

APLIKASI KURSUS ONLINE BERBASIS WEB SERVICE MENGGUNAKAN ARSITEKTUR MICROSERVICES

Dimas Purwanto¹, Wahyu Pramusinto², Gunawan Pria Utama³

^{1,2}Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
e-mail: ²wahyu.pramusinto@budiluhur.ac.id, ³gunawan.priautama@budiluhur.ac.id

ABSTRAK

Saat ini ada banyak aplikasi kursus online yang menyediakan berbagai materi yang dibutuhkan oleh masyarakat. Aplikasi kursus online terdiri dari banyak layanan yang saling terintegrasi. Untuk memudahkan dalam melakukan pemeliharaan dan pembaruan, aplikasi kursus online dibuat menggunakan arsitektur micro service. Pada arsitektur microservices, sebuah aplikasi dibagi menjadi beberapa layanan-layanan kecil yang masing-masing memiliki database terpisah. Setiap layanan akan saling berkomunikasi menggunakan web service. Web service berfungsi sebagai aplikasi transaksi data antar layanan yang terlibat di dalamnya. Web service berperan sebagai jembatan terhadap sistem sistem yang ada tanpa mempermasalahakan perbedaan platform yang digunakan. Aplikasi kursus online ini menggunakan RESTful web service dan pertukaran data dalam format JSON. Hasil dari penelitian ini adalah aplikasi kursus online yang dibagi menjadi 2 yaitu aplikasi mobile Android untuk admin dan aplikasi web untuk pengguna.

Kata Kunci: kursus online, RESTful web service, microservices

1. PENDAHULUAN

Saat ini ada banyak aplikasi kursus online yang menyediakan berbagai materi yang dibutuhkan oleh masyarakat. Sejak aktivitas belajar di sekolah ditiadakan karena pandemi corona, penggunaan platform pembelajaran online melonjak [1]. Kenaikan ini akan terus bertambah selama kebijakan belajar dari rumah masih diberlakukan oleh pemerintah.

Aplikasi kursus online terdiri dari banyak layanan yang saling terintegrasi. Untuk memudahkan dalam melakukan pemeliharaan dan pembaruan, aplikasi kursus online sebaiknya dibuat menggunakan arsitektur micro service. Mikroservice adalah sebuah pola pengembangan perangkat lunak yang baru dimana keseluruhan fungsi perangkat lunak disediakan oleh komponen – komponen perangkat lunak yang lebih kecil. Komponen-komponen ini menggunakan dan menyediakan layanan perangkat lunak dan dikembangkan oleh tim secara terpisah. [2].

Microservice merupakan sebuah arsitektur, di mana aplikasi perangkat lunak besar yang kompleks terdiri dari satu atau lebih layanan. Masing-masing microservices ini berfokus untuk melakukan dan menyelesaikan satu tugas saja dengan baik. Microservices dapat dikembangkan dengan bahasa pemrograman apapun. Mereka dapat berkomunikasi satu sama lain menggunakan Application Programming Interfaces (API) seperti REST. Sebuah microservice tidak perlu mengetahui arsitektur microservice lainnya [3].

Pada penelitian berjudul Implementasi Teknologi Mikroservice Pada Pengembangan Mobile Learning [2] dibuat sebuah aplikasi elearning yang terdiri dari beberapa layanan yaitu service soal, service nilai, service jawaban, service SMK, service mapel dan service pengguna. Semua request dari user akan melalui API Gateway yang selanjutnya akan mengarahkan request ini ke service atau layanan tertentu.

Pada penelitian berjudul Refactoring Arsitektur Microservice Pada Aplikasi Absensi PT. Graha Usaha Teknik dibuat sebuah aplikasi absensi. Penelitian ini melakukan refactoring arsitektur monolithic menjadi arsitektur microservice pada aplikasi absensi. Aplikasi absensi ini terdapat tujuh servis yang berhasil diidentifikasi, yaitu servis position, time constant, man hour, report daily, APD, man power dan OAuth2. Berdasarkan hasil pengujian perangkat lunak dengan menggunakan load test, arsitektur microservice dapat mengungguli arsitektur monolithic dalam mengembalikan response time [4].

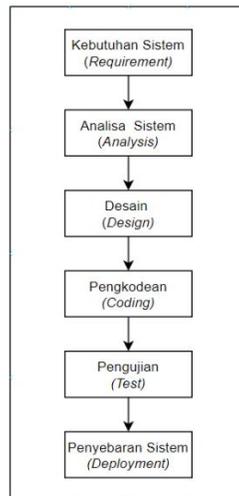
Web service merupakan sebuah perangkat lunak yang tidak terpengaruh oleh platform, arsitektur, maupun bahasa pemrograman, yang menyediakan layanan atau metode untuk pertukaran data yang dapat diakses melalui jaringan [5]. Salah satu jenis dari web service adalah REST atau RESTful (Representational State Transfer). REST sendiri memungkinkan system request dapat mengakses dan memanipulasi teks yang direpresentasikan dari sebuah web service. Web service API yang menggunakan REST disebut dengan RESTful API. Tidak seperti jenis web service lainnya, RESTful API tidak memiliki standar yang resmi untuk notasinya dikarenakan REST merupakan sebuah arsitektur [6].

Pada penelitian berjudul RESTful Web Service untuk Integrasi Sistem Akademik dan Perpustakaan Universitas Perjuangan dibuat sebuah web service dengan arsitektur REST. Penerapan teknologi web service dengan menggunakan arsitektur REST pada sistem informasi akademik dan sistem informasi perpustakaan mampu mengintegrasikan kedua sistem tersebut. Dengan penggunaan web service, proses input dan verifikasi data hanya dilakukan satu kali sehingga mengatasi terjadinya duplikasi data dan mengurangi pekerjaan input data [7].

Penelitian lain terkait RESTful API menggunakan autentikasi JSON Web Token (JWT) untuk membuat API menjadi lebih aman. Aplikasi sales order yang dibuat pada penelitian ini tidak dapat diakses jika tidak menggunakan token [8].

2. METODE PENELITIAN

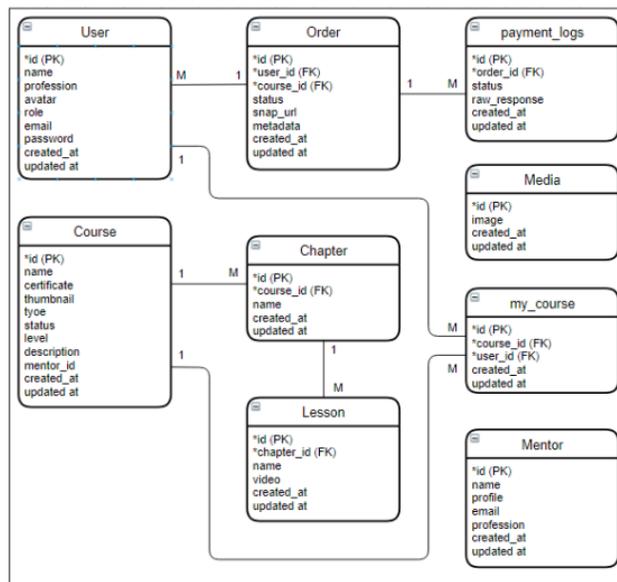
Metodologi yang digunakan dalam penelitian ini adalah dengan menggunakan metode Web Service Implementation Methodology (WSIM). Tahapan pengembangan dimulai dengan requirements hingga tahap deployment dapat dilihat pada gambar 1



Gambar 1. Metode Penelitian

2.1. Rancangan Database

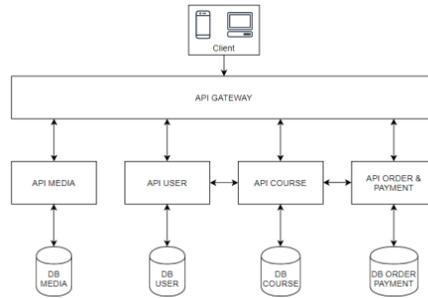
Aplikasi kursus online ini memiliki struktur database seperti pada gambar 2 di bawah ini



Gambar 2. Class Diagram

2.2. Rancangan Arsitektur Microservices

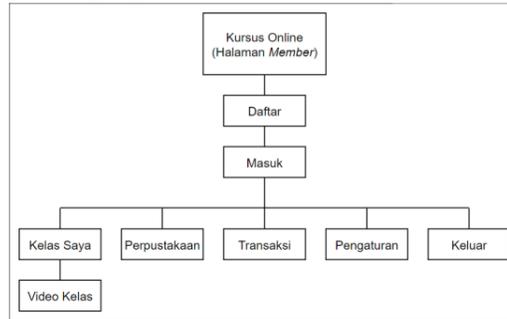
Pembuatan desain arsitektur microservices dilakukan untuk memberikan gambaran terhadap layanan yang dibutuhkan pada sistem usulan, dan memberikan gambaran terhadap alur komunikasi data antar layanan satu ke layanan lain. Diagram arsitektur microservices dapat dilihat pada gambar 3.



Gambar 3. Rancangan Arsitektur Microservices

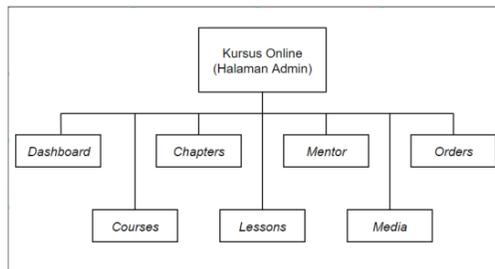
2.3. Rancangan Menu

Aplikasi kursus online ini dibagi menjadi 2 aplikasi yaitu aplikasi web dan aplikasi mobile Android. Pada gambar 4 dapat dilihat rancangan menu untuk halaman member. Member harus daftar dan login untuk bisa menggunakan menu kelas saya, perpustakaan, transaksi, pengaturan, keluar dan video kelas.



Gambar 4. Rancangan Menu Member

Pada gambar 5 dapat dilihat rancangan menu untuk halaman admin. Admin harus login terlebih dahulu untuk mengakses dashboard, chapters, mentor, orders, courses, lessons dan media.



Gambar 5. Rancangan Menu Admin

2.4. Rancangan Web Service Course

Pada tabel 1 bisa dilihat rancangan web service untuk layanan course.

Tabel 1. Rancangan Web Service Course

No	Nama Layanan	Method	URL	Parameter	Keluaran
1	Create Course	POST	{course_host} /api/courses	Request Body : name, certificate, thumbnail, type, status, level, price, mentor_id, description	status, data course, message
2	Update Course	PUT	{course_host} /api/courses/{id}	Request Body : name, certificate, thumbnail, type, status, level, price, mentor_id, description	status data course, message
3	Get All Course	GET	{course_host} /api/courses	Request Params (optional) :q, status, page	status, list data course, message
4	Detail Course	GET	{course_host} /api/courses/{id}		data course id
5	Delete Course	Delete	{course_host} /api/courses/{id}		status, message
6	Create Chapter	POST	{course_host} /api/chapters	Request Body : name, course_id	status, data chapter, message
7	Update Chapter	PUT	{course_host} /api/chapters/{id}	Request Body : name, course_id	status data chapter, message
8	Get All Chapter	GET	{course_host} /api/chapters	Request Params (optional) : course_id	status, list data chapter,message
9	Detail Chapter	GET	{course_host} /api/chapters/{id}		data chapter id
10	Delete Chapter	Delete	{course_host} /api/chapters/{id}		status, message
11	Create Lesson	POST	{course_host} /api/lessons	Request Body : name, video chapter_id	status, data lesson, message
12	Update Lesson	PUT	{course_host} /api/lessons/{id}	Request Body : name, video chapter_id	status data lesson, message
13	Get All Lesson	GET	{course_host} /api/lessons	Request Params (optional) : course_id	status, list data lesson, message
14	Detail Lesson	GET	{course_host} /api/lessons/{id}		data lesson id
15	Delete Lesson	Delete	{course_host} /api/lessons/{id}		status, message
16	Create My Courses	POST	{course_host} /api/my-courses	Request Body : user_id, course_id	status, data my- course, message
17	Get My Courses	GET	{course_host} /api/my-courses	Request Params (optional) :user_id	status, data my- course,message

4. HASIL DAN PEMBAHASAN

4.1. Spesifikasi Hardware dan Software

Spesifikasi yang digunakan untuk membuat implementasi aplikasi harus mendukung untuk mempermudah dalam pengoperasian kerja aplikasi sesuai yang diharapkan. Agar aplikasi dapat berjalan dengan baik, berikut spesifikasi yang diperlukan

a. Spesifikasi Minimal Perangkat Lunak (Software)

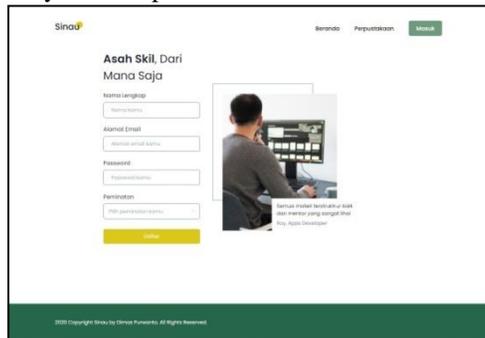
- Ubuntu 18.04.4 / Windows 10 64 Bit
- Nginx 1.14.0 / XAMPP
- Node JS 13.7.0
- PHP 7.4.12 / Laravel 7
- React JS
- Visual Studio Code
- Mysql Database
- Postman
- Google Chrome

b. Spesifikasi Minimal Perangkat Keras (Hardware)

- Processor : AMD A8-7410 APU
- Memory : 1GB
- Display : 136 x 768 (60 Hz)
- Harddisk : 500GB

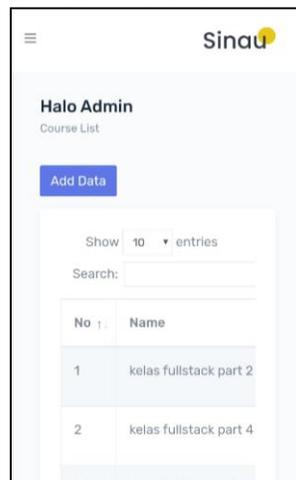
4.2. Tampilan Layar

Pada Gambar 6 merupakan tampilan layar untuk pendaftaran di website kursus online



Gambar 6. Halaman Pendaftaran

Pada gambar 7 merupakan tampilan layar aplikasi mobile untuk admin melihat data kursus online yang tersedia



Gambar 7. Halaman admin melihat data kursus online

4.3. Pengujian

Metode pengujian yang dilakukan adalah black box testing. Black box merupakan metode yang digunakan untuk menemukan kesalahan dan mendemonstrasikan fungsional aplikasi saat dioperasikan, apakah input diterima dengan benar dan output yang dihasilkan telah sesuai yang diharapkan. Pengujian black box pada service course bisa dilihat pada tabel 2

Tabel 2. Black Box Pada Service Courses

No	Nama Layanan	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	<i>Create Course</i>	Menginput <i>name, certificate, thumbnail, type, status, level, price, mentor_id, description</i> pada <i>request body</i> dengan metode POST	merespon dengan status <i>success</i>	Sesuai Harapan	Valid
2	<i>Update Course</i>	Mengubah data <i>Course</i> berdasarkan id dengan metode PUT	merespon dengan status <i>success</i> dan mendapatkan data <i>course</i>	Sesuai Harapan	Valid
3	<i>Get All Course</i>	Menampilkan semua data <i>course</i> dengan metode GET	merespon dengan status <i>success</i> dan mendapatkan data <i>course</i>	Sesuai Harapan	Valid
4	<i>Detail Course</i>	Menampilkan data <i>course</i> berdasarkan id dengan metode GET	merespon dengan status <i>success</i> dan mendapatkan data <i>course</i>	Sesuai Harapan	Valid
5	<i>Delete Course</i>	Menghapus data <i>course</i> berdasarkan id dengan metode DELETE	merespon dengan status <i>success</i>	Sesuai Harapan	Valid
6	<i>Create My Courses</i>	Menginput <i>user_id, course_id</i> pada <i>request body</i> dengan metode POST	merespon dengan status <i>success</i>	Sesuai Harapan	Valid
7	<i>Get My Courses</i>	Menampilkan semua data <i>my course</i> dengan metode GET	merespon dengan status <i>success</i> dan mendapatkan data <i>my course</i>	Sesuai Harapan	Valid
8	<i>Create Chapter</i>	Menginput <i>name, course_id</i> pada <i>request body</i> dengan metode POST Postman	Postman merespon dengan status <i>success</i>	Sesuai Harapan	Valid
9	<i>Update Chapter</i>	Mengubah data <i>Chapter</i> berdasarkan id dengan metode PUT	Postman merespon dengan status <i>success</i> dan mendapatkan data <i>chapter</i>	Sesuai Harapan	Valid
10	<i>Get All Chapter</i>	Menampilkan semua data <i>chapter</i> dengan metode GET	Postman merespon dengan status <i>success</i> dan mendapatkan data <i>chapter</i>	Sesuai Harapan	Valid
11	<i>Detail Chapter</i>	Menampilkan data <i>chapter</i> berdasarkan id dengan metode GET	Postman merespon dengan status <i>success</i> dan mendapatkan data <i>chapter</i>	Sesuai Harapan	Valid
12	<i>Delete Chapter</i>	Menghapus data <i>chapter</i> berdasarkan id dengan metode DELETE	Postman merespon dengan status <i>success</i>	Sesuai Harapan	Valid

13	<i>Create Lesson</i>	Menginput <i>name, video, chapter_id</i> pada <i>request body</i> dengan metode POST Postman	Postman merespon dengan status <i>success</i>	Sesuai Harapan	Valid
14	<i>Update Lesson</i>	Mengubah data <i>lesson</i> berdasarkan id dengan metode PUT	Postman merespon dengan status <i>success</i> dan mendapatkan data <i>lesson</i>	Sesuai Harapan	Valid
15	<i>Get All Lesson</i>	Menampilkan semua data <i>lesson</i> dengan metode GET	Postman merespon dengan status <i>success</i> dan mendapatkan data <i>lesson</i>	Sesuai Harapan	Valid
16	<i>Detail Lesson</i>	Menampilkan data <i>lesson</i> berdasarkan id dengan metode GET	Postman merespon dengan status <i>success</i> dan mendapatkan data <i>lesson</i>	Sesuai Harapan	Valid
17	<i>Delete Lesson</i>	Menghapus data <i>lesson</i> berdasarkan id dengan metode DELETE	Postman merespon dengan status <i>success</i>	Sesuai Harapan	Valid

5. KESIMPULAN

Setelah melalui tahap pengujian pada sistem ini, maka didapatkan kesimpulan bahwa dari hasil pengujian yang telah dilakukan, Aplikasi kursus online berbasis web service dengan arsitektur microservices telah berhasil dibangun dan diimplementasikan. Semua endpoint pada RESTful web service berhasil mengirimkan response sesuai harapan. Pada penelitian ini belum dibuat pengujian untuk menghitung kecepatan response antar layanan yang tersedia.

DAFTAR PUSTAKA

- [1] F. A. Burhan, "Pengguna Startup Pendidikan Melonjak Selama Masa Belajar di Rumah - Startup Katadata.co.id," *Katadata.co.id*, 2020. [Online]. Available: <https://katadata.co.id/agungjatmiko/digital/5e9a4212792a8/pengguna-startup-pendidikan-melonjak-selama-masa-belajar-di-rumah>. [Accessed: 30-Jun-2021].
- [2] M. Sendiang, S. Kasenda, and J. Purnama, "Implementasi Teknologi Mikroservice pada Pengembangan Mobile Learning," *J. Appl. Informatics Comput.*, vol. 2, no. 2, pp. 63–66, 2018.
- [3] R. V. Shahir Daya Nguyen Van Duy, Kameswara Eati, Carlos M Ferreira, Dejan Glozic, Vasfi Gucer, Manav Gupta, Sunil Joshi, Valerie Lampkin, Marcelo martins, Shishir Narain, "Microservices from Theory to Practice Creating Applications in IBM Bluemix Using the Microservices Approach," *Ibm*, p. 170, 2015.
- [4] R. Mufrizal and D. Indarti, "Refactoring Arsitektur Microservice Pada Aplikasi Absensi PT. Graha Usaha Teknik," *J. Nas. Teknol. dan Sist. Inf.*, vol. 5, no. 1, pp. 57–68, 2019.
- [5] R. Perwira and B. Santosa, "Implementasi Web Service pada Integrasi Data Akademik dengan Replika Pangkalan Data Dikti," *Telematika*, vol. 14, no. 01, pp. 1–11, 2017.
- [6] P. F. Tanaem, D. Manongga, and A. Iriani, "RESTful Web Service Untuk Sistem Pencatatan Transaksi Studi Kasus PT . XYZ," vol. 2, no. April, 2016.
- [7] R. Rizal and A. Rahmatulloh, "Restful Web Service Untuk Integrasi Sistem Akademik Dan Perpustakaan Universitas Perjuangan," *J. Ilm. Inform.*, vol. 7, no. 01, p. 54, 2019.
- [8] E. Edy, F. Ferdiansyah, W. Pramusinto, and S. Waluyo, "Pengamanan Restful API menggunakan JWT untuk Aplikasi Sales Order," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 3, no. 2, pp. 106–112, 2019.