



# Phase 1

## Roadmap Tahap 1 (12 Minggu)

### Bulan 1 → Dasar Backend & Database

#### Minggu 1: HTTP & Client-Server

1. Pahami konsep client-server, request-response cycle.
2. Belajar HTTP methods (GET, POST, PUT, DELETE), status codes (200, 400, 401, 404, 500).
3. Coba praktek: pakai `curl` atau Postman untuk kirim request ke public API.

#### Minggu 2: REST API Konsep

1. Apa itu REST & prinsipnya (stateless, resource-based, URI design).
2. Belajar struktur JSON request/response.
3. Praktek: bikin REST API sederhana (misal CRUD todo list) dengan **Express.js (Node.js)** atau **Django/Flask (Python)** atau **Spring Boot (Java)**.

#### Minggu 3: SQL Dasar (PostgreSQL/MySQL)

1. Instal database lokal (PostgreSQL lebih disarankan).
2. Belajar SQL dasar: `CREATE TABLE`, `INSERT`, `SELECT`, `UPDATE`, `DELETE`.
3. Pahami tipe data (INT, VARCHAR, DATE, JSON).
4. Latihan bikin database sederhana (contoh: database mahasiswa).

#### Minggu 4: SQL Lanjutan

1. Belajar **JOIN** (INNER, LEFT, RIGHT).
2. Pahami **transaction** ( `BEGIN`, `COMMIT`, `ROLLBACK` ).
3. Indexing (kenapa perlu, cara buat).

4. Praktek query untuk simulasi data real (contoh: join mahasiswa dengan mata kuliah).
- 

## Bulan 2 → Linux, Git, dan Integrasi Database

### Minggu 5: Dasar Linux

1. Command dasar: `ls`, `cd`, `pwd`, `cat`, `grep`, `find`.
2. Permission & process management ( `chmod`, `ps`, `kill` ).
3. Coba jalankan PostgreSQL/MySQL lewat terminal.

### Minggu 6: Git Workflow

1. Git dasar: `init`, `clone`, `add`, `commit`, `push`.
2. Branching ( `git branch`, `git checkout -b`, `merge` ).
3. GitHub/GitLab workflow: PR (Pull Request).
4. Latihan dengan project mini bareng repo GitHub.

### Minggu 7: Koneksi API ↔ Database

1. Pelajari cara koneksi backend framework ke PostgreSQL/MySQL.
2. Implementasi CRUD dengan database.
3. Strukturisasi project (MVC atau layered architecture).
4. Coba validasi data sederhana (contoh: tidak boleh insert buku tanpa judul).

### Minggu 8: Authentication Dasar

1. Konsep authentication vs authorization.
  2. Implement login/registrasi dengan **hash password (bcrypt/argon2)**.
  3. Simpan session/token di database.
  4. Coba pakai **JWT (JSON Web Token)** untuk autentikasi.
- 

## Bulan 3 → Project "Library Management System"

### Minggu 9: Setup Project

1. Buat repo GitHub.

2. Setup backend framework + database.
3. Desain ERD (Entity Relationship Diagram) untuk tabel:
  - User
  - Book
  - Borrowing (peminjaman)
4. Buat migration/DDL SQL.

### **Minggu 10: Implementasi CRUD**

1. CRUD Book (add, list, update, delete).
2. CRUD User (register, update profile).
3. CRUD Borrowing (pinjam, kembalikan).
4. Test API dengan Postman.

### **Minggu 11: Authentication & Middleware**

1. Tambahkan endpoint login/registrasi.
2. Implementasi JWT middleware untuk proteksi API.
3. Hanya user login yang bisa pinjam buku.
4. Tambahkan role (admin/user).

### **Minggu 12: Finishing + Deployment**

1. Testing seluruh endpoint (Postman collection).
  2. Dokumentasi API (pakai Swagger/OpenAPI atau markdown sederhana).
  3. Deploy ke cloud gratis (misal **Render**, **Railway**, atau **Heroku**).
  4. Presentasi mini project "Library Management System".
- 

## **Ringkasan**

- **Bulan 1:** HTTP, REST API, SQL dasar.
- **Bulan 2:** Linux, Git workflow, integrasi API dengan DB, auth dasar.
- **Bulan 3:** Bangun full project (CRUD + Auth + Deployment).

---