 --name jenkins --group-add 0 -v
jenkins_home:/var/jenkins_home -v /var/run/docker.sock:/var/run/docker.sock my-jenkins
```

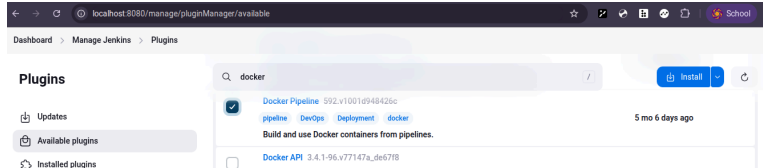
Saya menggunakan jenkins custom saya sendiri.

```
david ~ > kuliah > devops > laprak8 > docker run \
-d \
-p 8080:8080 \
-p 50000:50000 \
--name jenkins \
--group-add 0 \
-v jenkins_home:/var/jenkins_home \
-v /var/run/docker.sock:/var/run/docker.sock \
my-jenkins
9431c17d7481e815f617353962710a3013a029fd624d8d85df2769f2cb3589e1
```

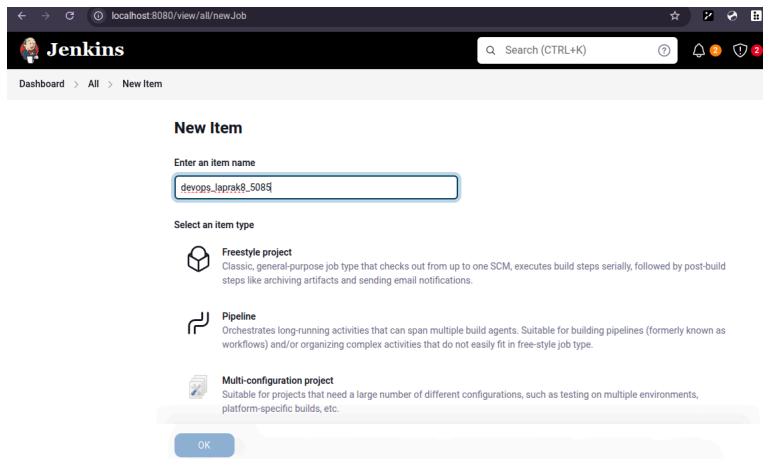
11. Buka <http://localhost:8080/> pada web browser. Jika baru pertama kali menggunakan jenkins dapat mengikuti arahan di jenkins tersebut. Namun jika jenkins sudah terinstall

kita tinggal login.

12. Pertama install Plugin *docker pipeline* pada *Manage Jenkins > Plugins* . Setelah terinstall lebih baik Restart Jenkins dengan checklist pada bagian bawah pada tab yang otomatis terlalihkan di log instalasi.

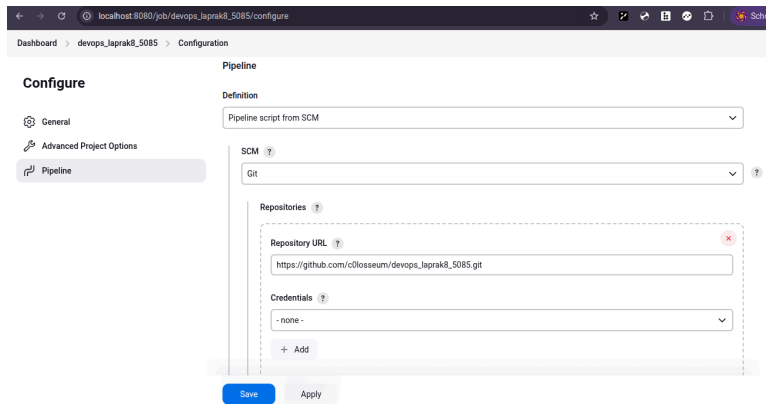


13. Kembali ke Dashboard lalu pergi ke + *New Item* > Isi nama Itemnya > Pilih item type Pipeline > OK

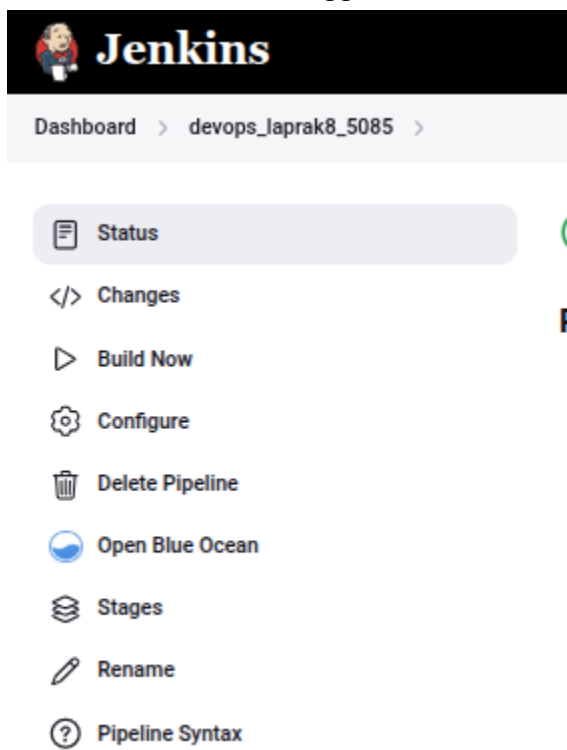


14. Pilih tab Pipeline kemudian pada kolom *Definition* pilih *Pipeline Script from SCM* kemudian isi *Repository URL* dengan github yang sudah diupload tadi, jangan lupa isi brachnya sesuai dengan github tadi, setelah itu *Save*.





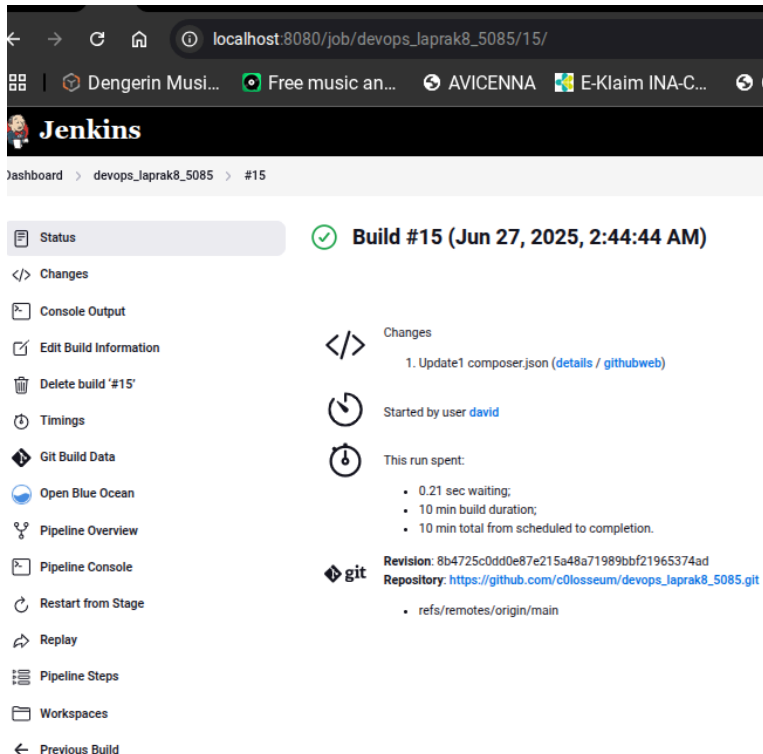
15. Jika sudah dibuat kita tinggal tekan *Build Now*



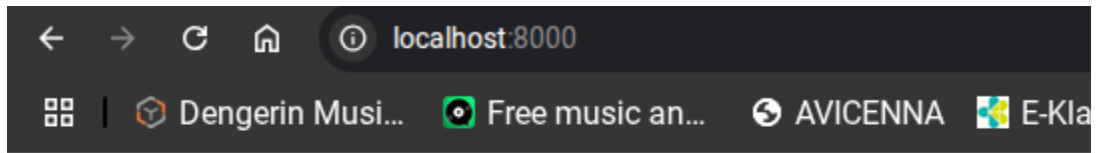
16. Setelah muncul proses Buildnya kita bisa mengeceknya di *Console Output*



17. Setelah percobaan ke 15 saya baru berhasil menampilkan output sederhana tersebut



18. Berikut hasil output pada website <http://localhost:8000/>



Aplikasi PHP Sederhana Berhasil Dijalankan!

Pipeline CI/CD dengan Jenkins dan Docker sukses.

Hostname: 2cb76f2cc4a3