

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sebagai salah satu kunci dalam memajukan institusi, Direktorat PuTI memiliki andil besar dalam pengembangan infrastruktur teknologi di lingkungan Telkom University. Terbukti hingga saat ini, Direktorat PuTI telah berhasil menciptakan 107 aplikasi, 330 website universitas, dan 1 framework yakni IGracias sebagai bentuk dukungan terhadap kegiatan akademik [1].

Sebagaimana ditunjukkan oleh laman Direktorat PuTI, website menjadi salah satu produk dengan jumlah terbanyak pada Telkom University [1]. Adanya inovasi berupa web tentunya membuat layanan akademik dan penunjangnya menjadi lebih efisien. Namun seiring dengan perkembangan universitas dan peningkatan jumlah civitas akademika, kompleksitas proses bisnis pada Telkom University menjadi kian meningkat [2]. Hal tersebut tentu berimplikasi pada munculnya masalah-masalah yang berkaitan dengan sistem informasi yang telah dibangun sebelumnya sehingga dibutuhkan beberapa perombakan layanan.

Problematika yang kerap kali dijumpai dalam dunia *web development* masa kini adalah tingginya jumlah pengguna dan arus pertukaran data [3]. Kondisi tersebut kerap kali membuat *cost* dari suatu aplikasi menjadi lebih tinggi apabila tidak ditangani dengan baik. Fakta tersebut tentunya sesuai dengan masalah yang dihadapi Direktorat PuTI akibat semakin tingginya kompleksitas *business process* pada universitas.

Salah satu solusi dari kondisi ini yakni melalui pemilihan bahasa pemrograman yang paling cocok dengan *environment* suatu aplikasi. Dengan bahasa pemrograman yang tepat, *developer* dapat dengan mudah mengurangi konsumsi sumber daya komputasi dari server tanpa mengurangi performa aplikasi yang bersangkutan sehingga biaya aplikasi pun dapat dikurangi [4].

Hingga saat ini terdapat beberapa bahasa pemrograman yang cukup populer diterapkan dalam pengembangan aplikasi berbasis web. Bahasa pemrograman yang dimaksud antara lain adalah PHP, Node.js, dan Python [3]. Tentunya ketiga bahasa tersebut memiliki performa yang berbeda-beda baik dari sisi penggunaan sumber daya komputasinya maupun kecepatan dalam merespon data.

Pengembangan aplikasi berbasis web pada Universitas Telkom sebagian besar menggunakan dua bahasa pemrograman yakni PHP dan Node.js. Sebagaimana dijelaskan oleh Komal Paliwal [5] dalam jurnalnya, bahasa pemrograman Node.js dinilai memiliki beberapa kelebihan tertentu dibandingkan dengan bahasa pemrograman PHP. Kelebihan tersebut antara lain kemudahan dalam penskalaan di sisi *service back end* dan kemampuan penanganan lalu lintas internet yang lebih baik.

Akan tetapi hingga saat ini masih terdapat beberapa aplikasi pada Direktorat PuTI yang menggunakan basis bahasa pemrograman PHP. Kondisi tersebut tentu perlu dibenahi mengingat semakin rumitnya proses bisnis yang ditangani oleh Direktorat PuTI.

Selain memilih bahasa pemrograman sesuai kebutuhan, solusi lain yang dapat dilakukan adalah penerapan arsitektur *microservices*. Arsitektur *microservices* merupakan sebuah teknik pengembangan aplikasi yang memungkinkan *developer* untuk menyusun aplikasi berbasis layanan yang saling terisolir [6]. Arsitektur ini memungkinkan aplikasi menjadi lebih mudah untuk dipelihara. Selain itu, jika terdapat permasalahan dalam salah satu servis aplikasi masih dapat berjalan dengan baik [6].

Namun sayangnya pengembangan aplikasi IGracias Universitas Telkom masih menggunakan arsitektur *monolith*. Arsitektur tersebut merupakan arsitektur tradisional dengan teknik mengemas *service* aplikasi ke dalam satu aplikasi utuh. Tentunya terdapat beberapa kekurangan dalam arsitektur *monolith* jika proses bisnis dari aplikasi tersebut terlalu kompleks. Kekurangan tersebut diantaranya cukup sulit untuk dilakukan pemeliharaan fitur, penambahan fitur, dan proses mengatasi *bugs* atau *error* yang muncul [6].

Oleh karena itu, beberapa upaya dilakukan oleh Direktorat PuTI untuk meningkatkan performa aplikasi dan layanan akademik pada Telkom University. Upaya tersebut antara lain dilakukannya migrasi bahasa pemrograman dan perombakan arsitektur aplikasi. Salah satu aplikasi yang menjadi fokus utama untuk dikembangkan ulang adalah aplikasi IGracias.

Migrasi struktur aplikasi dilaksanakan terhadap aplikasi IGracias yang awalnya masih menerapkan struktur *monolith* untuk kemudian dipecah menjadi beberapa aplikasi dalam bentuk *microservices*. Adapun salah satu aplikasi pecahannya adalah aplikasi Registrasi Mahasiswa Baru. Pengembangan aplikasi Registrasi Mahasiswa Baru juga menggunakan basis bahasa pemrograman yang lebih *up to date* yakni Node.js. Kedua langkah ini merupakan bentuk komitmen Direktorat PuTI untuk meningkatkan kualitas Telkom University.

1.2 Rumusan Masalah dan Solusi

Adapun berdasarkan latar belakang yang penulis buat, dapat diambil beberapa rumusan masalah sebagai berikut:

1. Bagaimana cara mengembangkan aplikasi IGracias agar lebih mudah untuk dilakukan pemeliharaan dari sisi arsitektur aplikasi?
2. Bagaimana cara meningkatkan performa sub aplikasi IGracias dari sisi bahasa pemrograman yang digunakan?

Kemudian solusi yang penulis tawarkan terhadap permasalahan yang telah dirinci antara lain sebagai berikut:

1. Pelaksanaan migrasi arsitektur pada aplikasi IGracias yang sebelumnya berupa arsitektur *monolith* menjadi *microservices*. Migrasi dilaksanakan dengan memecah aplikasi IGracias menjadi beberapa aplikasi sehingga pemeliharaan terhadap aplikasi menjadi lebih mudah. Salah satu sub aplikasi yang dikembangkan adalah aplikasi Registrasi Mahasiswa Baru;
2. Pelaksanaan migrasi bahasa pemrograman pada sub aplikasi IGracias. Sub aplikasi IGracias yakni aplikasi Registrasi Mahasiswa baru dikembangkan menggunakan bahasa pemrograman yang berbeda dari induk aplikasinya. Aplikasi IGracias memakai bahasa pemrograman PHP, sedangkan sub nya dikembangkan menggunakan bahasa pemrograman Node.js yang dianggap lebih mumpuni.

1.3 Tujuan

Dalam menyusun karya tulis ini, tujuan yang penulis harap dapat tercapai diantaranya:

1. Mengetahui cara mengembangkan aplikasi IGracias agar lebih mudah untuk dilakukan pemeliharaan dengan menerapkan arsitektur aplikasi yang tepat;
2. Mengetahui cara meningkatkan performa sub aplikasi IGracias dari sisi bahasa pemrograman.

1.4 Batasan Masalah

Penulis memberi batasan dalam pembahasan karya tulis ini antara lain sebagai berikut:

1. Pengaplikasian arsitektur *microservice* pada aplikasi induk IGracias;
2. Pengembangan sub aplikasi Registrasi Mahasiswa Baru dari sisi *backend persistence service*;
3. Penggunaan bahasa pemrograman Node.js dalam pengembangan sub aplikasi;

1.5 Penjadwalan Kerja

Adapun penjadwalan kerja penulis selama menjalankan magang dua semester dalam satuan minggu adalah sebagai berikut.

1. Penjadwalan kerja semester ganjil 2022/2023

No	Tanggal	Keterangan/Aktivitas Harian
1.	Minggu 1 (4 Juli 2022 – 8 Juli 2022)	- Learn TypeScript
2.	Minggu 2 (11 Juli 2022 – 15 Juli 2022)	- Learn TypeScript
3.	Minggu 3 (18 Juli 2022 – 22 Juli 2022)	- Learn DBMS PostgreSQL and MongoDB

No	Tanggal	Keterangan/Aktivitas Harian
4.	Minggu 4 (25 Juli 2022 – 29 Juli 2022)	- Learn RESTful API
5.	Minggu 5 (1 Agustus 2022 – 5 Agustus 2022)	- First meet internship
6.	Minggu 6 (8 Agustus 2022 – 12 Agustus 2022)	- Learn framework NestJS
7.	Minggu 7 (15 Agustus 2022 – 19 Agustus 2022)	- Learn framework NestJS
8.	Minggu 8 (22 Agustus 2022 – 26 Agustus 2022)	- Learn Prisma ORM
9.	Minggu 9 (29 Agustus 2022 – 2 September 2022)	- Learn Prisma ORM - Meeting with team
10.	Minggu 10 (5 September 2022 – 9 September 2022)	- Create CRUD API endpoint persistence service for MyTelu Form
11.	Minggu 11 (12 September 2022 – 16 September 2022)	- Create CRUD API endpoint persistence service for MyTelu Form - Create mock data for testing
12.	Minggu 12 (19 September 2022 – 23 September 2022)	- Create CRUD API endpoint persistence service for Master Data MyTelu - Create mock data for testing
13.	Minggu 13 (26 September 2022 – 30 September 2022)	- Create CRUD API endpoint persistence service for Master Data MyTelu
14.	Minggu 14 (3 Oktober 2022 – 7 Oktober 2022)	- Create CRUD API endpoint persistence service for Master Data MyTelu - Create mock data for testing
15.	Minggu 15 (10 Oktober 2022 – 14 Oktober 2022)	- Create CRUD API endpoint persistence service for Student Admission
16.	Minggu 16 (17 Oktober 2022 – 21 Oktober 2022)	- Create CRUD API endpoint persistence service for Student Admission - Create mock data for testing
17.	Minggu 17 (24 Oktober 2022 – 28 Oktober 2022)	- Create CRUD API endpoint persistence service for Student Admission - Meeting with team
18.	Minggu 18 (31 Oktober 2022 – 4 November 2022)	- Create and fixing error in all persistence CRUD API endpoint - Implement requested feature from back end aggregation
19.	Minggu 19 (7 November 2022 – 11 November 2022)	- Create and fixing error in all persistence CRUD API endpoint

No	Tanggal	Keterangan/Aktivitas Harian
		<ul style="list-style-type: none"> - Implement requested feature from back end aggregation - Meeting with team
20.	Minggu 20 (14 November 2022 – 18 November 2022)	<ul style="list-style-type: none"> - Create and fixing error in all persistence CRUD API endpoint - Implement requested feature from back end aggregation - Meeting with team
21.	Minggu 21 (21 November 2022 – 25 November 2022)	<ul style="list-style-type: none"> - Create and fixing error in all persistence CRUD API endpoint - Implement requested feature from back end aggregation
22.	Minggu 22 (28 November 2022 – 2 Desember 2022)	<ul style="list-style-type: none"> - Create and fixing error in all persistence CRUD API endpoint - Implement requested feature from back end aggregation
23.	Minggu 23 (5 Desember 2022 – 9 Desember 2022)	<ul style="list-style-type: none"> - Create and fixing error in all persistence CRUD API endpoint - Implement requested feature from back end aggregation
24.	Minggu 24 (12 Desember 2022 – 16 Desember 2022)	<ul style="list-style-type: none"> - Create and fixing error in all persistence CRUD API endpoint - Implement requested feature from back end aggregation
25.	Minggu 25 (19 Desember 2022 – 23 Desember 2022)	<ul style="list-style-type: none"> - Create and fixing error in all persistence CRUD API endpoint - Implement requested feature from back end aggregation
26.	Minggu 26 (26 Desember 2022 – 30 Desember 2022)	<ul style="list-style-type: none"> - Create and fixing error in all persistence CRUD API endpoint - Implement requested feature from back end aggregation

Tabel 1.1 Tabel Pelaksanaan Kerja Semester Ganjil 2022/2023

2. Penjadwalan kerja semester genap 2022/2023

No	Tanggal	Keterangan/Aktivitas Harian
1.	Minggu 1 (2 Jan 23 - 6 Jan 23)	<ul style="list-style-type: none"> - Implement requested feature from aggregation - Fixing bugs - Testing feature - Maintenance persistence API - Maintenance database - Meeting with team

No	Tanggal	Keterangan/Aktivitas Harian
2.	Minggu 2 (9 Jan 23 - 13 Jan 23)	- Implement requested feature from aggregation - Fixing bugs - Testing feature - Maintenance persistence API - Maintenance database - Meeting with team
3.	Minggu 3 (16 Jan 23 - 20 Jan 23)	- Implement requested feature from aggregation - Fixing bugs - Testing feature - Maintenance persistence API - Maintenance database - Meeting with team
4.	Minggu 4 (23 Jan 23 - 27 Jan 23)	- Implement requested feature from aggregation - Fixing bugs - Testing feature - Maintenance persistence API - Maintenance database - Meeting with team
5.	Minggu 5 (30 Jan 23 - 03 Feb 23)	- Implement requested feature from aggregation - Fixing bugs - Testing feature - Maintenance persistence API - Maintenance database - Meeting with team
6.	Minggu 6 (06 Feb 23 - 10 Feb 23)	- Implement requested feature from aggregation - Fixing bugs - Testing feature - Maintenance persistence API - Maintenance database - Meeting with team
7.	Minggu 7 (13 Feb 23 - 17 Feb 23)	- Implement requested feature from aggregation - Fixing bugs - Testing feature - Maintenance persistence API - Maintenance database - Meeting with team
8.	Minggu 8 (20 Feb 23 - 24 Feb 23)	- Implement requested feature from aggregation - Fixing bugs - Testing feature - Maintenance persistence API - Maintenance database - Meeting with team

No	Tanggal	Keterangan/Aktivitas Harian
9.	Minggu 9 (27 Feb 23- 3 Mar 23)	- Implement requested feature from aggregation - Fixing bugs - Testing feature - Maintenance persistence API - Maintenance database - Meeting with team
10.	Minggu 10 (6 Mar 23 - 10 Mar 23)	- Implement requested feature from aggregation - Fixing bugs - Testing feature - Maintenance persistence API - Maintenance database - Meeting with team
11.	Minggu 11 (13 Mar 23 - 17 Mar 23)	- Implement requested feature from aggregation - Fixing bugs - Testing feature - Maintenance persistence API - Maintenance database - Meeting with team
12.	Minggu 12 (20 Mar 23 - 24 Mar 23)	- Implement requested feature from aggregation - Fixing bugs - Testing feature - Maintenance persistence API - Maintenance database - Meeting with team
13.	Minggu 13 (27 Mar 23 - 31 Mar 23)	- Implement requested feature from aggregation - Fixing bugs - Testing feature - Maintenance persistence API - Maintenance database
14.	Minggu 14 (3 Apr 23 - 7 Apr 23)	- Implement requested feature from aggregation - Fixing bugs - Testing feature - Maintenance persistence API - Maintenance database
15.	Minggu 15 (10 Apr 23 - 14 Apr 23)	- Implement requested feature from aggregation - Fixing bugs - Testing feature - Maintenance persistence API - Maintenance database - Meeting with team
16.	Minggu 16 (17 Apr 23 - 18 Apr 23)	- Implement requested feature from aggregation - Fixing bugs - Testing feature

No	Tanggal	Keterangan/Aktivitas Harian
		<ul style="list-style-type: none"> - Maintenance persistence API - Maintenance database - Meeting with team
17.	Minggu 17 (26 Apr 23- 28 Apr 23)	<ul style="list-style-type: none"> - Implement requested feature from aggregation - Fixing bugs - Testing feature - Maintenance persistence API - Maintenance database - Meeting with team
18.	Minggu 18 (2 Mei 23 - 5 Mei 23)	<ul style="list-style-type: none"> - Implement requested feature from aggregation - Fixing bugs - Testing feature - Maintenance persistence API - Maintenance database - Meeting with team
19.	Minggu 19 (8 Mei 23 - 12 Mei 23)	<ul style="list-style-type: none"> - Implement requested feature from aggregation - Fixing bugs - Testing feature - Maintenance persistence API - Maintenance database - Meeting with team

Tabel 2.2 Tabel Pelaksanaan Kerja Semester Genap 2022/2023