

# **LAPORAN TUGAS UJIAN AKHIR SEMESTER (UAS)**

## **“Aplikasi Studylink”**

**Dosen pengampu :**

**Yogi Yulianto, M.Kom**



**Disusun Oleh :**

**AGNES DWETASARI**

**TI.24.CA.1 / 312410092**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**PEMROGRAMAN MOBILE 1**

**UNIVERSITAS PELITA BANGSA**

**2025**

# **PENDAHULUAN**

## **Latar Belakang**

Perkembangan teknologi informasi khususnya di bidang mobile application telah membawa perubahan besar dalam dunia pendidikan. Aktivitas akademik seperti pengelolaan jadwal kuliah, tugas, ujian, forum diskusi, hingga manajemen profil mahasiswa kini dituntut untuk lebih terintegrasi dan mudah diakses. Oleh karena itu, dibutuhkan sebuah aplikasi yang mampu membantu mahasiswa dalam mengatur seluruh aktivitas akademiknya secara terpusat.

Aplikasi StudyLink dikembangkan sebagai solusi untuk membantu mahasiswa dalam mengelola kegiatan perkuliahan secara digital. Aplikasi ini berbasis Android dengan dukungan backend API menggunakan Node.js dan Express.js, serta penyimpanan data menggunakan file JSON. StudyLink dirancang agar mudah digunakan, memiliki fitur yang lengkap, serta mendukung pembaruan data secara real-time melalui API.

## **Rumusan Masalah**

Berdasarkan latar belakang di atas, rumusan masalah dalam pengembangan aplikasi StudyLink adalah sebagai berikut:

1. Bagaimana merancang aplikasi Android yang dapat mengelola aktivitas akademik mahasiswa?
2. Bagaimana mengintegrasikan aplikasi Android dengan backend API?
3. Bagaimana mengimplementasikan fitur CRUD (Create, Read, Update, Delete) pada aplikasi?
4. Bagaimana menerapkan sistem autentikasi dan manajemen profil pengguna?

## **Tujuan**

Tujuan dari pembuatan aplikasi StudyLink adalah:

1. Mengembangkan aplikasi Android untuk manajemen aktivitas akademik.
2. Mengimplementasikan REST API sebagai backend aplikasi.
3. Menerapkan fitur CRUD pada berbagai modul akademik.
4. Membuat sistem login, registrasi, dan pengelolaan profil pengguna.

## **Manfaat**

Manfaat dari aplikasi StudyLink antara lain:

- Membantu mahasiswa mengatur jadwal, tugas, dan ujian.
- Mempermudah penyimpanan dan pengelolaan data akademik.
- Menjadi media pembelajaran penerapan Android dan REST API.

# **LANDASAN TEORI**

## **Android**

Android adalah sistem operasi berbasis Linux yang digunakan pada perangkat mobile. Android menyediakan Software Development Kit (SDK) yang memungkinkan pengembang membangun aplikasi menggunakan bahasa pemrograman Java.

## **Rest API**

REST API (Representational State Transfer Application Programming Interface) adalah arsitektur komunikasi antara client dan server menggunakan protokol HTTP dengan metode GET, POST, PUT, dan DELETE.

## **Node.js dan Express.js**

Node.js merupakan platform JavaScript yang berjalan di sisi server, sedangkan Express.js adalah framework Node.js yang memudahkan pembuatan REST API.

## **JSON**

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan dan mudah dibaca, digunakan sebagai media penyimpanan data dan komunikasi API.

## **Retrofit**

Retrofit adalah library Android yang digunakan untuk melakukan komunikasi HTTP dengan REST API secara mudah dan terstruktur.

# **ANALISIS DAN PERANCANGAN SISTEM**

## **Analisis Kebutuhan Sistem**

### **Kebutuhan Fungsional**

Aplikasi StudyLink memiliki fitur-fitur berikut:

1. Registrasi dan Login pengguna
2. Manajemen profil pengguna (edit data dan upload foto)
3. Manajemen jadwal (Schedule)
4. Manajemen tugas (Assignment)
5. Manajemen ujian (Exam)
6. Forum diskusi
7. Kalender akademik
8. Fitur pencarian data

### **Kebutuhan Non-Fungsional**

- Aplikasi berjalan pada perangkat Android
- Backend berjalan pada server lokal menggunakan Node.js
- Data disimpan dalam file JSON
- Antarmuka pengguna sederhana dan mudah digunakan

## **Arsitektur Sistem**

Aplikasi StudyLink menggunakan arsitektur Client-Server, di mana:

- Client: Aplikasi Android
- Server: Node.js + Express.js
- Database: File db.json

## **Perancangan API**

Contoh endpoint API yang digunakan:

- POST /login
- POST /register
- GET /users/:id
- PUT /users/:id/profile
- POST /users/:id/photo
- CRUD assignments, schedules, exams, forum, dan calendar

## **Diagram Alur Sistem (Flowchart Singkat)**

1. User login
2. Sistem memverifikasi data ke API
3. Jika berhasil → Dashboard
4. User memilih menu (Schedule, Assignment, Course, Exam, Forum, Kalender)
5. Data ditampilkan dari server

# **IMPLEMENTASI SISTEM**

## **Implementasi Aplikasi Android**

Aplikasi Android dikembangkan menggunakan bahasa Java dengan Android Studio. Library Retrofit digunakan untuk komunikasi API, sedangkan Glide digunakan untuk menampilkan gambar profil pengguna.

## **Implementasi Backend**

Backend dikembangkan menggunakan Node.js dan Express.js. Data disimpan dalam file db.json dan dikelola menggunakan operasi CRUD.

## **Manajemen Profil Pengguna**

Pengguna dapat:

- Melihat profil
- Mengedit nama depan, nama belakang, dan email
- Mengunggah foto profil

Perubahan data akan langsung tersimpan di server dan diperbarui di aplikasi.

## **Keamanan**

- Password dienkripsi menggunakan bcrypt
- Autentikasi login menggunakan JSON Web Token (JWT)

# PENGUJIAN SISTEM

## Pengujian Fungsional

Pengujian dilakukan pada setiap fitur:

- Login dan Register berhasil
- CRUD data berjalan dengan baik
- Update profil dan upload foto berhasil

## Hasil Pengujian

Hasil pengujian menunjukkan bahwa seluruh fitur aplikasi StudyLink berjalan sesuai dengan kebutuhan sistem.

## Kesimpulan

Aplikasi StudyLink berhasil dikembangkan sebagai aplikasi manajemen aktivitas akademik berbasis Android dengan dukungan backend REST API. Aplikasi ini mampu membantu mahasiswa dalam mengelola jadwal, tugas, ujian, dan data profil secara terintegrasi.

## **Estimasi waktu menggunakan PERT**

PERT (Program Evaluation and Review Technique)

Rumus: TE = (O + 4M + P) / 6

O = waktu optimis

M = waktu paling realistik

P = waktu pesimis

No	Kegiatan Level 1	Kegiatan Level 2	O (hari)	M (hari)	P (hari)
1	<b>Perencanaan</b>	Analisis kebutuhan	2	4	6
		Studi kelayakan	1	3	5
		Penyusunan WBS & timeline	1	2	3
2	<b>Desain Sistem</b>	Desain UI/UX	3	5	8
		Desain Arsitektur Sistem	2	4	6
		Desain Database	2	3	5
3	<b>Pengembangan (Development)</b>	Backend API (auth, booking, tracking)	5	10	15
		Frontend Mobile App	6	12	18
		Integrasi API + Map & GPS	4	8	12

No	Kegiatan Level 1	Kegiatan Level 2	O (hari)	M (hari)	P (hari)
4	<b>Testing &amp; Deployment</b>	Unit Testing	2	4	6
		System & Integration Testing	3	6	10
		Deployment Play Store	1	2	4
5	<b>Maintenance</b>	Fixing & Patch Update	2	4	7
		Monitoring & Support	3	6	9

## Gantt Chart sederhana

## Identifikasi 5 risiko proyek + cara mitigasi

### 1. Risiko: Gangguan pada Sistem GPS / Lokasi Tidak Akurat

Mitigasi:

- Gunakan API lokasi tingkat tinggi seperti **Google Maps / Mapbox**.
- Terapkan **kalibrasi GPS** otomatis + fallback ke WiFi & Cell Tower.
- Lakukan uji lokasi di area padat / gedung tinggi.
- Tambahkan fitur *manual pin location* jika GPS error.

### 2. Risiko: Keterlambatan Pengembangan (Deadline Tidak Terpenuhi)

Mitigasi:

- Buat WBS dan Gantt Chart yang jelas.
- Prioritaskan fitur inti (MVP: login, order, tracking, pembayaran).
- Gunakan metodologi Agile/Scrum dengan sprint pendek (1–2 minggu).
- Lakukan *daily standup* untuk memantau progress.
- Tambahkan buffer time.
- Tetapkan *template coding* & standard teknis agar developer konsisten.

### 3. Risiko: Aplikasi Error

Mitigasi:

- Gunakan **testing multi-device**
- Implementasi **load balancing** pada server.
- Gunakan **CI/CD** untuk memantau bug secara otomatis.
- Monitoring performa memakai **Firebase Crashlytics / Sentry**.

### 4. Risiko: Keamanan Data & Serangan Cyber

Mitigasi:

- Gunakan **HTTPS + SSL pinning**.
- Enkripsi data sensitif (AES, hashing password SHA-256/bcrypt).
- Terapkan **multi-factor authentication** (OTP/email).
- Audit keamanan rutin + penetration test.
- Batasi akses API dengan role-based access control.

### 5. Risiko: Aplikasi Tidak Sesuai Kebutuhan Pengguna (User Experience Buruk)

Mitigasi:

- Lakukan survei/observasi pengguna sebelum desain.
- Buat prototipe UI/UX (Figma) dan lakukan *user testing*.
- Gunakan desain yang sederhana (one-click order).
- Tambahkan fitur tutorial onboarding untuk pengguna baru.
- Perbaiki berdasarkan feedback pada setiap sprint.

**WBS**  
**GUGUS TUGAS (TASK) PROYEK**

Kebutuhan Non Fungsional	
Aspek	Deskripsi Kebutuhan Non Fungsional
Keamanan	Enkripsi data dan pembatasan hak akses pengguna.
Performa	Waktu respon cepat dan mampu memproses banyak transaksi.
Reliabilitas	Sistem memiliki tingkat availability $\geq 99\%$ , backup dan recovery otomatis.
Aksesibilitas	Antarmuka mudah digunakan dan mendukung perangkat mobile.
Skalabilitas	Sistem dapat dikembangkan sesuai peningkatan volume pengguna atau data.
Maintenance	Dukungan update fitur dan perbaikan bug tanpa gangguan operasional.
Audit Trail	Merekam seluruh aktivitas dan perubahan data untuk keamanan.

## Kesimpulan

Dokumen SRS ini menjadi pedoman utama dalam pengembangan sistem logistik agar sistem memenuhi kebutuhan bisnis, efisiensi operasional, serta keamanan dan keandalan dalam pengelolaan distribusi barang.

## Use Case Diagram



