

# BAB 1

## PENDAHULUAN

---

### 1.1 Latar Belakang

Sistem pelaporan PDDIKTI (Pangkalan Data Dikti) yang digunakan oleh Universitas Telkom saat ini masih menggunakan metode secara manual. Prosedur manual membutuhkan banyak tenaga kerja, memakan waktu, dan rentan terhadap kesalahan. Selain itu, karena penundaan pengiriman, data yang diberikan seringkali salah dan tidak akurat. Bisa mengalami masalah dalam mengirimkan data dari aplikasi neo feeder ke PDDIKTI.

Aplikasi T-Feeder adalah sistem yang dimaksudkan untuk membantu universitas telkom, mengelola dan melaporkan data PDDIKTI. Dengan menggunakan aplikasi ini, universitas dapat mengunggah data secara langsung ke portal PDDIKTI dari sistem internal mereka, menghindari proses manual yang memakan waktu. Aplikasi ini dirancang untuk membantu universitas telkom mengelola dan melaporkan data PDDIKTI secara elektronik.

Pengembangan aplikasi T-Feeder dikerjakan oleh dua divisi yaitu divisi Backend dan divisi Frontend. Backend adalah sebuah sistem di balik website atau aplikasi. Pengguna tidak melihatnya secara langsung, tapi backend yang membuat semua fungsi aplikasi bekerja. Backend memiliki tugas seperti mengidentifikasi database, mengidentifikasi kebutuhan laporan, membuat kode program, dan membuat API. Setelah kode program dibuat kemudian hasil pengembangan dilanjutkan oleh divisi frontend.

Permasalahan yang muncul adalah diperlukan penghubung antara aplikasi t-feeder dan aplikasi pelaporan pddikti. Untuk itu dibutuhkan api yang dapat menghubungkan aplikasi T-Feeder dengan PDDIKTI. Penelitian ini berfokus pada pengembangan aplikasi t feeder dimana pengembangan API aplikasi T-Feeder yang dilakukan pada penelitian merupakan salah satu bagian yang dikerjakan oleh divisi Backend

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, rumusan masalah pengembangan API pada aplikasi T-Feeder adalah Bagaimana mengembangkan API yang dapat mengirimkan data dari aplikasi t-feeder ke PDDIKTI ?

## 1.3 Tujuan

Penelitian pengembangan API pada aplikasi T-Feeder ini bertujuan untuk Mengembangkan dan mengimplementasikan API untuk aplikasi T-Feeder yang mampu melakukan pengiriman dari aplikasi t-feeder ke aplikasi PDDIKTI. Sehingga Mengurangi jumlah kesalahan dan waktu yang dibutuhkan dalam proses pelaporan data PDDIKTI.

## 1.4 Batasan Masalah

Pengembangan API pada aplikasi T-Feeder dibatasi pada hal-hal berikut:

1. Model waterfall yang dikembangkan hanya terbatas pada analisis kebutuhan, desain sistem, implementasi, pengujian, dan tidak melakukan setelah pengujian
2. Pengembangan API hanya akan mencakup aspek backend dari aplikasi T-Feeder dan tidak termasuk pengembangan frontend dan integrasi.
3. Pengembangan API T-Feeder yang dilakukan hanya terbatas pada pembuatan aplikasi API pada fitur login, profile, role user, daftar fakultas, dan daftar prodi tidak sampai pada tahapan integrasi
4. API T-Feeder ini hanya akan dikembangkan untuk pelaporan data yang diperlukan oleh PDDIKTI.
5. API T-Feeder ini hanya akan dikembangkan untuk platform web.

## 1.5 Definisi Operasional

Berikut definisi operasional yang ada pada penelitian ini:

1. API (Application Programming Interface): adalah antarmuka yang memungkinkan aplikasi perangkat lunak untuk berkomunikasi dan saling bertukar data. Ini menyediakan aturan dan definisi untuk membuat permintaan dan menerima respons antar aplikasi.
2. T-Feeder: adalah aplikasi berbasis web yang dimaksudkan untuk membantu universitas telkom, mengelola dan melaporkan data PDDIKTI.
3. PDDIKTI (Pangkalan Data Pendidikan Tinggi): adalah sistem basis data yang mengelola informasi tentang perguruan tinggi di Indonesia, termasuk data mahasiswa, dosen, dan program studi.
4. BSLA (Bagian Standar dan Layanan Akademik): merupakan sebuah bagian di dalam Direktorat Akademik di Universitas Telkom yang mengelola administrasi akademik termasuk penerbitan ijazah digital, mengelola fasilitas perkuliahan, berkoordinasi untuk pelaksanaan wisuda, mengelola pelaporan PDDIKTI, serta merencanakan, memonitor, dan mengevaluasi program kerja dengan pihak internal dan eksternal.
5. SSO (Single Sign-On): adalah teknologi yang memungkinkan pengguna untuk mengakses aplikasi dan situs web dengan satu akun pengguna. Saat pengguna mencoba mengakses aplikasi, sistem SSO akan mengotentikasi identitas mereka.

## 1.6 Metode Pengerjaan

Penelitian Pengembangan API Pada Aplikasi T-Feeder untuk Pelaporan PDDIKTI ini menggunakan metode Pengerjaan Waterfall. Metode waterfall merupakan pendekatan tradisional dalam pengembangan perangkat lunak yang membagi proses pengembangan menjadi beberapa tahap yang berurutan dan terstruktur. Berikut adalah penjelasan detail mengenai masing-masing tahapan tersebut:

### 1. Analisis Kebutuhan (Requirements Analysis)

Pada tahap ini, dilakukan pengumpulan dan analisis kebutuhan dari sistem yang akan dikembangkan. Kegiatan yang dilakukan meliputi:

- Identifikasi Kebutuhan Pengguna: Mengumpulkan kebutuhan dari berbagai pemangku kepentingan, seperti administrator kampus, staf IT, dan pihak pengelola PDDIKTI.
- Dokumentasi Kebutuhan: Menyusun dokumen spesifikasi kebutuhan (SRS - Software Requirements Specification) yang mendetail mengenai fungsi-fungsi yang harus ada dalam API dan bagaimana API tersebut akan berinteraksi dengan aplikasi T-Feeder dan PDDIKTI.

### 2. Desain Sistem (System Design)

Tahap desain sistem dibagi menjadi dua bagian utama:

- Desain Arsitektur: Menentukan arsitektur sistem, termasuk komponen-komponen utama, bagaimana komponen tersebut saling berinteraksi, serta teknologi yang akan digunakan.
- Desain Detail: Menyusun desain detail untuk setiap komponen, termasuk model data, struktur API, endpoint yang diperlukan, dan spesifikasi teknis lainnya.

### 3. Implementasi (Implementation)

Pada tahap ini, desain yang telah dibuat diimplementasikan menjadi kode program. Kegiatan yang dilakukan meliputi:

- Pengembangan API: Menulis kode untuk API sesuai dengan spesifikasi yang telah dibuat. Ini termasuk membuat endpoint, mengatur autentikasi dan otorisasi, serta memastikan API dapat berkomunikasi dengan aplikasi T-Feeder dan PDDIKTI.

- Pengujian Unit: Melakukan pengujian unit untuk memastikan setiap bagian dari API berfungsi sesuai dengan yang diharapkan.

#### 4. Pengujian (Testing)

Setelah tahap implementasi selesai, dilakukan pengujian lebih lanjut untuk memastikan bahwa API berfungsi dengan benar dan memenuhi semua kebutuhan yang telah ditentukan:

- Pengujian Integrasi: Menguji integrasi API dengan aplikasi T-Feeder dan sistem PDDIKTI untuk memastikan semua komponen bekerja dengan baik bersama-sama.
- Pengujian Sistem: Menguji API secara keseluruhan dalam lingkungan yang mirip dengan lingkungan produksi untuk memastikan semua fitur bekerja dengan baik.
- Pengujian Akseptansi Pengguna (UAT): Melibatkan pengguna akhir untuk menguji API dan memastikan bahwa API memenuhi kebutuhan mereka.

Dengan menggunakan metode waterfall, setiap tahapan di atas harus diselesaikan secara berurutan sebelum melanjutkan ke tahap berikutnya. Ini memastikan bahwa setiap aspek dari pengembangan API dilakukan dengan hati-hati dan terdokumentasi dengan baik, meskipun mungkin kurang fleksibel untuk perubahan yang mendadak dibandingkan dengan metode Agile.

### 1.7 Jadwal Pengerjaan

Kegiatan magang ini berlangsung selama 6 bulan mulai dari tanggal 5 februari dan berakhir pada 5 agustus. Berikut jadwal dan kegiatan yang dilakukan selama magang berlangsung:

Tabel 1 Jadwal Pengerjaan

No	Deskripsi Kerja	Februari			Maret				April				Mei				Juni				Juli				Agustus	
		1	2	3	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	
1	On Boarding Magang	■	■	■																						
2	Pembuatan API				■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■						
3	Review API																■	■	■	■						
4	Perbaikan API																							■	■	■