Modul 0: Persiapan Lingkungan Pengembangan (Runtime, IDE, & Framework)

Mata Kuliah: Rekayasa Perangkat Lunak Lintas Platform / Integrasi Sistem

Tujuan Praktikum:

1. Mahasiswa mampu melakukan instalasi dan konfigurasi *Runtime* (Node.js) sebagai fondasi aplikasi web dan backend.
2. Mahasiswa mampu mengonfigurasi Integrated Development Environment (IDE) VS Code dengan ekstensi yang mendukung produktivitas lintas platform.
3. Mahasiswa mampu menyiapkan SDK (Flutter) dan CLI (NestJS) yang akan digunakan selama satu semester.
4. Mahasiswa mampu memverifikasi kesiapan sistem melalui terminal/shell.

1. Pendahuluan

Sebelum memulai pengembangan aplikasi lintas platform, perangkat keras harus disiapkan dengan perangkat lunak pendukung yang tepat. Karena mata kuliah ini menargetkan Web, Mobile, dan Backend, maka kita membutuhkan lingkungan yang mampu melakukan kompilasi dan eksekusi kode secara efisien. Modul ini mencakup langkah-langkah untuk dua sistem operasi utama yang umum digunakan di laboratorium: **Windows** dan **Linux (Ubuntu/Debian-based)**.

2. Implementasi Langkah-Langkah Instalasi

A. Runtime: Node.js (LTS Version)

Node.js diperlukan untuk menjalankan NestJS (Backend) dan React (Web).

- **Windows:** Unduh *Windows Installer (.msi)* dari nodejs.org (Pilih versi LTS). Jalankan dan ikuti instruksi hingga selesai.
- **Linux:** Jalankan perintah berikut di terminal:
  sudo apt update

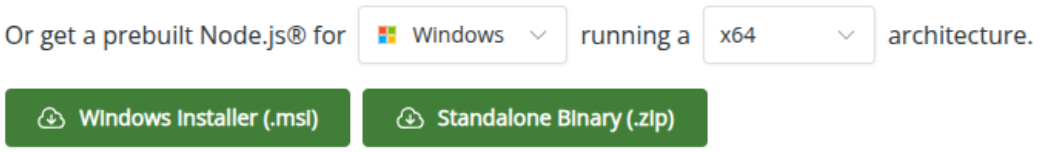
  sudo apt install nodejs npm

Instalasi Node.js di Windows

Berikut adalah panduan praktikum langkah demi langkah instalasi Node.js di Windows:

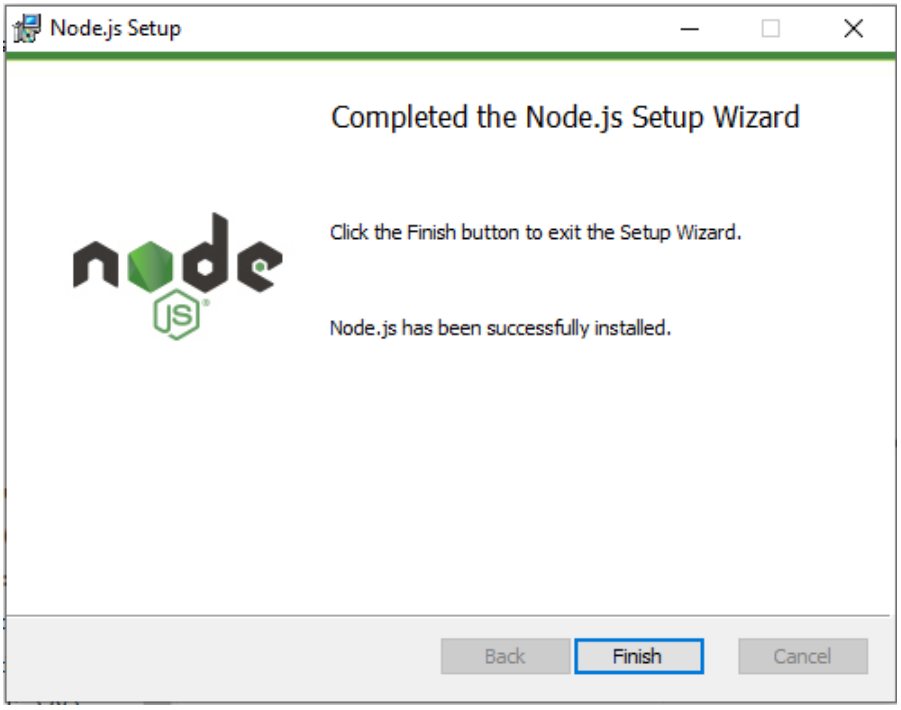
1. Persiapan dan Unduh

- Buka browser dan kunjungi laman resmi Node.js. (https://nodejs.org/en/download)
- Pilih versi prebuilt Node.js atau versi LTS (Long Term Support) untuk stabilitas yang lebih baik, atau Current untuk fitur terbaru.
- Klik tombol unduh untuk Windows (file `.msi`).



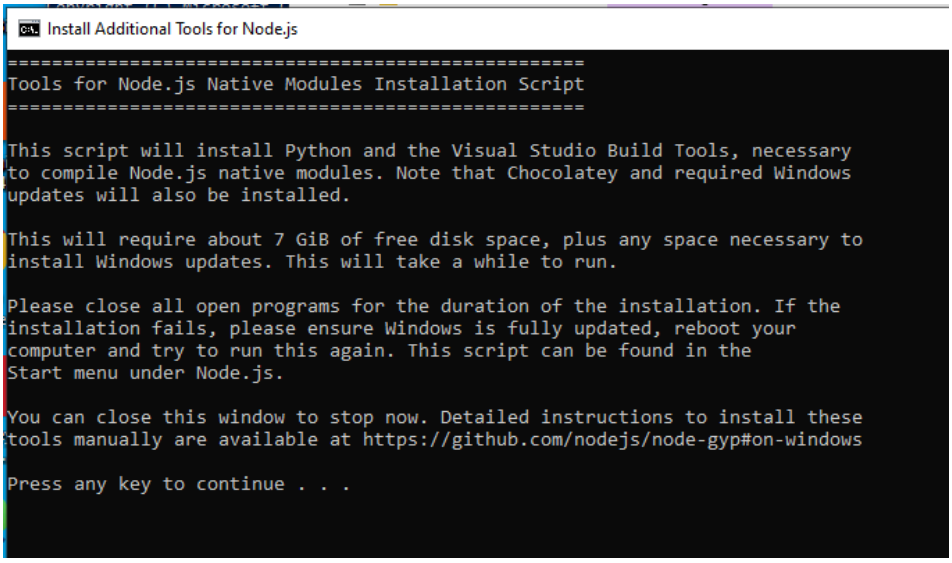
2. Proses Instalasi

1. Jalankan file `.msi` yang telah diunduh.
2. Pada jendela instalasi, klik Next.
3. Centang kotak "I accept the terms in the License Agreement", lalu klik Next.
4. Tentukan folder instalasi (biarkan default/default), lalu klik Next.
5. Pada bagian *Custom Setup*, biarkan pengaturan default (fitur-fitur utama dipilih) dan klik Next.
6. Pada tahap *Tools for Native Modules*, Anda bisa mencentang "Automatically install the necessary tools" (menginstal Chocolatey/Python/Visual Studio) jika diperlukan. Jika tidak yakin, lewatkan dan klik Next.
7. Klik Install untuk memulai, lalu klik Finish setelah selesai.



3. Instalasi Tools Tambahan (Otomatis)

1. Setelah instalasi Node.js selesai dan Anda menekan *Finish*, sebuah jendela PowerShell (biru) akan muncul secara otomatis.
2. Tekan tombol apa saja (Enter) untuk melanjutkan proses instalasi Python dan Visual Studio Build Tools.
3. Jendela ini akan mengunduh dan menginstal komponen tambahan. Proses ini memakan waktu dan kuota internet yang cukup besar.
4. Tutup jendela PowerShell setelah proses selesai (biasanya ada keterangan *Press any key to exit*).



4. Verifikasi Instalasi

- Buka Command Prompt (CMD) atau PowerShell dengan menekan `Win + R`, ketik `cmd`, lalu Enter.
- Periksa versi Node.js dengan perintah:

```
node -v
```

- Periksa versi npm (Node Package Manager) dengan perintah:

```
npm -v
```

- Jika kedua perintah tersebut menampilkan angka versi, Node.js berhasil diinstal.

4. Uji Coba: Membuat Skrip "Hello World" (Opsional)

1. Buat file baru bernama `app.js` menggunakan notepad atau kode editor.
2. Isi dengan kode: `console.log("Hello World dari Node.js!");`.
3. Buka CMD, arahkan ke folder tempat file berada, lalu jalankan:

```
node app.js
```

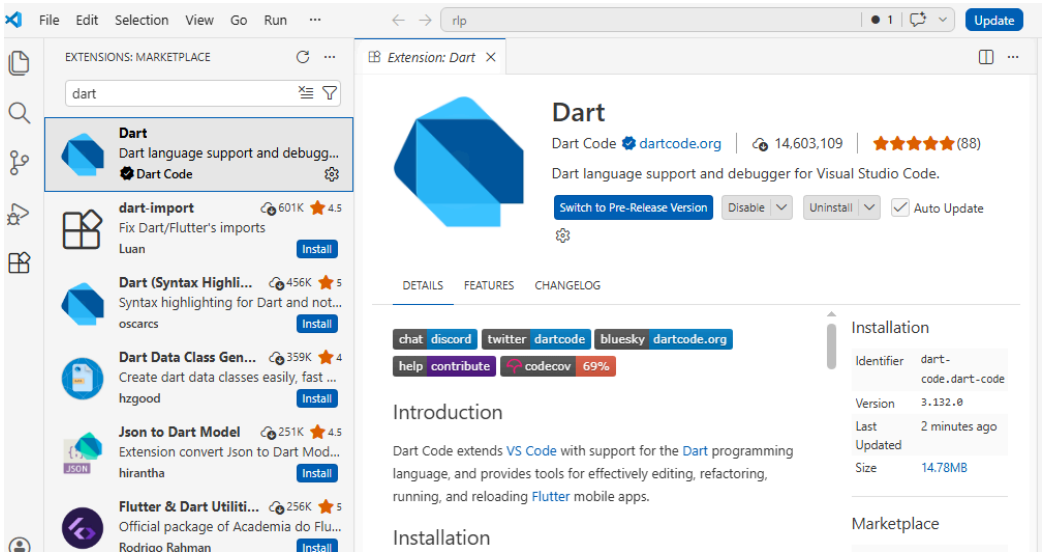
4. Terminal akan menampilkan hasil "Hello World dari Node.js!"

```
C:\Users\vboxuser\Documents\rlp>node app.js
Hello World dari Node.js!
```

B. IDE: Visual Studio Code & Ekstensi

IDE utama yang digunakan adalah VS Code karena ringan dan memiliki dukungan komunitas yang luas.

- **Instalasi:** Unduh dari code.visualstudio.com.
- **Ekstensi Wajib:** Buka menu Extensions (Ctrl+Shift+X) dan instal:
    1. **ESLint & Prettier:** Untuk kerapian kode.
    2. **Tailwind CSS IntelliSense:** Untuk styling web.
    3. **Flutter & Dart:** Untuk pengembangan mobile.
    4. **Thunder Client / Postman:** Untuk pengujian API di dalam IDE.



Langkah-langkah Praktikum Instalasi Ekstensi VS Code (Windows)

1. Buka VS Code: Jalankan aplikasi Visual Studio Code di Windows Anda.
2. Buka Menu Ekstensi:
    a. Klik ikon Extensions berbentuk empat kotak di sidebar kiri (Activity Bar).
    b. *Atau* gunakan shortcut keyboard: tekan tombol `Ctrl+Shift+X` secara bersamaan.
3. Cari Ekstensi:
    a. Ketik nama ekstensi (misal: "Python", "Prettier", "Live Server") pada kotak pencarian di bagian atas panel kiri.
4. Pilih dan Instal:
    a. Pilih ekstensi yang sesuai dari daftar hasil pencarian.
    b. Klik tombol Install berwarna hijau pada ekstensi tersebut.
5. Verifikasi:
    a. Tunggu proses instalasi selesai (tombol install berubah menjadi ikon gir/settings).
    b. Beberapa ekstensi mungkin memerlukan *restart* atau *reload* VS Code, yang akan ditunjukkan dengan tombol Reload Required.

C. Version Control: Git

Git digunakan untuk mengelola kode dan pengerjaan tugas di GitHub (Modul 9).

- **Windows:** Unduh di git-scm.com.
- **Linux:** `sudo apt install git`.

Instalasi & Konfigurasi Git di Windows

1. Unduh (Download) Git

1. Buka browser dan kunjungi situs resmi Git: git-scm.com.
2. Pilih opsi "64-bit Git for Windows Setup" (atau 32-bit sesuai OS Anda) untuk mengunduh installer.

2. Proses Instalasi

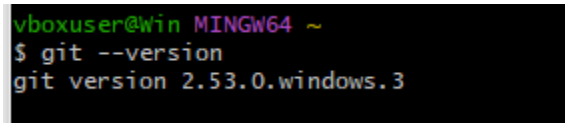
1. Jalankan file installer (`.exe`) yang telah diunduh.
2. Klik Next pada halaman lisensi.
3. Tentukan lokasi instalasi (default: `C:\Program Files\Git`), klik Next.
4. Pada komponen pemilihan, biarkan default, klik Next.
5. Untuk opsi editor, Anda bisa memilih Notepad++ atau editor lain (default: Vim), klik Next.
6. Pilih opsi default (Recommended) untuk penyesuaian nama branch utama menjadi "main" atau "master".
7. Klik Next terus hingga tombol Install muncul, lalu klik Install.
8. Tunggu proses selesai, klik Finish.

3. Verifikasi Instalasi

1. Buka *Command Prompt* (CMD) dengan menekan `Win + R`, ketik `cmd`, lalu tekan Enter.
2. Ketik perintah berikut untuk memastikan Git terinstal:

```
git --version
```

3. Jika berhasil, versi Git (misal: `git version 2.x.x.windows.x`) akan muncul.



4. Konfigurasi Awal Git

Setelah terinstal, Anda wajib mengatur nama dan email agar commit Git tercatat. Buka CMD/Git Bash dan jalankan:

1. Mengatur Username:

```
git config --global user.name "Nama Anda"
```

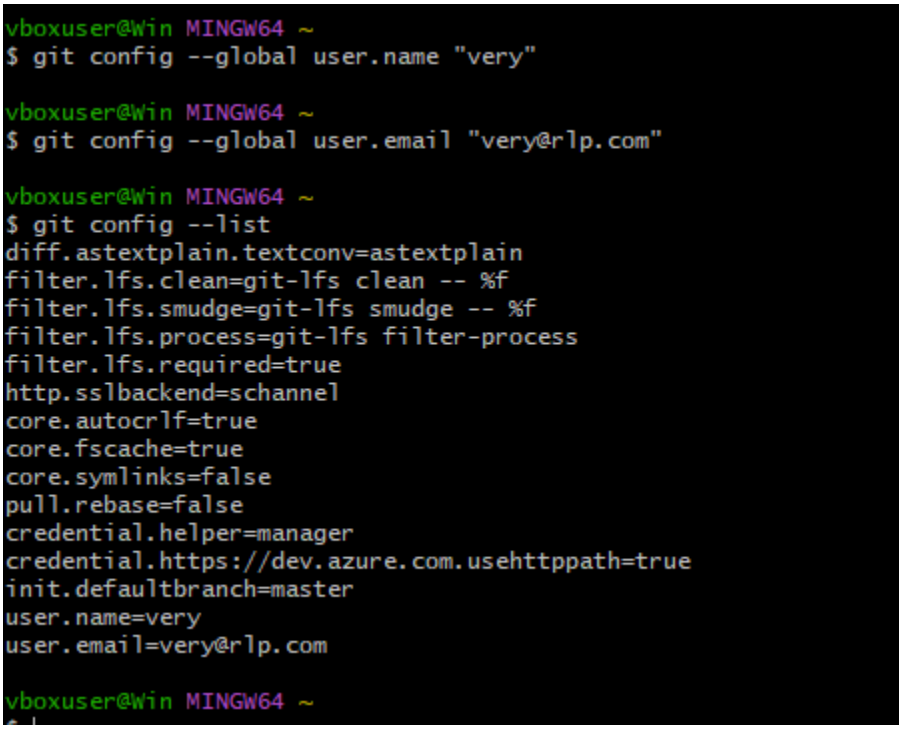
2. Mengatur Email:

```
git config --global user.email "emailanda@example.com"
```

3. Verifikasi Konfigurasi:

```
git config --list
```

Konfigurasi ini penting untuk mengidentifikasi siapa yang membuat perubahan dalam proyek.



D. Framework Backend: NestJS CLI

NestJS akan digunakan mulai Modul 2. Instal secara global menggunakan NPM.

npm install -g @nestjs/cli

Instruksi Praktikum Instalasi NestJS CLI

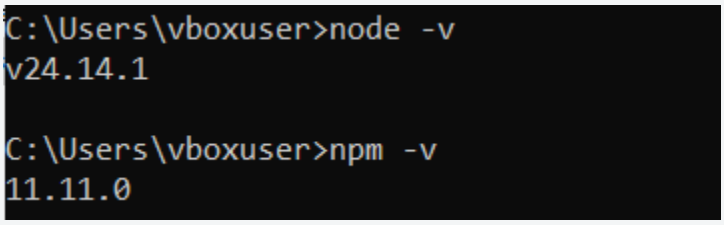
1. Persiapan Prasyarat (Node.js)

- Unduh dan instal Node.js versi LTS dari situs resmi nodejs.org.

- Buka Terminal (CMD/PowerShell) dan verifikasi instalasi dengan perintah:

```
node -v
npm -v
```

- Pastikan kedua perintah tersebut menampilkan versi.

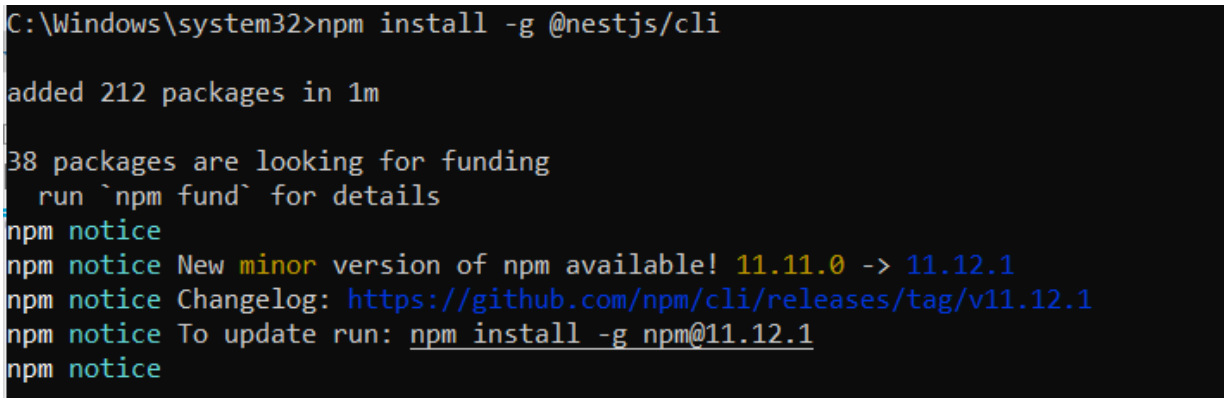


2. Instalasi NestJS CLI Secara Global

- Buka Command Prompt atau PowerShell sebagai Administrator untuk menghindari error izin akses.
- Jalankan perintah berikut:

```
npm install -g @nestjs/cli
```

- Tunggu hingga proses instalasi selesai.



3. Verifikasi Instalasi CLI

- Pastikan CLI terinstal dengan memeriksa versinya:

```
nest -v
```

- Jika muncul nomor versi, berarti NestJS CLI berhasil terinstal.

```
C:\Windows\system32>nest -v
11.0.21
```

E. Mobile SDK: Flutter

Instalasi Flutter membutuhkan perhatian ekstra pada variabel lingkungan (*Environment Variables*).

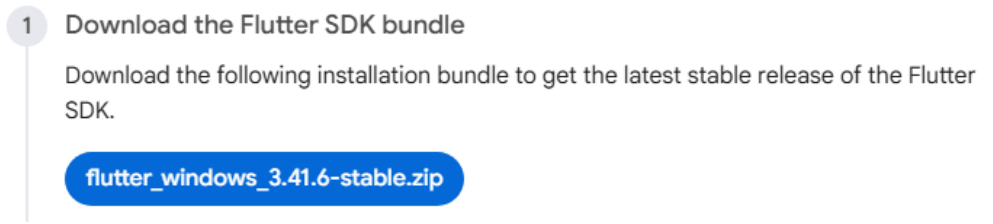
1. Unduh SDK Flutter di docs.flutter.dev.
2. Ekstrak file ke direktori yang aman (Contoh: `C:\src\flutter` atau `/home/user/development/flutter`).
3. **Path Configuration:**
   - **Windows:** Tambahkan path folder `bin` flutter ke *System Environment Variables*.
   - **Linux:** Tambahkan `export PATH="$PATH:[PATH_KE_FLUTTER]/bin"` ke file `.bashrc` atau `.zshrc`.
4. Jalankan `flutter doctor` untuk melihat komponen yang kurang (seperti Android SDK).

Persiapan Perangkat (Prasyarat)

- Windows 7 SP1 atau lebih baru (64-bit).
- Windows PowerShell 5.0 atau yang lebih baru.
- Git for Windows (instal untuk manajemen versi).
- Android Studio (untuk Android SDK dan emulator).

1. Instalasi Flutter SDK (Manually)

1. Download Flutter SDK: Unduh file `.zip` Flutter SDK versi *stable* terbaru dari halaman resmi Flutter. (https://docs.flutter.dev/get-started/install)



2. Ekstrak File: Ekstrak file zip ke lokasi permanen, disarankan bukan di `C:\Program Files\`. *Rekomendasi:* `C:\src\flutter` atau `C:\dev\flutter`.
3. Update Path (Environment Variables):
    1. Cari "env" atau "edit the system environment variables" di search bar Windows.
    2. Klik Environment Variables.
    3. Di bagian User variables, cari `Path` dan klik Edit.
    4. Klik New dan tambahkan path lengkap ke folder `bin` hasil ekstrak (contoh: `C:\src\flutter\bin`).
    5. Klik OK/Apply di semua jendela.

3. Konfigurasi Android dan Validasi

1. Instal Android Studio: Instal Android Studio, buka SDK Manager, dan pastikan Android SDK Command-line Tools terinstal.
2. Jalankan Flutter Doctor: Buka Command Prompt atau PowerShell, lalu jalankan perintah:

```
flutter doctor
```

3. Perbaiki Masalah: Ikuti petunjuk `flutter doctor` jika ada komponen yang kurang (terutama lisensi Android).

4. Setup IDE (VS Code)

1. Buka VS Code.
2. Masuk ke menu Extensions (`Ctrl+Shift+X`).
3. Cari dan instal ekstensi Flutter.

5. Membuat Proyek Pertama

1. Buka VS Code, tekan `Ctrl+Shift+P`.
2. Ketik `Flutter: New Project`.
3. Pilih `Application`, beri nama proyek, dan simpan.

3. Verifikasi Kesiapan (Checklist)

Mahasiswa wajib melakukan pengecekan melalui Terminal/Command Prompt untuk memastikan semua *tools* dapat dipanggil dari mana saja.

Komponen	**Perintah Verifikasi**	**Output yang Diharapkan**
Node.js	`node -v`	v18.x.x atau v20.x.x
NPM	`npm -v`	9.x.x atau lebih tinggi
Git	`git --version`	git version 2.x.x
Nest CLI	`nest -v`	10.x.x atau lebih tinggi

Flutter	`flutter --version`	Flutter 3.x.x

4. Penjelasan Logika Persiapan

- **LTS (Long Term Support):** Kita menggunakan versi LTS agar stabilitas aplikasi terjaga selama satu semester dan meminimalkan *bug* versi eksperimental.
- **Global CLI:** Menginstal NestJS dengan flag `-g` memungkinkan kita memanggil perintah `nest` di folder mana pun tanpa harus menginstalnya berulang kali.
- **Flutter Doctor:** Ini adalah fitur unik Flutter yang secara cerdas mendiagnosis masalah pada mesin lokal (seperti lisensi Android yang belum disetujui atau SDK yang hilang).

5. Tugas Praktikum

1. Lakukan instalasi seluruh perangkat lunak di atas pada komputer laboratorium atau laptop pribadi.
2. Jalankan perintah `flutter doctor` dan pastikan tidak ada tanda silang merah pada bagian inti (Flutter & Toolchain).
3. **Kumpulkan:** Tangkapan layar (*screenshot*) terminal yang menunjukkan hasil dari perintah verifikasi (Node, Git, Nest, dan Flutter) dalam satu file PDF.

Hasil Praktikum: Laporan kesiapan lingkungan pengembangan yang menunjukkan status "Ready" untuk memulai Modul 1.