



PENGEMBANGAN *REST API* DENGAN MENGGUNAKAN *EXPRESS JS* UNTUK Mencari Mentor Pribadi

Nasrul¹, Achmad Izhar²

^{1,2}Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640
nasrul@nurulfikri.ac.id, achm19068ti@student.nurulfikri.ac.id

Abstract

Currently there is a problem faced by many students, namely in understanding and mastering programming. These problems occur due to limitations in understanding program code, limited access to learning media, and limited learning time. To answer this challenge, the design and development of the Rest API for finding personal mentors is proposed as a solution that can improve the quality and effectiveness of learning program code. In this research, the Express JS framework is used to implement the Rest API, which interacts with third party APIs such as Midtrans and Redis. The MVC development approach and the Scrum method are applied to manage the application development process. The aim of this research is to design and build a Rest API that allows students to find a personal mentor that suits their needs in learning code. It is hoped that the use of Express JS and the Scrum method in the development of this application will facilitate the creation of effective and efficient private coding applications.

Keywords: Express JS, Mentor, MVC, Rest API, Scrum

Abstrak

Saat ini ada sebuah permasalahan yang dihadapi oleh banyak mahasiswa, yaitu dalam memahami dan menguasai pemrograman. Permasalahan tersebut terjadi akibat keterbatasan dalam memahami kode program, keterbatasan akses ke media pembelajaran, dan keterbatasan waktu pembelajaran. Untuk menjawab tantangan ini, perancangan dan pengembangan Rest API untuk pencarian mentor pribadi diusulkan sebagai solusi yang dapat meningkatkan kualitas dan efektivitas pembelajaran kode program. Dalam penelitian ini, digunakan *Framework Express JS* untuk mengimplementasikan *Rest API*, yang berinteraksi dengan *API* pihak ketiga seperti *Midtrans* dan *Redis*. Pendekatan pengembangan *MVC* dan metode *Scrum* diterapkan untuk mengatur proses pengembangan aplikasi. Tujuan dari penelitian ini adalah untuk merancang dan membangun sebuah *Rest API* yang memungkinkan mahasiswa mencari mentor pribadi yang sesuai dengan kebutuhan mereka dalam pembelajaran kode program. Diharapkan bahwa penggunaan *Express JS* dan metode *Scrum* dalam pengembangan aplikasi ini akan memfasilitasi penciptaan aplikasi *private coding* yang efektif dan efisien.

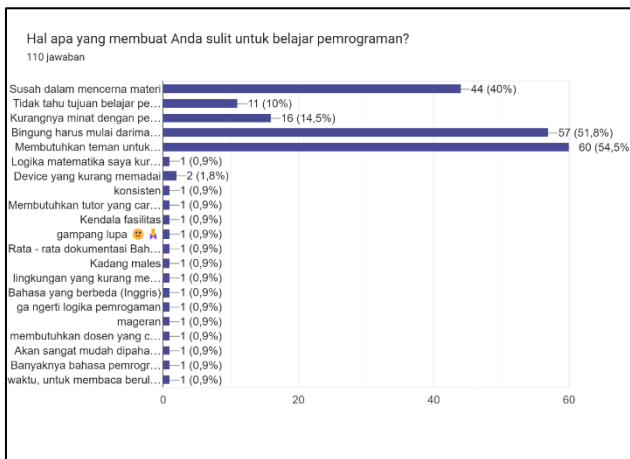
Kata kunci: Express JS, Mentor, MVC, Rest API, Scrum

1. PENDAHULUAN

Hampir semua kegiatan yang terjadi saat ini dapat dengan mudah dilakukan dengan bantuan teknologi. Ada banyak cara memajukan teknologi di Indonesia, salah satunya dengan bersekolah atau berkuliah di jurusan teknik, seperti teknik informatika, teknologi informasi, sistem informasi, sistem informatika, manajemen informatika atau jurusan lainnya. Menurut penelusuran yang dilakukan, setiap tahunnya lulusan TI di Indonesia tidak kurang dari 400 ribu orang, harapannya lulusan ini menjadi fondasi perkembangan teknologi di Indonesia yang tertinggal dari beberapa negara lain yang ada di dunia saat ini.

Akan tetapi mayoritas lulusan TI Indonesia saat ini masih belum memenuhi sumber daya manusia yang dibutuhkan oleh industri pada saat ini, walaupun SDM mahasiswa dalam ranah teknologi di Indonesia diperkirakan dapat mencapai 200 ribu orang. dan jumlah tersebut akan terus bertambah, seiring dengan perkembangan teknologi informasi yang begitu naik pesat [1]. Ada berbagai penyebab hal tersebut dapat terjadi, berdasarkan identifikasi Narenda selaku CEO Dicoding, saat ini ada 3 masalah yang menyebabkan permasalahan terjadi, pertama faktor kurikulum pembelajaran, kedua karena kualitas pengajar TI belum merata, dan yang terakhir adalah dari kualitas sumber daya manusia belum memadai [2].

Setelah mengetahui permasalahan pada saat ini, peneliti mengadakan survei lanjutan kepada 110 mahasiswa jurusan Teknik Informatika (TI) dimana hasil survei lanjutan tersebut menyimpulkan bahwa terdapat 46 mahasiswa jurusan TI merasa susah dalam mencerna materi pemrograman. Dari 12 mahasiswa menyatakan bahwa mereka tidak tahu tujuan belajar pemrograman, 16 mahasiswa kurang minat dengan pemrograman, 61 mahasiswa bingung memulai belajar pemrograman dari mana dan 61 mahasiswa membutuhkan teman untuk dapat mempelajari pemrograman bersama dan sisanya menjawab yang lainnya. Adapun survei lanjutan terhadap 110 mahasiswa tersebut seperti halnya yang terlampir pada gambar 1.



Gambar 1. Jawaban Survei Lanjutan Terhadap 110 Mahasiswa

Dari hasil survei lanjutan yang dilakukan, maka telah diperoleh 98,5% dari 110 mahasiswa yang melakukan pengisian survei menyatakan bahwa membutuhkan mentor pribadi untuk belajar pemrograman dengan mudah dan berkualitas dengan fitur pencarian dan fitur reservasi mentor yang mudah. Hasil survei lanjutan dapat dilihat pada gambar 2.



Gambar 2. Survei Membutuhkan Mentor untuk Belajar Pemrograman

Berdasarkan permasalahan saat ini dan survei yang telah dilakukan, maka peneliti melakukan rancangan dan pembangunan *Rest API* menggunakan *express js* untuk mencari mentor pribadi. *Rest API* yang akan dikembangkan

diharapkan dapat memberikan kontribusi terbaik pencarian mentor sesuai dengan kebutuhan dan preferensi pengguna.

Dengan mentoring, pendampingan dan pendekatan tertentu lebih luas diharapkan dapat membangun hubungan antara mentor dan *mentee* dalam waktu yang panjang [3], sehingga dapat memperluas dan meningkatkan aksesibilitas pembelajaran individu yang mungkin terbatas oleh faktor geografis atau keterbatasan pertukaran informasi, diskusi, dan penjelasan konsep dan praktik pemrograman. Selain itu, diharapkan implementasi aplikasi *private coding* dengan *express js* memberikan manfaat signifikan bagi mahasiswa yang ingin belajar pemrograman dan mencari mentor pribadi.

2. METODE PENELITIAN

Penelitian ini memiliki rancangan penelitian seperti metode analisis, metode dalam pengumpulan data, dan metode untuk pengembangan aplikasi *private coding*. Metode analisis ini menggunakan metode kualitatif dengan kuesioner yang berisi pertanyaan relevan untuk mengetahui hal-hal yang dibutuhkan pengguna aplikasi dalam menggunakan aplikasi *private coding* dan apa yang dibutuhkan oleh peneliti dalam mengembangkan penelitian ini menjadi lebih baik dan bermanfaat dalam jangka waktu panjang untuk masyarakat.

2.1 Metode Pengumpulan Data

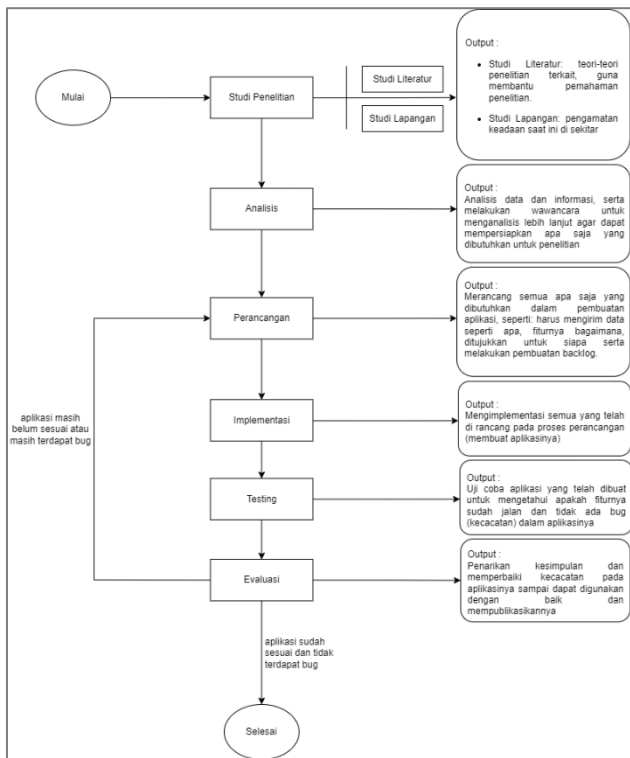
Metode pengumpulan data dilakukan dengan kondisi alami untuk mempermudah kebutuhan penelitian dengan sumber data utama dan teknik pengumpulan data sebagai berikut:

1. Wawancara calon pengguna aplikasi, mentor, atau individu yang terlibat dalam proses pembelajaran agar memberikan pemahaman mendalam tentang kebutuhan, harapan, pengalaman pengguna serta perspektif *private coding* dan pencarian mentor.
2. Observasi dan pengamatan langsung pengguna aplikasi dan melibatkan mentor pembelajaran.
3. Kuesioner mahasiswa ranah teknologi (jurusan teknik informatika, sistem informasi, teknologi informasi, dan lainnya) terkait pemrograman dan kuesioner pengguna serta mentor pemrograman untuk mendapatkan umpan balik aplikasi *private coding*.

Selain metode analisis dan pengumpulan data, penelitian ini menggunakan *scrum* untuk pengembangan aplikasi. Metode *Scrum* adalah kerangka kerja yang sifat pekerjaannya kompleks dan cepat berubah. *Scrum* ini dapat menjawab permasalahan yang adaptif kompleks dan menghasilkan kreativitas dan inovatif [4]. Setiap acara di dalam *Scrum* memiliki batasan waktu, yang artinya selalu memiliki durasi maksimum. Pada saat *sprint* dimulai, durasinya tetap dan tidak dapat diperpendek maupun diperpanjang. Acara-acara lainnya dapat diakhiri saat tujuan dari acara tersebut telah tercapai [5].

2.2 Tahapan Penelitian

Tahapan penelitian menggunakan metode *Scrum* dapat dilihat pada gambar 3 berikut.



Gambar 3. Tahapan Penelitian

Tahapan penelitian pada gambar 3 dapat dijelaskan sebagai berikut:

1. Studi Literatur

Studi literatur mencari referensi artikel, jurnal penelitian sebelumnya dan pembuatan *prototype* aplikasi berbasis *web*. *Prototype* adalah versi tahap awal dari sistem perangkat lunak yang digunakan untuk memberikan gambaran tentang sebuah ide, menguji desain, menemukan sebanyak mungkin masalah, dan menemukan solusi untuk masalah tersebut [6]. Hasil studi literatur dalam penelitian ini berupa pembuatan model penelitian untuk menjadi acuan dan hal apa saja yang dibutuhkan dan diperlukan agar mencapai tujuan penelitian.

2. Analisis

Pada tahap analisis, peneliti mengidentifikasi permasalahan yang ada terkait proses pembuatan *Rest API* aplikasi *private coding*, yang nantinya dikembangkan untuk memecahkan permasalahan dalam penelitian ini. Dengan mengidentifikasi permasalahan tersebut, peneliti kemudian menganalisis hasil wawancara untuk membuat *user stories*. *User story* menjelaskan siapa saja yang akan menjadi pengguna sistem, tujuannya dan fitur-fitur yang ada di dalam aplikasi [7].

3. Pembuatan *Product Backlog*

Pada tahapan pembuatan *product backlog* ini, peneliti membuat berdasarkan *user stories* dari analisis yang

dilakukan dan membuat rangkaian alur *Rest API*. *Product backlog* berkembang seiring dengan berkembangnya produk dan lingkungan produk dikembangkan [8].

4. Implementasi

Pada tahap implementasi yaitu pembuatan *Rest API* aplikasi *private coding* berbasis web berdasarkan *product backlog* yang sudah dibuat sebelumnya. Pengembangan *Rest API* aplikasi *private coding* menggunakan pendekatan arsitektur *MVC*, yaitu dengan memisahkan lapisan data (*model*), lapisan tampilan (*view*) dan lapisan penghubung antara data dan tampilan (*controller*). Dengan penerapan arsitektur ini diharapkan pengembangan aplikasi dapat bersifat *modular* dan memberikan kemudahan dalam proses *maintenance* kode program serta mendapatkan performa aplikasi yang baik [9].

5. Testing

Pada penelitian ini akan dilakukan dua pengujian yaitu *Black Box Testing*, *User Acceptance Test (UAT)* dan kuisioner pengguna dengan analisis menggunakan pengukuran *Skala Likert (Likert's Summated Rating)*. *Black Box Testing* sebagai metode untuk menguji fungsionalitas aplikasi. Sedangkan *UAT* sebagai metode pengujian kesesuaian antara hasil pengembangan aplikasi dengan kebutuhan pengguna [10]. *Skala Likert* adalah skala yang dapat digunakan untuk membandingkan skor sikap, pendapat dan persepsi seseorang atau sekelompok orang untuk melihat perkembangan atau perubahan sikap sebelum dan sesudah eksperimen [11].

6. Kesimpulan

Pada tahap ini, maka akan ditarik kesimpulan berdasarkan tahapan-tahapan yang telah dilakukan pada tahapan sebelumnya dan peneliti akan memberikan saran untuk dikembangkan lagi oleh peneliti selanjutnya.

3. HASIL DAN PEMBAHASAN

3.1 Analisis dan Perancangan

Berikut ini hasil dan pembahasan rancang bangun *rest API* menggunakan *Express JS* guna mencari mentor pribadi untuk pembelajaran:

1. *User Stories*

Setelah melakukan tahapan wawancara berupa kuisioner kepada masyarakat yang berpotensi menjadi pengguna aplikasi *private coding*, peneliti menganalisis data yang telah dikumpulkan untuk kemudian dibuat beberapa *user stories* yang terdapat di dalam tabel 1.

Tabel 1. *User Stories*

No	Sebagai	Saya Ingin	Sehingga
1	User	Ada verifikasi <i>email</i>	<i>Email</i> saya tidak digunakan oleh orang lain.
2	Admin	Melihat daftar semua mentor yang telah terdaftar di aplikasi	Saya dapat mengelola dan memverifikasi profil mentor dengan lebih mudah.
3	Admin	Melihat daftar	Saya dapat mengelola data

No	Sebagai	Saya Ingin	Sehingga
		semua siswa yang telah terdaftar di aplikasi	siswa dan memastikan validitas informasi yang diberikan.
4	Mentor	Membuka kursus	Saya bisa menawarkan keahlian dan pengetahuan saya kepada para siswa.
5	Admin	Memverifikasi kursus yang akan dibuat mentor	Mentor tidak dapat sembarangan membuka kursus
6	Mentor	Mengubah informasi kursus yang telah saya buat	Saya dapat memberikan detail yang akurat dan terkini kepada para siswa tanpa harus membuat ulang kursus lagi
7	Mentor	Melihat informasi kursus yang telah saya buat	Saya bisa mendapatkan informasi terbaru kursus saya
8	Mentor	Mengatur jadwal ketersediaan mengajar saya	Saya dapat menghindari tabrakan jadwal dan memungkinkan siswa untuk memesan sesi kursus dengan mudah.
9	Mentor	Melihat daftar siswa yang telah memesan kursus saya	Saya dapat mempersiapkan diri dan materi yang sesuai sebelum sesi kursus dimulai.
10	Siswa, Tamu	Melihat daftar mentor dan kursus yang tersedia	Saya dapat memilih mentor yang cocok dan kursus yang sesuai dengan kebutuhan belajar saya.
11	Siswa	Melakukan pemesanan kursus dari mentor pilihan saya	Saya bisa mendapatkan akses untuk belajar dari mentor yang kompeten dan berkualitas.
12	Siswa	Melakukan pembayaran kursus dengan berbagai metode pembayaran yang berbeda	Saya dapat memilih opsi pembayaran yang paling nyaman dan sesuai dengan preferensi saya.
13	Siswa	Menerima invoice transaksi setelah memesan kursus	Saya dapat melihat informasi penting seperti nama mentor, detail kursus, dan tanggal kursus berlangsung.
14	Admin	Memproses pembayaran kursus yang telah dipesan oleh siswa	Saya dapat memastikan transaksi pembayaran berjalan lancar dan memberikan akses kepada siswa untuk memulai kursus.
15	Siswa	Membatalkan pemesanan kursus dengan syarat dan ketentuan tertentu	Jika ada perubahan rencana mendadak, saya bisa membatalkan kursus dengan cara yang mudah dan jelas.
16	Siswa	Memberikan rating dan komentar setelah menyelesaikan kursus	Pengalaman belajar saya dapat membantu mentor dalam meningkatkan kualitas pengajaran dan membantu calon siswa lain untuk memilih mentor yang tepat.
17	Siswa	Melihat pesan dan notifikasi dari mentor atau admin terkait kursus atau transaksi saya	Saya tetap mendapatkan informasi terkini dan dapat merespons pesan dengan tepat waktu.
18	Mentor	Melihat rating dan komentar yang diberikan oleh siswa	Saya dapat mengevaluasi dan meningkatkan kualitas pengajaran saya untuk memberikan pengalaman

No	Sebagai	Saya Ingin	Sehingga
		terhadap kursus saya	belajar yang lebih baik bagi siswa.
19	Mentor	Menerima notifikasi ketika ada pemesanan kursus baru	Saya dapat segera menyiapkan diri dan menyusun materi sesuai dengan kebutuhan siswa yang telah memesan kursus.
20	Siswa	Melihat daftar kursus yang sedang berlangsung	Saya dapat bergabung dalam sesi kursus yang sedang berjalan jika sesuai dengan jadwal dan kebutuhan belajar saya.
21	Siswa	Melihat rekomendasi mentor berdasarkan minat dan keterampilan saya	Saya bisa lebih mudah menemukan mentor yang tepat dan sesuai dengan bidang yang saya inginkan.
22	Siswa	Menerima notifikasi saat kursus yang saya pesan akan segera dimulai	Saya dapat mempersiapkan diri dan mengingatkan diri saya tentang jadwal kursus yang telah saya pesan.
23	Siswa	Melihat riwayat kursus yang telah saya selesaikan	Saya bisa melacak perkembangan belajar saya dan mengingat kembali materi yang telah dipelajari di masa lalu.
24	Mentor	Melihat daftar siswa yang telah menyelesaikan kursus saya	Saya dapat memberikan rekomendasi atau testimonial kepada siswa-siswa tersebut berdasarkan performa dan prestasi belajar mereka.

2. Product Backlog

Berdasarkan gambaran *user stories*, peneliti merangkum fitur aplikasi *private coding* berisi penjelasan *task* atau tugas yang dilakukan dalam pengembangan aplikasi, prioritas fitur dan estimasi *Story Points* sebagai ukuran kompleksitas atau waktu pengerjaan setiap *task*. Dalam penjelasannya, setiap *Task* diberi penilaian prioritas sesuai kepentingannya. Untuk mengukur kompleksitas setiap *Task*, digunakan skala *Story Points*. 1 nilai *Story Points* memerlukan estimasi waktu pengerjaan sekitar 2-8 jam, 2 nilai *Story Points* memerlukan estimasi waktu pengerjaan antara 10-20 jam dan 3 nilai *Story Points* memerlukan estimasi waktu pengerjaan sekitar 21-40 jam. Deskripsi aplikasi disertakan untuk memberikan gambaran tentang tujuan dan fungsi aplikasi. Dengan informasi yang lengkap ini, diharapkan pengembangan aplikasi *private coding* dapat dilakukan dengan terstruktur dan efisien yang dapat dilihat pada tabel 2.

Tabel 2. Product Backlog

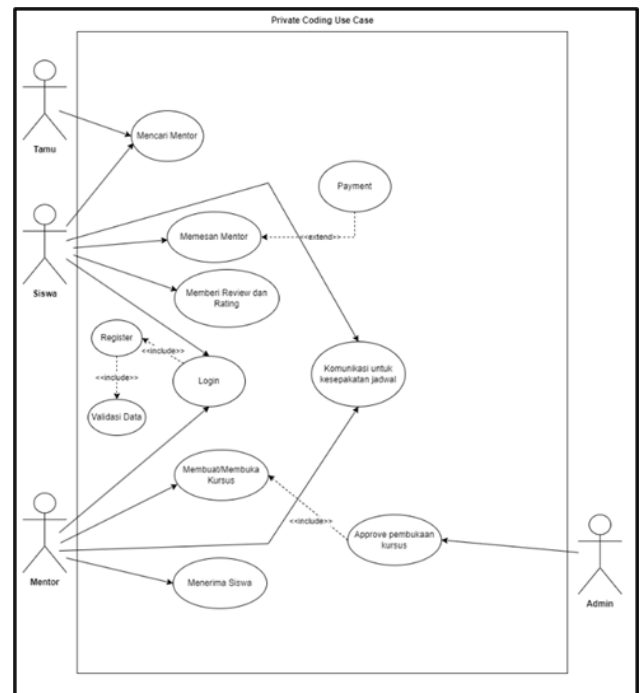
No	Task	Priority	Story Points	Deskripsi
1	Pembuatan <i>User Requirement</i>	Tinggi	1	Menyusun dan mendokumentasikan kebutuhan pengguna secara rinci untuk mengidentifikasi fitur-fitur utama aplikasi
2	Pembuatan <i>Use Case</i>	Tinggi	1	Menggambarkan interaksi antara aktor dan sistem untuk setiap fitur yang

No	Task	Priority	Story Points	Deskripsi
				teridentifikasi dalam <i>user requirement</i> .
3	Pembuatan <i>Class Diagram</i>	Tinggi	1	Mendesain struktur dan hubungan antara kelas-kelas dalam aplikasi untuk memodelkan logika dan relasi data
4	Pembuatan <i>Activity Diagram</i>	Tinggi	1	Menggambarkan alur kerja dari berbagai proses dalam aplikasi untuk memahami bagaimana pengguna berinteraksi dengan sistem.
5	Pembuatan <i>Schema Database</i>	Tinggi	1	Merancang skema <i>database</i> untuk menyimpan informasi/data aplikasi <i>private coding</i> .
6	<i>Migration Table Database</i>	Tinggi	1	Melaksanakan migrasi skema basis data ke dalam sistem basis data untuk memastikan struktur basis data sesuai desain.
7	<i>Setup Redis</i>	Tinggi	1	Mengonfigurasi dan mengimplementasikan Redis sebagai penyimpanan cache untuk meningkatkan kinerja aplikasi.
8	<i>Seeder for dummy data</i>	Rendah	1	Membuat data <i>dummy</i> untuk keperluan pengujian dan pengembangan aplikasi.
9	<i>API Login dan Logout</i>	Tinggi	1	Menerapkan <i>API</i> untuk mengelola proses <i>login</i> dan <i>logout</i> pengguna dengan menggunakan token JWT.
10	<i>API Register</i>	Tinggi	2	Mengimplementasikan <i>API</i> untuk mengizinkan pengguna mendaftar dan membuat akun baru.
11	<i>Verifikasi Email menggunakan OTP</i>	Tinggi	2	Menerapkan mekanisme verifikasi email dengan menggunakan <i>one-time password</i> (OTP).
12	<i>Create Token dan Refresh Token</i>	Tinggi	2	Membuat dan mengimplementasikan token dan <i>refresh</i> token untuk mengelola otentikasi dan otorisasi pengguna.
13	<i>API Profile</i>	Sedang	2	Mengimplementasikan <i>API</i> untuk mengelola profil pengguna.
14	<i>API Course</i>	Tinggi	3	Menerapkan <i>API</i> untuk menampilkan informasi tentang kursus yang tersedia.
15	<i>API buy course</i>	Tinggi	3	Membuat <i>API</i> untuk memproses pembelian kursus oleh siswa.
16	<i>API transactions</i>	Tinggi	2	Terapkan <i>API</i> untuk tampilkan informasi transaksi siswa.

No	Task	Priority	Story Points	Deskripsi
17	<i>API notifications</i>	Sedang	1	Mengimplementasikan <i>API</i> untuk kelola notifikasi yang dikirim oleh aplikasi.
18	Integrasi dengan <i>Payment gateway (midtrans)</i>	Sedang	2	Mengintegrasikan sistem pembayaran dengan midtrans atau <i>gateway</i> pembayaran lainnya.
19	<i>API review course</i>	rendah	2	Menerapkan <i>API</i> untuk memungkinkan siswa memberikan ulasan dan penilaian terhadap kursus yang telah diikuti.

3. Use Case Diagram

Pada tahapan analisis sistem yang akan digunakan untuk mengetahui kebutuhan sistem, akan dibuat sebuah *use case diagram*. *Use case diagram* merupakan sebuah *tools* yang digunakan untuk membuat pemodelan interaksi *user* dengan sistem [12]. *Use case diagram* bisa mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat [13].



Gambar 4. Use Case Diagram Rest API Private Coding

Pada gambar 4 merupakan *use case diagram* rancang bangun *Rest API* untuk mencari mentor pribadi dengan penjelasan sebagai berikut:

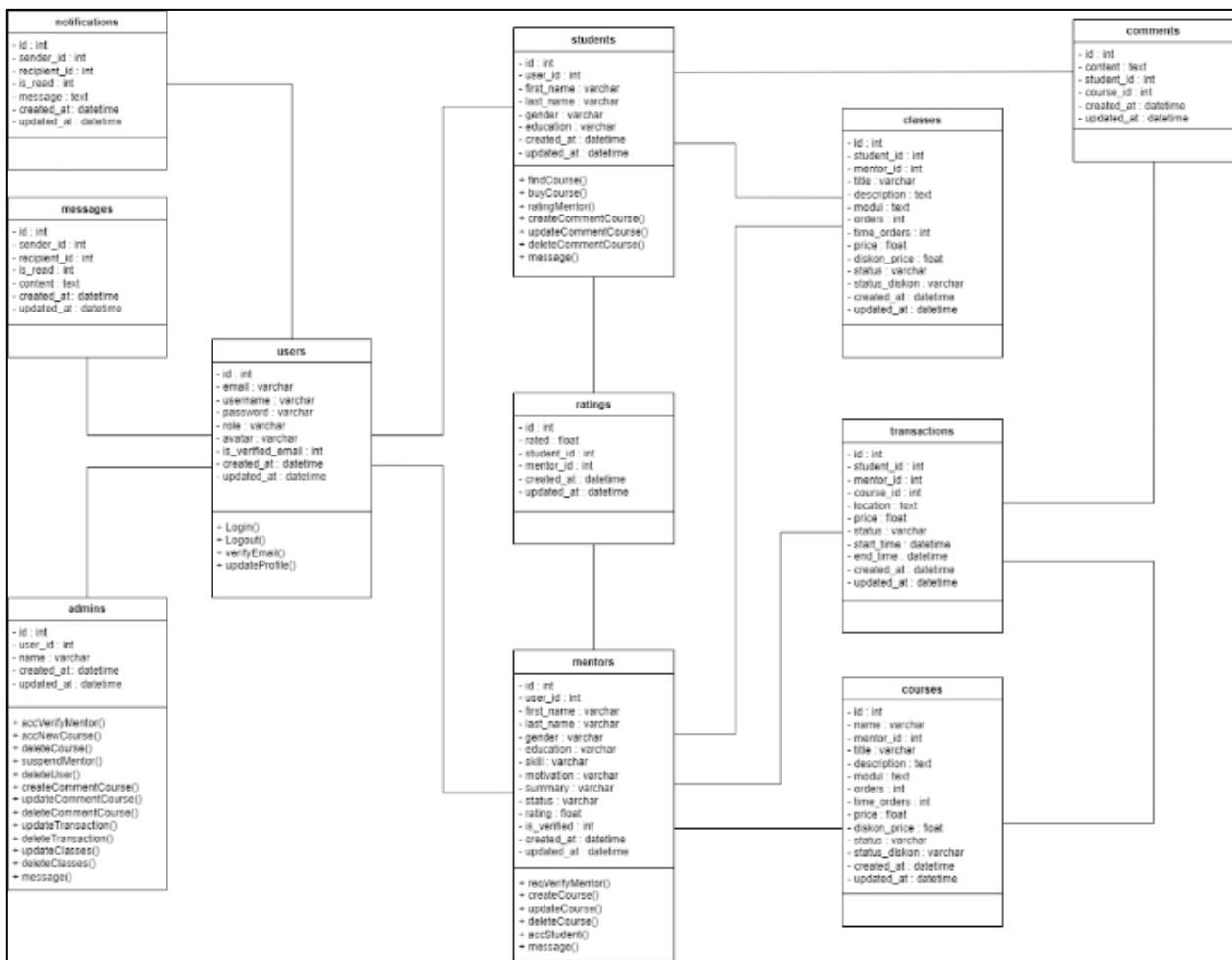
- a. Siswa dapat mencari mentor, berkomunikasi untuk membentuk kesepakatan jadwal bersama dengan mentor, kemudian siswa melakukan pemesanan mentor pemrograman yang dibutuhkan dimana ketika siswa melakukan pemesanan mentor siswa harus melakukan *payment* sesuai ketentuan dan memberikan *review* mentor pemrograman pada aplikasi. Tidak

hanya itu, siswa melakukan *login* tetapi sebelum *login* aplikasi, siswa melakukan registrasi akun dahulu dan validasi data agar akun yang didaftarkan oleh siswa dapat digunakan baik.

- b. Adapun mentor melakukan *login* pada aplikasi *private coding*, tetapi mentor harus registrasi akun dan validasi data terlebih dahulu. Kemudian mentor dapat membuat atau membuka kursus yang sesuai dengan keahliannya dan melakukan komunikasi untuk kesepakatan jadwal bersama siswa. Setelah siswa memesan mentor dan melakukan pembayaran, maka mentor menerima siswa yang membutuhkan bimbingannya untuk mempelajari pemrograman.

4. Class Diagram

Class diagram adalah salah satu jenis diagram dalam pemodelan sistem yang digunakan untuk menggambarkan struktur dan hubungan antar kelas dalam suatu sistem. *Class diagram* memberikan gambaran tingkat tinggi tentang komponen-komponen yang penting dalam sistem dan bagaimana mereka saling berhubungan [14]. *Class diagram* pada aplikasi *private coding* menggambarkan relasi hubungan di antara kelas-kelas, objek, atribut dan operasi pada aplikasi. Gambar 5 berikut ini adalah *class diagram* pada rancang bangun *Rest API* untuk mencari mentor pribadi.



Gambar 5. Class Diagram Rest API Private Coding

Pada gambar 5 terdapat 10 *class* yang memiliki atribut guna menjelaskan bagaimana kualitas yang baik dari suatu kelas tersebut, dan terdapat operasi yang terjadi pada masing-masing *class*. Adapun relasi *class* tersebut digambarkan dengan notasi *multiplicity*. Pada relasi antar *class* digambarkan dengan notasi *multiplicity*, *class students* berhubungan dengan *class* yang disebut sebagai *one-to-many*, yaitu *class students* dapat memiliki banyak relasi yang bersifat *dependent* atau memiliki ketergantungan satu dengan yang lainnya terhadap *class* yang ada. Hal ini

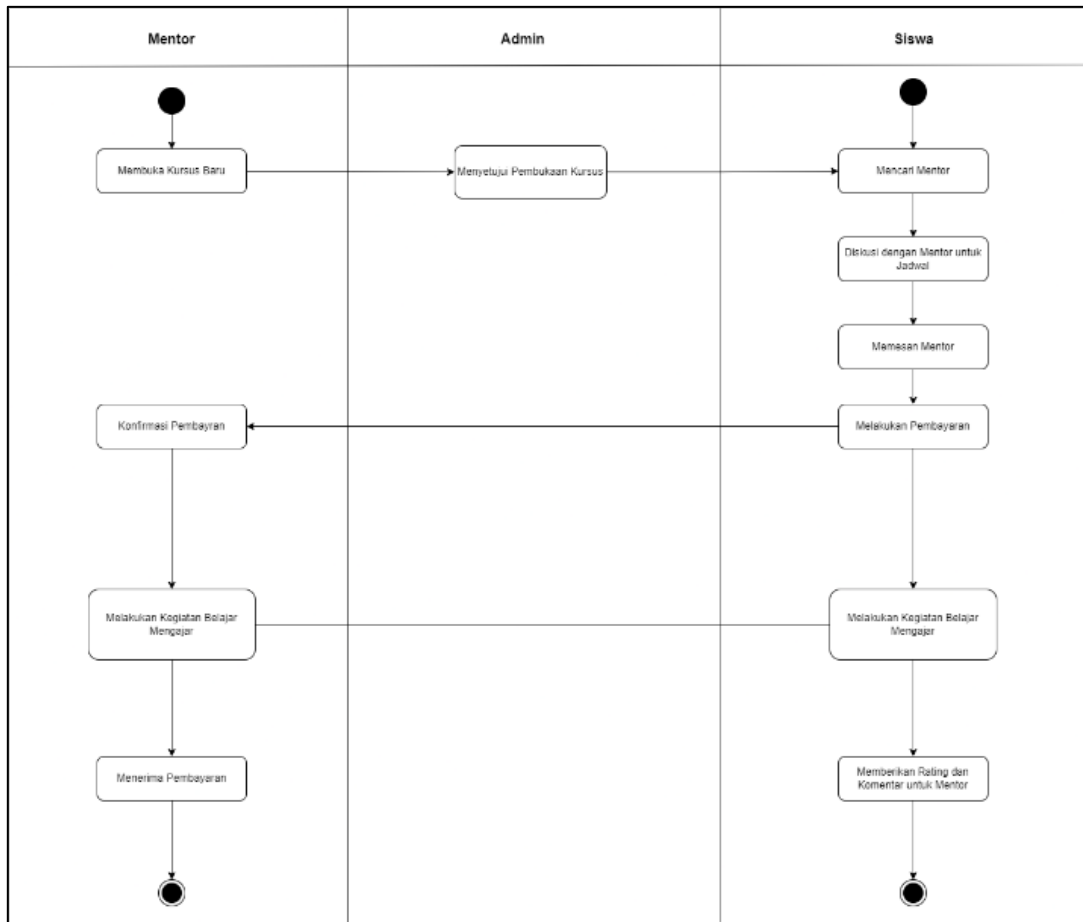
disebabkan karena *class* tersebut begitu membutuhkan data dari *class* yang tersedia untuk mengatur jadwal pembelajaran. Selanjutnya untuk *class* yang tidak memiliki relasi disebut *independent class* yang tidak tergantung pada *class* lain untuk operasi.

5. Activity Diagram

Activity diagram digunakan untuk menggambarkan tentang aktivitas yang terjadi pada sistem. *Activity diagram* menunjukkan langkah-langkah dalam proses kerja dari

sebuah sistem dari pertama sampai akhir [15]. *Activity diagram* juga akan menggambarkan konsep aliran data/kontrol, aksi terstruktur serta dirancang dengan baik

dalam suatu sistem [16]. Berikut ini *activity diagram* rancang bangun *Rest API* menggunakan *Express JS* untuk mencari mentor pribadi pada aplikasi *private coding*.



Gambar 6. Activity Diagram Rest API Private Coding

Pada gambar 6 dapat disimpulkan bahwa terdapat 3 alur proses yang terjadi dalam perancangan *Rest API aplikasi private coding* yaitu mentor, admin dan siswa. Alur aplikasi *private coding* ini, terjadi ketika mentor membuka kursus baru pada aplikasi, setelah itu admin melakukan persetujuan ke kursus yang ingin dibuka dan di saat bersamaan siswa dapat melakukan proses pencarian mentor. Selanjutnya, siswa melakukan diskusi pada aplikasi *private coding* bersama mentor untuk menentukan jadwal pembelajaran dan jika kedua pihak tepat maka siswa akan melakukan pemesanan mentor dan melakukan pembayaran. Setelah pembayaran berhasil, mentor melakukan konfirmasi pembayaran dan memulai kegiatan belajar dan mengajar bersama siswa sesuai dengan waktu yang ditentukan. Jika waktu pembelajaran berjalan, maka mentor akan menerima pembayaran dari siswa dan siswa dapat memberikan *rating* (penilaian) dan komentar sesuai dari pembelajaran yang telah diberikan oleh mentor.

3.2 Implementasi

Pada bagian ini akan membahas implementasi rancang bangun *Rest API* menggunakan *Express JS* guna mencari mentor pribadi dari aplikasi *private coding*. Dengan adanya *Rest API*, pengguna lain tidak dapat secara langsung mengakses *database*, melainkan harus *request* melalui internet, lalu setelah itu dapat mengakses *Rest API* dan kemudian *server Rest API* meresponsnya [17]. Sedangkan *Express JS* adalah satu *web framework* paling populer dan dokumentasinya yang lengkap dan penggunaannya yang cukup mudah, serta dapat membuat dan mengembangkan berbagai produk seperti aplikasi web ataupun *REST API* [18].

Berikut ini implementasi fitur-fitur *Rest API* pada aplikasi *private coding* diantaranya:

1. Fungsi Register Siswa

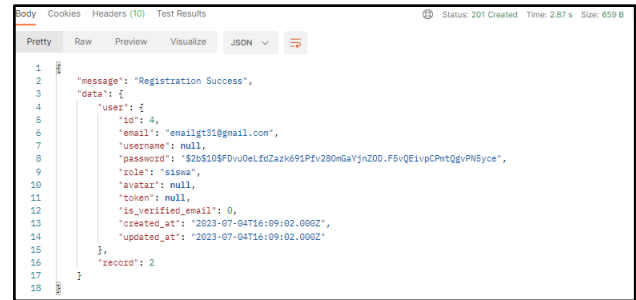
Dalam penerapan fungsi *Rest API* adapun tahapan register siswa yang dibuat untuk registrasi akun siswa. Berikut adalah tampilan register siswa yang dibuat:



Gambar 7. Registrasi Siswa

Pada gambar 7 dapat dijelaskan cara mendaftarkan dengan *role* siswa, *user* harus *merequest* ke `http://localhost:3000/register` dengan *method POST*. Lalu di dalam *body request*nya harus berisi suatu `{“email”, “password”, “role”, “first_name”, “last_name”, “education”, “gender”}`. Lalu jika berhasil akan

menghasilkan *response* seperti tampak pada gambar 8 berikut.



Gambar 8. Registrasi Sukses

Setelah itu, maka di dalam *database table users* akan terisi data-data yang di-*input*, seperti tampak pada gambar 9 dan di dalam *database table students* akan berisi data seperti tampak pada gambar 10 .

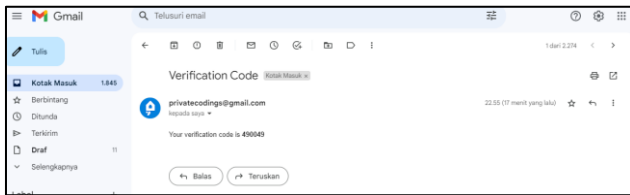
	id	email	username	password	role	avatar	token	is_verified_email	created_at	updated_at
	1	admin@gmail.com	admin	\$2b\$10\$im48pUUIQLSUs0i2lYqu8G35lv9E#6zgV\$svdKLW...	admin	NULL	NULL	1	2023-07-04 23:07:32	2023-07-04 23:07:32
	2	siswa@gmail.com	alwan	\$2b\$10\$9MCq8L2toCj5bHhCn8H5eIMbPWSSnyJ6PRWVidbmaB...	siswa	NULL	NULL	1	2023-07-04 23:07:32	2023-07-04 23:07:32
	3	mentor@gmail.com	zharonk	\$2b\$10\$HdZ6LUioWWYaN58UgDWaujU0Jum2iC4EgSMcXYXm6...	mentor	NULL	NULL	1	2023-07-04 23:07:32	2023-07-04 23:07:32
	4	emailgt31@gmail.com	NULL	\$2b\$10\$FDvuOeLfdZazk691Pfv28OmGaYjnZOD.F5vQEivPCPm...	siswa	NULL	NULL	0	2023-07-04 23:09:02	2023-07-04 23:09:02

Gambar 9. Data Database Table Users

	id	user_id	first_name	last_name	gender	education	created_at	updated_at
	1	2	Alwan	Dini	Perempuan	Perguruan Tinggi	2023-07-04 23:07:32	2023-07-04 23:07:32
	2	4	john	doe	pria	SD	2023-07-04 23:09:02	2023-07-04 23:09:02

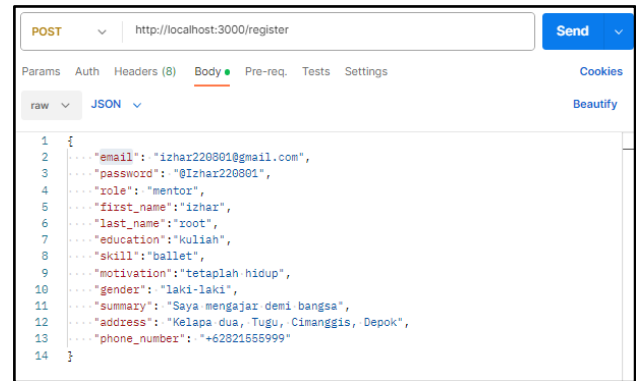
Gambar 10. Data Tabel Students

Kemudian, dari data tersebut akan dikirimkan 6 digit kode *One Time Password (OTP)* pada email yang telah terdaftar pada aplikasi *private coding*, 6 digit kode OTP ini untuk memverifikasikan email akun yang telah terdaftar, agar akun yang terdaftar pada aplikasi aktif dan digunakan, seperti tampak pada gambar 11.



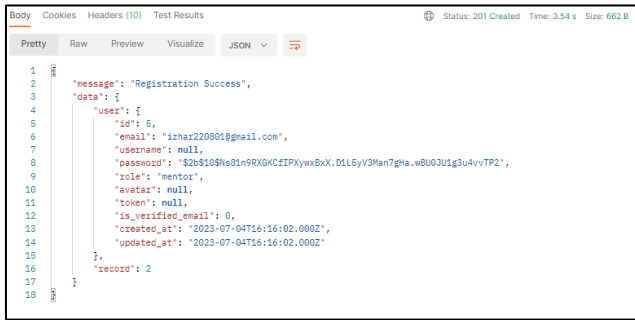
Gambar 11. Kode OTP via E-Mail

2. Fungsi Register Mentor



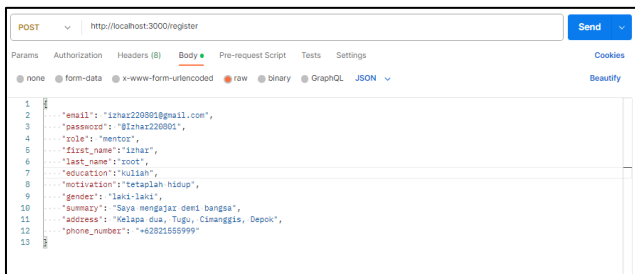
Gambar 12. JSON Register

Pada gambar 12, fungsi tersebut dapat berfungsi untuk mendaftarkan dengan *role* yang tersedia pada aplikasi seperti *role* mentor dan *user* harus melakukan *request* ke `http://localhost:3000/register` dengan *method POST*. Lalu di dalam *body request*nya harus berisi informasi terkait `{“email”, “password”, “role”, “first_name”, “last_name”, “education”, “gender”, “skill”, “motivation”, “summary”, “address”, “phone_number”}` agar informasi yang dihasilkan berhasil. Dan apabila berhasil maka akan menghasilkan *response* berhasil seperti tampak pada gambar 13 berikut ini.



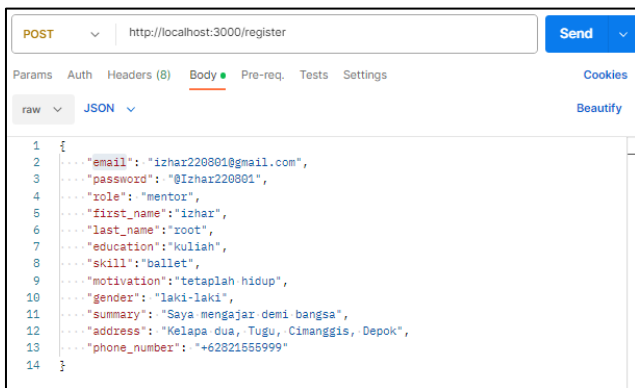
Gambar 13. Response Berhasil

Untuk dapat mendaftarkan dengan *role* mentor, *user* harus *me-request* ke `http://localhost:3000/register` dengan *method* *POST*. Lalu di dalam *body request*nya harus berisi informasi: {"email", "password", "role", "first_name", "last_name", "education", "gender", "skill", "motivation", "summary", "address", "phone_number"}. Pada gambar 14 merupakan tampilan *invalid register* jika *user* ada yang tidak valid dalam melakukan registrasi:



Gambar 14. Invalid Register

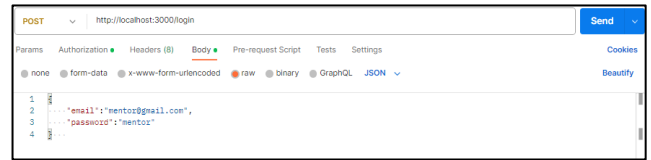
User juga dapat melakukan registrasi dengan *email* yang sudah terdaftar dengan hasil seperti tampak pada gambar 15.



Gambar 15. Registrasi dengan Email

3. Fungsi Login

Untuk dapat *login*, *user* harus *me-request* ke `http://localhost:3000/login` dengan *method* *POST*. Lalu di dalam *body request*nya harus berisi: {"email", "password"}, seperti tampak di gambar 16.



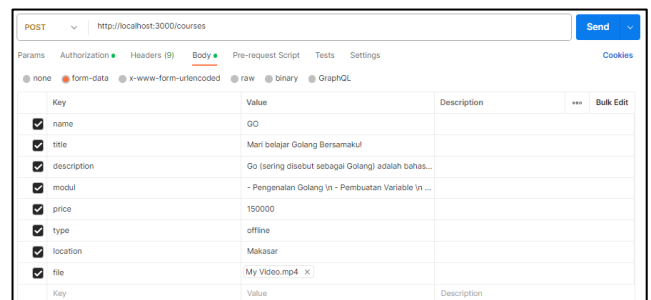
Gambar 16. Login

Setelah *login user* akan mendapatkan response {"token", "refreshToken"}. *Token* disini yang nantinya akan dimasukan ke dalam *Bearer Token* setiap kali *user* ingin *me-request* ke *API* tertentu, supaya aplikasi mengenali bahwa yang meminta adalah *user* dan *token* yang aktif selama 900detik/15menit.

Lalu untuk *refresh token*, *user* diminta kembali *token*nya. Saat *token*nya masa aktif *token* sudah habis dan kembali *refresh token*, maka masa aktifnya akan ditambah selama 86400detik/24jam.

4. Verifikasi Email Akun

Melakukan *request* dengan akun yang belum ter-*verifikasi*. Seperti yang sudah dibahas pada fungsi *register*, bahwa akun yang mendaftar harus untuk *memverifikasi* email akunnnya, supaya akunnnya dapat aktif.



Gambar 17. Request dengan Akun

Pada gambar 17 menggambarkan bahwa *user* mencoba untuk melakukan *request* akses ke `http://localhost:3000/courses` dengan email yang belum *terverifikasi* pada aplikasi. Oleh karena itu, *user* tersebut akan mendapatkan *response request* akun seperti gambar 18.

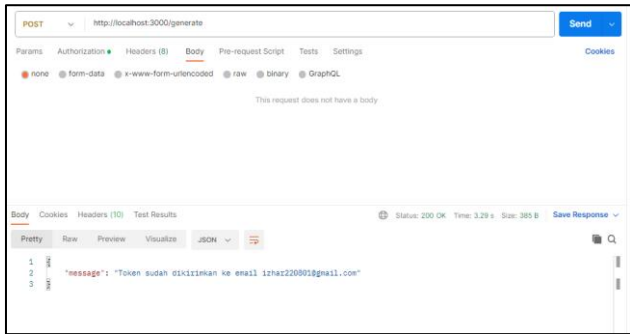


Gambar 18. Response Request Akun

Pada gambar 18 berisi informasi *response request* akun pada aplikasi yang telah dilakukan, maka kemudian akan menampilkan *message* sudah atau belum *terverifikasi*.

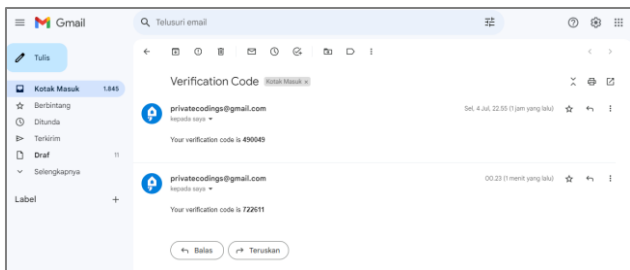
Untuk melakukan *verifikasi* email akun, *user* harus *me-request* ke `http://localhost:3000/verif` dengan *method* *POST*. Lalu di dalam *body request*-nya harus berisi : {"otp"} Untuk *OTP* akan dikirimkan ke *email* dan bisa didapatkan saat *user* berhasil mendaftarkan akun atau *user*

bisa *merequest* ke <http://localhost:3000/generate> dengan *method POST*. Pada gambar 19 merupakan contoh *user* dengan email “*izhar220801@gmail.com*” *me-request* ke <http://localhost:3000/generate> untuk mendapatkan OTP:



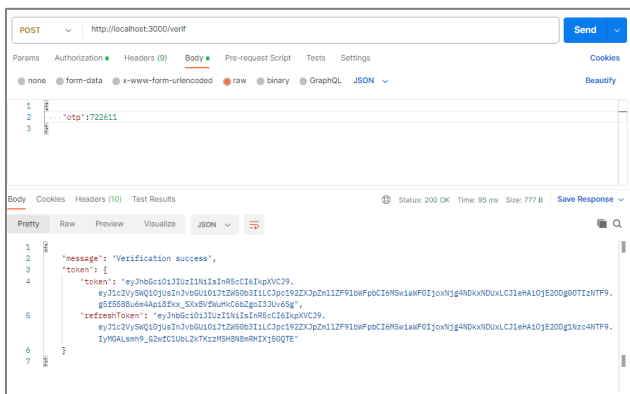
Gambar 19. Verifikasi Email

Setelah itu OTP baru akan dikirimkan ke *email* yang sudah dimasukkan sebelumnya seperti tampak pada gambar 20.



Gambar 20. OTP menerima Email

Setelah itu *user* dapat memasukkan OTP ke dalam *body request* {"otp"}. Jika OTP benar maka akan ditampilkan informasi seperti gambar 21 berikut ini.



Gambar 21. Informasi Verifikasi OTP

3.3 Hasil Pengujian

1. Blackbox Testing

Pengujian *blackbox* memiliki 27 pengujian yang dilakukan oleh *developer*. Jumlah pengujian yang berhasil didapatkan sebanyak 27 keberhasilan, sehingga persentase keberhasilannya yaitu:

- Persentase keberhasilan = (Jumlah skenario pengujian berhasil / Jumlah total skenario pengujian) x 100%
- Persentase keberhasilan = (27 / 27) x 100% = 100%
- Jadi, dapat disimpulkan bahwa sistem telah berhasil menyelesaikan skenario pengujian.

2. Kuesioner

Berikut ini adalah hasil kuesioner yang didapatkan dari responden, dengan hasil kuesioner (*skala likert*) berikut ini:

- Jawaban Sangat Setuju (SS) = 20 x 4 = 80
- Jawaban Setuju (S) = 16 x 3 = 48
- Jawaban Tidak Setuju (TS) = 0 x 2 = 0
- Jawaban Sangat Tidak Setuju (STS) = 0 x 1 = 0
- Total Skor = 128

Rumus nilai maksimum = (Jumlah responden * Nilai tertinggi *likert* * Total pernyataan) Nilai maksimum = 4 * 4 * 9 = 144
 Penyelesaian Akhir Indeks (%) = (total skor / skor max) * 100 indeks (%) = (128 / 144) * 100 = 88,8%.

Hasil tes *skala likert* yang mencapai 88,8% , artinya masuk dalam interval 80-100 dan menunjukkan bahwa responden sangat setuju terhadap topik yang diuji.

4. KESIMPULAN

Berdasarkan hasil pengujian rancang bangun *Rest API* menggunakan *Express JS* guna mencari mentor pribadi untuk pembelajaran maka didapatkan kesimpulan berikut:

- Rancangan dan pengembangan *Rest API private coding* berjalan dengan baik dan efisien, karena terfokus pada penggunaan teknologi *Express JS* sebagai *backend* aplikasi *private coding*.
- Implementasi fungsi pencarian mentor pribadi pada aplikasi *private coding* telah berfungsi dengan baik sehingga pengguna yang ingin mencari mentor pemrograman menjadi lebih mudah. Hal tersebut sesuai dengan rumus nilai maksimum perhitungan *skala likert* yang telah dilakukan oleh peneliti terhadap fungsionalitas aplikasi. Hasil pengujian yang dilakukan untuk fungsi aplikasi memperoleh persentase hasil keberhasilan mencapai 88,8% dengan interval 80-100 yang menunjukkan bahwa responden sangat setuju terhadap topik penelitian. Dan melalui hasil analisis pengujian *skala likert* yang dilakukan, mayoritas pengguna menyatakan bahwa aplikasi *private coding* dapat mempermudah pencarian mentor pemrograman.

Adapun rancang bangun *Rest API* menggunakan *Express JS* guna mencari mentor pribadi untuk pembelajaran masih memiliki beberapa kekurangan, sehingga perlu pengembangan lanjutan. Berikut adalah masukan terkait dengan pengembangan aplikasi *private coding* untuk penelitian yang akan datang, diantaranya:

- Saat ini aplikasi *private coding* masih dalam tahap pengembangan sehingga fungsi keamanan aplikasi belum sepenuhnya baik. Atas dasar tersebut, peneliti

berharap tim dapat memperbaiki fitur dan layanan aplikasi termasuk dengan keamanannya.

2. Saat ini aplikasi *private coding* masih belum memiliki fitur yang dapat merekomendasikan mentor dengan melihat peminatan siswa yang mendaftar. Sehingga dengan demikian peneliti berharap dengan adanya fitur tersebut bisa membantu siswa dalam mendapatkan mentor yang sesuai.

Ucapan Terima Kasih

Ucapan terima kasih diberikan kepada saudara Achmad Izhar, mahasiswa STT Terpadu Nurul Fikri yang telah merancang dan membangun *Rest API* aplikasi *private coding* guna mencari mentor pribadi. Penghargaan yang setinggi-tingginya juga diberikan kepada tim LPPM STT Terpadu Nurul Fikri yang telah memfasilitasi penerbitan jurnal ini.

DAFTAR PUSTAKA

- [1] B. Arif, "Miris, 400 Ribu Sarjana TI Per Tahun Tak Memenuhi Kualifikasi Industri." <https://edukasi.sindonews.com/read/387390/211/miris-400-ribu-sarjana-ti-per-tahun-tak-memenuhi-kualifikasi-industri-1617620642?showpage=all> (accessed Sep. 14, 2022).
- [2] Narenda, "Miris, 400 Ribu Sarjana TI Per Tahun Tak Memenuhi Kualifikasi Industri," *CEO Dicoding*, 2021. <https://edukasi.sindonews.com/read/387390/211/miris-400-ribu-sarjana-ti-per-tahun-tak-memenuhi-kualifikasi-industri-1617620642?showpage=all> (accessed Sep. 14, 2022).
- [3] M. Gilang, "Aplikasi Berbasis Web dan Keunggulannya yang Bisa Anda Ketahui," 2021. <https://www.qubisa.com/article/pengertian-mentoring-mentor-dan-mentee> (accessed Sep. 25, 2022).
- [4] P. Agriza, N. Azizah, M. Asfi, and I. Syafrinal, "Implementasi Model Scrum Pada Sistem Informasi Pembelajaran Diluar Kampus Untuk Skema Wirausaha Kampus Merdeka," *Syntax J. Inf.*, vol. 10, n, pp. 1–12, 2021.
- [5] Z. Imaduddin, H. Saptono, S. T. Fauziah, H. A. Tawakal, and D. Hamzah, "Aplikasi Monitoring Perkembangan Janin (Antenatal Care) dengan Metode Scrum Berbasis Perangkat Mobile," *J. Teknol. Terpadu*, vol. Vol. 5, No, 2019.
- [6] R. W. Imam Haromain, Sirojul Munir, "Sistem Informasi Geografis Pencarian Layanan Vaksin dan PCR Covid-19 Menggunakan Google Maps API dan Jalur Terpendek," *J. Teknol. Terpadu*, vol. Vol. 8, No, pp. 86–93, 2022.
- [7] F. Nadhira, M. I. Wahyuddin, and R. T. K. Sari, "Penerapan Metode Agile Scrum Pada Rancangan SisIAM4," *J. Media Inform. Budidarma*, vol. 6, no. 1, p. 560, 2022, doi: 10.30865/mib.v6i1.3525.
- [8] S. N. Rafianto, Dimas, "Penerapan Metode Scrum Pada Pembuatan User Experience Landing Page Sistem Informasi Lentera," *J. Sist. Inf. dan Sains Teknol*, vol. 3, no, pp. 1–14, 2021.
- [9] R. W. Sirojul Munir, Ahmad Rio Adriansyah, Radon Dhelika, "Implementasi Arsitektur Aplikasi MVC pada Perancangan Aplikasi Bank Sampah Berbasis Web," *J. Teknol. Terpadu*, vol. Vol. 4, No, 2018.
- [10] D. A. P. Ammar Musthofa Firdaus, "Aplikasi Pencari Tempat Magang Berbasis Android Menggunakan Metode Agile Scrum," *J. Inform. UPGRIS*, vol. Vo. 8, No., 2022.
- [11] S. T. F. Sirojul Munir, Anifatul Aufah, Kuati Septiani, "Analisis dan Rancang Bangun Prototype Web Market Place UMKM Juara," *J. Teknol. Terpadu*, vol. Vol. 6, No, pp. 66–71, 2020.
- [12] L. Setiyani, "Desain Sistem: Use Case Diagram Pendahuluan," in *Prosiding Seminar Nasional: Inovasi & Adopsi Teknologi*, 2021, pp. 246–260.
- [13] B. A. Rahmatul Zakia, Tifanny Nabarian, "Rancang Bangun Antarmuka Berbasis Website Design Method(Wdm) untuk Toko Baju Online," *J. Inform.*, vol. Vol. 9No., pp. 24–33, 2023.
- [14] B. A. H. Muchamad Hasan Basri, "Perancangan Sistem Informasi Kepegawaian Berbasis Web Pada Suara Merdeka Network," 2023.
- [15] N. W. A.-H. Abdul Aziz Kamil, "SISTEM INFORMASI AKADEMIK SMPN 3 TELUK KUANTAN," *J. Perencanaan, Sains, Teknol. dan Komput.*, vol. Vol. 6, No, pp. 79–87, 2023.
- [16] M. W. Tia Arianti, Amal Fa'izi, Safri Adam, "Perancangan Sistem Informasi Perpustakaan Menggunakan Diagram UML (Unified Modelling Language)," *J. Ilm. Komput. Terap. dan Inf.*, vol. Vol. 1, No, 2022.
- [17] Baharudin, "Implementasi Web Service dengan Metode Rest API untuk Integrasi Data Covid 19 di Sulawesi Selatan," *J. Sintaks Log.*, vol. Vol. 2, No, 2022.
- [18] A. Firdaus, "Rancang Bangun Sistem Informasi Perpustakaan menggunakan Web Service pada Jurusan Teknik Komputer Polstri," *J. Informanika*, vol. Vol.5, No., 2019.