

Jenkins

Modul Praktikum W4 – Integrasi Jenkins

Mata Kuliah: Komputasi Awan

Semester: Ganjil 2025–2026

Dosen Pengampu: M. Nizar P. Ma'ady, S.Kom., M.IM., M.Kom.

Disusun oleh: Asisten Praktikum

Tujuan Pembelajaran


Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu:

1. Mengintegrasikan **Jenkins** dengan **GitHub**.
2. Melakukan instalasi dan konfigurasi **WSL2 (Windows Subsystem for Linux 2)**.
3. Membuat dan mengelola **akun user di Jenkins**.
4. Menggunakan **Ngrok** untuk membuat link publik dari localhost (uji login Jenkins).

Bagian 1 — Instalasi dan Integrasi Jenkins ke GitHub

◆ 1. Persiapan Awal


Sebelum mulai, pastikan:

- Jenkins telah terinstal di komputer/laptop Anda.
- Git telah terinstal. Jika belum, unduh dari:
 <https://git-scm.com/download>
Sesuaikan dengan sistem operasi Anda.

◆ 2. Instalasi dan Konfigurasi Ngrok

Ngrok digunakan untuk membuat **localhost** agar bisa diakses publik oleh GitHub.

Langkah-langkah:

1. Unduh Ngrok di:
 <https://ngrok.com/download>
2. Ekstrak dan install Ngrok sesuai OS Anda.
3. Jalankan Ngrok dari Command Prompt:
4. `ngrok http 8080`

Port 8080 adalah port default Jenkins.

5. Setelah itu, akan muncul link publik seperti contoh:
 6. <https://94f2-36-67-147-34.ngrok-free.app>
-

◆ 3. Menyiapkan GitHub Webhook

Webhook digunakan agar GitHub bisa memicu build otomatis di Jenkins setiap ada perubahan (push/pull request).

Langkah-langkah:

1. Masuk ke repository GitHub Anda → **Settings** → **Webhooks** → **Add Webhook**.
 2. Isi form:
 - **Payload URL** → gunakan link Ngrok diikuti /github-webhook/, contoh:
<https://94f2-36-67-147-34.ngrok-free.app/github-webhook/>
 - **Content type**: application/json
 - Centang event: **Pushes** dan **Pull requests**
 - Klik **Save**
-

◆ 4. Konfigurasi Git di Jenkins

1. Buka Jenkins → **Manage Jenkins** → **Tools** → **Git**
 2. Pada **Git installations**, isi:
 3. Path to Git executable:
 4. C:\Program Files\Git\bin\git.exe
 5. Klik **Save**.
-

◆ 5. Membuat Project Freestyle di Jenkins

1. Klik **New Item** → **Freestyle Project**.
 2. Pada bagian **Source Code Management**, pilih **Git**, lalu isi:
 3. Repository URL: <https://github.com/username/nama-project.git>
 4. Pada bagian **Build Triggers**, centang:
 5. GitHub hook trigger for GITScm polling
 6. Klik **Save** dan **Build Now** untuk menjalankan.
-

◆ Tujuan

WSL2 memungkinkan Anda menjalankan sistem operasi Linux langsung di Windows tanpa perlu dual-boot atau virtual machine.

Jenkins dan Docker sering berjalan lebih baik dengan WSL2.

◆ Langkah Instalasi WSL2

1. **Unduh Docker Desktop:**

👉 <https://www.docker.com/products/docker-desktop>

2. **Aktifkan Fitur WSL:**

- Buka menu Turn Windows features on or off
- Centang **Windows Subsystem for Linux**

3. **Update WSL melalui Command Prompt (jalankan sebagai Administrator):**

4. `wsl --update`

5. **Cek versi WSL:**

6. `wsl -l -v`

7. **Restart komputer Anda.**

8. Pastikan muncul **Version 2** pada distro Linux yang aktif.

Contoh hasil:

9. NAME STATE VERSION

10. Ubuntu Running 2

11. **Buka Docker Desktop**, ikuti panduan setup, dan login ke akun Docker Anda.

12. Instalasi berhasil jika Docker berhasil login dan menunjukkan status **Running**.

Bagian 3 — Membuat Akun User di Jenkins

◆ Tujuan

Untuk memberi akses bagi mahasiswa atau anggota tim agar dapat login ke Jenkins sesuai hak akses masing-masing.

Langkah-langkah:

1. Masuk ke Jenkins sebagai admin.
2. Pilih **Manage Jenkins → Security → Manage Users**.
3. Klik **Create User**.
4. Isi data berikut:

- Username
 - Password
 - Full Name
 - Email Address
5. Klik **Create User**.
 6. Pastikan user muncul di daftar pengguna Jenkins.

💡 *Akun ini nantinya bisa digunakan mahasiswa untuk login via link publik (Ngrok).*

Bagian 4 — Membuat Link Publik Jenkins via Ngrok

Setelah Jenkins berjalan di localhost (biasanya `http://localhost:8080`):

1. Jalankan perintah di Command Prompt:
2. `ngrok http 8080`
3. Salin link publik yang diberikan oleh Ngrok, misalnya:
4. `https://abcd-1234-xyz.ngrok-free.app`
5. Akses Jenkins dari browser menggunakan link tersebut:
6. `https://abcd-1234-xyz.ngrok-free.app`
7. Coba login menggunakan akun user yang telah dibuat.
Jika berhasil, maka integrasi publik Jenkins melalui Ngrok telah berfungsi.

Bagian 5 — Integrasi Jenkins Pipeline ke GitHub

◆ 1. Instal Plugin Pipeline

1. Buka **Manage Jenkins → Plugins → Installed**
Cari **Pipeline**
2. Jika belum ada, buka tab **Available → cari: Pipeline → Install**.

◆ 2. Membuat Pipeline Sederhana

1. Klik **New Item → Pipeline**
2. Isi nama, lalu pilih **Pipeline → OK**
3. Pada bagian **Pipeline Script**, tuliskan:

```
pipeline {  
  agent any  
  stages {
```

```
stage('Hello') {  
    steps {  
        echo 'My Jenkins Pipeline'  
    }  
}  
}
```

4. Klik **Apply → Save → Build Now**

Jika berhasil, akan muncul log **“My Jenkins Pipeline”** di console output.

◆ **3. Integrasi Pipeline dengan GitHub (Jenkinsfile)**

1. Pastikan plugin **Pipeline Stage View** sudah terinstal.
 2. Ubah **Definition** pipeline menjadi:
 3. Pipeline script from SCM
 4. Pilih **SCM: Git**
Lalu isi **Repository URL** dengan repository GitHub Anda.
 5. Di repository GitHub, buat file baru bernama **Jenkinsfile**, lalu isi dengan kode pipeline di atas.
 6. Klik **Apply → Save → Build Now**
Jenkins akan menjalankan pipeline berdasarkan Jenkinsfile dari GitHub.
-

Latihan Mahasiswa

1. Integrasikan Jenkins Anda dengan repository GitHub pribadi.
 2. Buat satu akun user baru dan coba login melalui link publik Ngrok.
 3. Buat pipeline sederhana (Hello World).
 4. Dokumentasikan hasil screenshot integrasi dan login Jenkins.
-

Tugas Minggu Depan

Selesaikan konfigurasi Jenkins yang belum selesai (terutama integrasi GitHub dan pipeline). Pastikan link publik dari Ngrok berfungsi dan user dapat login.