

# BAB 1

## PENDAHULUAN

---

### 1.1 Latar Belakang

Dalam rangka menunjang aspek keahlian profesional Fakultas Ilmu Terapan Universitas Telkom telah menyediakan sarana dan prasarana penunjang pendidikan dengan lengkap, namun sarana dan prasarana tersebut hanya menunjang aspek keahlian profesional secara teori saja. Dalam dunia kerja nantinya dibutuhkan keterpaduan antara pengetahuan akan teori yang telah didapatkan dari bangku perkuliahan dan pelatihan praktik di lapangan guna memberikan gambaran tentang dunia kerja yang sebenarnya.

Magang Kerja merupakan bentuk perkuliahan melalui kegiatan bekerja secara langsung di dunia kerja. Magang Kerja ini merupakan suatu kegiatan praktik bagi mahasiswa dengan tujuan mendapatkan pengalaman dari kegiatan tersebut, yang nantinya dapat digunakan untuk pengembangan profesi. Kegiatan magang kerja ini dilaksanakan di Bagian Standar dan Layanan Akademik atau yang dikenal sebagai BSLA, merupakan sebuah bagian di dalam Direktorat Akademik di bawah Koordinasi dari Wakil Rektor I Bidang Akademik Universitas Telkom yang memiliki tugas pokok Mengelola pelaporan PDDikti.

Pangkalan Data Pendidikan Tinggi (PDDikti) merupakan sebuah sistem penyimpanan data yang dikelola Pusat Data dan Informasi (Pusdatin) Kementerian Ristek dan Pendidikan Tinggi (Kemristekdikti). Perguruan tinggi dapat mengolah data yang menunjukkan profil perguruan tinggi tersebut melalui sebuah antarmuka aplikasi PDDikti *FEEDER*. Kemudian data tersebut disinkronisasi dengan data yang ada di Pusdatin Kemristekdikti. Data yang tersedia di PDDikti merupakan data yang akurat, karena proses pelaporan data akademik secara berkala dilakukan dua kali setiap semester dan perkembangan akademik setiap mahasiswa dapat ditampilkan pada aplikasi Forlap yang bisa diketahui oleh masyarakat luas [1].

Peran *Backend* sangat dibutuhkan dalam pembuatan atau pengembangan aplikasi *T-Feeder*. *Backend developer* bertugas untuk membuat model dan *controller* pada aplikasi *T-Feeder*. *Controller* mengandung proses *input* dari pengguna, kemudian mengirim perintah kepada *mode* dan *view* agar melakukan proses sesuai dengan masukan. Singkatnya, *controller* merupakan pemetaan aksi pengguna terhadap respons *system*. Dan Model mengandung fungsi-fungsi yang berhubungan dengan *database*, seperti *create*, *read*, *update*, dan *delete*.

*Backend* adalah tempat di mana proses pada suatu *system* informasi atau aplikasi berjalan, data dapat ditambahkan, diubah maupun dihapus. *Backend* biasanya mengurus segala jenis proses yang tidak berhubungan langsung dengan pengguna, seperti *server* dan basis data. *Backend* dibutuhkan dalam pengembangan sistem dan manajemen data pada *system*.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah yang dapat diambil yaitu :

1. Bagaimana mengimplementasikan API *backend* yang bersih dan terdokumentasi dengan baik untuk mendukung integrasi dengan aplikasi klien dan layanan eksternal?
2. Bagaimana mengintegrasikan *backend* aplikasi dengan sistem eksternal seperti basis data, layanan pihak ketiga, atau sistem autentikasi?
3. Bagaimana mengelola dan menyimpan data pengguna dengan efisien untuk memastikan konsistensi data?

## 1.3 Tujuan

Adapun tujuan dalam melaksanakan proyek tersebut :

- a. Untuk membantu terhubung nya aplikasi T-Feeder dengan portal PDDikti melalui API.
- b. Untuk memudahkan perguruan tinggi dalam menyusun dan mengirimkan laporan dalam jumlah besar ke aplikasi *Neo Feeder* PDDikti.

- c. Untuk memastikan bahwa data yang disampaikan oleh perguruan tinggi selalu akurat dan *up-to-date*.

#### 1.4 Batasan Masalah

Batasan masalah dapat berisi:

1. Aplikasi *web* yang dibangun menggunakan *Framework Laravel*,
2. Laporan ini berfokus pada pengembangan sebuah aplikasi *T-Feeder* berbasis *website*,
3. Aplikasi ini berfokus pada *backend* aplikasi *T-Feeder*

#### 1.5 Definisi Operasional

- a. *Backend* merupakan suatu program yang berjalan pada sisi *server (server-side)* yang melakukan tugas untuk berinteraksi langsung dengan basis data dalam melakukan manipulasi data ke basis data, sehingga *backend* tidak melakukan interaksi secara langsung kepada pengguna [2].
- b. Dalam melakukan pelaporan data, dibuatlah aplikasi *Feeder PDDikti*. Aplikasi tersebut merupakan aplikasi yang berfungsi menjamin kualitas, relevansi, keterjangkauan dan pemetaan dari data dan informasi Perguruan Tinggi. Penerapan aplikasi ini digunakan untuk melakukan sinkronisasi data perguruan tinggi yang memanfaatkan *web service*. *Web Service* 18 merupakan informasi-informasi yang bisa diakses oleh siapa saja dan oleh perangkat apa saja [3].
- c. API merupakan antarmuka yang digunakan untuk mengakses aplikasi atau layanan dari sebuah program [4]. API memungkinkan *developer* untuk memakai fungsi yang sudah ada dari aplikasi lain sehingga tidak perlu membuat ulang dari awal. Salah satu arsitektur *backend* adalah *Representational State Transfer (REST)*. *REST* merupakan seperangkat prinsip arsitektur yang melakukan transmisi data melalui antarmuka yang terstandarisasi seperti HTTP. *REST* API sendiri merupakan istilah yang

dipakai untuk layanan *web* yang mengimplementasikan arsitektur *REST* sebagai API [5].

- d. *Laravel Lumen* atau yang biasa disebut *Lumen* merupakan sebuah *micro-framework* berbasis bahasa pemrograman PHP yang ditujukan untuk pembuatan *microservices website* dan API yang cepat [6]. *Lumen* dikembangkan oleh pengembang *framework* PHP *Laravel*, yaitu *Taylor Otwell* sebagai sebuah proyek yang bertujuan untuk membuat *Laravel* versi ringan. *Lumen* dijadikan *framework* yang dapat membuat *REST API* dengan fiturnya yang kuat dan cepat seperti *Eloquent ORM*, *validation*, *middleware*, *routing*, dan lain sebagainya.
- e. Pengertian *MySQL* Pada perkembangannya, *MySQL* disebut juga *SQL* yang merupakan singkatan dari *Structured Query Language*. *SQL* merupakan bahasa terstruktur yang khusus digunakan untuk mengolah *database*. *SQL* pertama kali didefinisikan oleh *American National Standards Institute (ANSI)* pada tahun 1986. *MySQL* adalah sebuah sistem manajemen *database* yang bersifat *open source*. *MySQL* merupakan sistem manajemen *database* yang bersifat *relational*. Artinya, data yang dikelola dalam *database* yang akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan jauh lebih cepat. *MySQL* dapat digunakan untuk mengelola *database* mulai dari yang kecil sampai dengan yang sangat besar [7].

## 1.6 Metode Pengerjaan

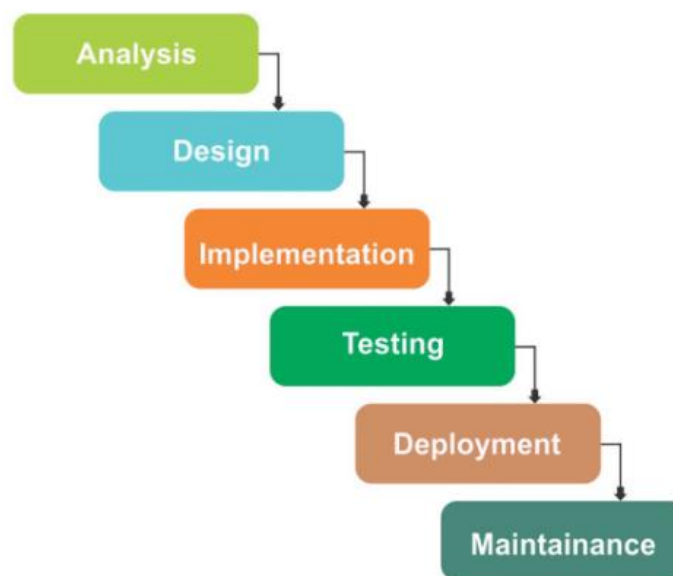
Metode ini sangat tepat untuk pengembangan perangkat lunak ketika menentukan solusi optimal untuk menyelesaikan permasalahan di Bagian Standar & Layanan Akademik.

### 1. Metode *Waterfall*

Pengembangan *system* ini dilakukan dengan menggunakan metode *Waterfall*. Metode *Waterfall* adalah pendekatan dalam pengembangan perangkat lunak dengan tahapan analisis kebutuhan, perancangan,

implementasi, pengujian, dan pemeliharaan. *Waterfall* adalah salah satu metode pengembangan sistem perangkat lunak. Metode ini memiliki keunggulan yakni proses pengembangan yang terstruktur dan terorganisir dengan baik. Kemudahan dalam pemahaman struktur hingga menghasilkan perangkat lunak dapat terdokumentasi dengan baik.

## Metode Waterfall dalam Pengembangan Perangkat Lunak



Gambar 2. 1 Metode Waterfall

Metode ini memungkinkan kontrol yang ketat terhadap jadwal, dan biaya. Nantinya hal ini akan berpengaruh pada kualitas, serta memungkinkan penyelesaian satu tahap sebelum memulai lanjutannya. Penggunaan Metode *Waterfall* mengikuti pendekatan linear atau sekuensial. Di mana pengembangan perangkat lunak dilakukan dalam tahap-tahap yang terdefinisi dengan jelas dan saling terkait [8].

Berikut adalah urutan tahapan Metode Air terjun atau *Waterfall* :

### 1. Analisis Kebutuhan (*Requirement Analysis*)

Tahap ini dimulai dengan memahami kebutuhan dan tujuan dari perangkat lunak yang akan dikembangkan. Tim pengembang akan mempelajari kebutuhan dan persyaratan pengguna, serta menentukan fitur-fitur dan fungsi yang diperlukan.

### 2. Perancangan (*Design*)

Setelah memahami kebutuhan, tim yang menggunakan Metode *Waterfall* merancang arsitektur, desain, dan spesifikasi teknis *software*. Perancangan juga melibatkan pembuatan diagram alir dan desain antarmuka pengguna.

### 3. Implementasi (*Implementation*)

Implementasi mengarah pada pembuatan kode program, dan pengujian untuk memastikan kualitas perangkat lunak yang dibangun.

### 4. Pengujian (*Testing*)

Setelah kode program selesai dibuat, tahap pengujian dilakukan untuk memastikan *software* berfungsi dengan baik. Hasilnya ialah perangkat lunak yang mampu memenuhi persyaratan pengguna.

### 5. Pemeliharaan (*Maintenance*)

Proses pemeliharaan baru dilaksanakan apabila produk sudah dikeluarkan oleh developer kepada konsumen. Tim pengembang akan terus memperbaiki, memperbarui, dan memperluas perangkat lunak sesuai dengan kebutuhan pengguna. Tahapan ini tidak hanya menjaga kondisi perangkat tetap berjalan baik, namun juga melakukan *upgrade* berkala. Dengan begitu tingkat kepuasan pengguna akan meningkat seiring dengan perawatan dan perbaikan yang dilakukan.

### 1.7 Jadwal Pengerjaan

Penjadwalan magang di Bagian Standar dan Layanan Akademik ( BSLA ) – Universitas Telkom dimulai dari tanggal 5 February 2024 – 5 Juni 2024 dengan jadwal 3 hari kerja secara offline di Ruang Layanan BSLA, Gedung Rektorat/Gedung Bangkit Lt.1 . Adapun rincian pelaksanaan praktik kerja magang yang dilaksanakan oleh penulis sebagai berikut:

Tabel 1. 1 Pelaksanaan Kerja

No	Deskripsi Kerja	Feb				Mar				Apr				Mei				Jun				Jul				Agu			
		1	2	1	1	2	3	4	2	3	4	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Preparation	■	■	■	■																								
2	Planing					■	■	■	■	■	■	■	■																
3	Design						■	■	■	■	■	■	■																
4	Develop						■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
5	Testing						■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■