

Pengembangan *Back-End* Modul Manajemen Kursus pada *Website* Latihan TOEFL ITP di Telkom University Language Center (LaC) dengan Metode *Iterative Incremental* Menggunakan Laravel

1st Sufiana Arumdita
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia

sufianaardt@student.telkomuniversity.ac.id

2nd Faishal Mufied Al Anshary, S.Kom.,
M.Kom., M.Sc.

Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia

faishalmufied@telkomuniversity.ac.id

3rd Dr. Ilham Perdana, S.T., M.T.
Fakultas Rekayasa Industri

Universitas Telkom
Bandung, Indonesia

ilhamp@telkomuniversity.ac.id

Abstrak—Bahasa Inggris merupakan keterampilan krusial di era global saat ini, dengan TOEFL ITP sebagai salah satu indikator kompetensi yang diakui luas. Telkom University Language Center (LaC) menyelenggarakan kursus TOEFL ITP, namun menghadapi tantangan signifikan. Ini mencakup keterbatasan akses pendaftaran, proses pendaftaran hingga pembayaran yang masih manual dan rentan kesalahan terlebih ketika jumlah peserta meningkat drastis karena proses yang masih belum terintegrasi serta tidak terintegrasi dengan pembelajaran mandiri. Kondisi ini menghambat efisiensi operasional. Oleh karena itu, penelitian ini bertujuan merancang, mengembangkan modul manajemen kursus pada website Latihan TOEFL ITP, dan melakukan pengujian untuk mengatasi permasalahan pada LaC tersebut. Pengumpulan data dilakukan dengan mewawancarai stakeholder di LaC. Metode pengembangan yang digunakan adalah *iterative incremental* dengan dua iterasi, yang mencakup tahapan *planning*, *design* yang menghasilkan artefak berupa diagram UML, *development* menggunakan framework Laravel berbasis MVC dan basis data MySQL yang berfokus pada pengembangan backend, *testing* fungsionalitas backend yang dilakukan dengan metode *unit testing* dan *blackbox testing* dengan menguji respons API menggunakan tool Postman, serta evaluasi. Hasil pengujian API pada iterasi kedua menunjukkan bahwa seluruh endpoint telah berfungsi sesuai kebutuhan sistem termasuk perbaikan dari iterasi sebelumnya diharapkan sistem backend memberikan dukungan teknis yang esensial terhadap frontend modul manajemen kursus.

Kata kunci—Laravel, API, *iterative incremental*, manajemen kursus

I. PENDAHULUAN

Bahasa Inggris merupakan bahasa internasional yang berperan penting dalam komunikasi global di berbagai bidang, mulai dari bisnis hingga pendidikan. Berdasarkan laporan Ethnologue tahun 2024, jumlah penutur bahasa Inggris di seluruh dunia mencapai 1,5 miliar orang, menjadikannya sebagai bahasa yang paling banyak digunakan secara global, baik sebagai bahasa ibu maupun sebagai bahasa kedua [1]. Dalam dunia kerja dan pendidikan tinggi, kemampuan berbahasa Inggris tidak hanya sekadar nilai lebih, tetapi menjadi salah satu syarat utama untuk

berkembang. Penelitian menunjukkan bahwa penguasaan bahasa Inggris dapat meningkatkan kinerja karyawan secara signifikan [2]. Salah satu cara untuk membuktikan kompetensi ini adalah melalui sertifikasi seperti TOEFL ITP. Tes ini telah diakui oleh lebih dari 2.500 institusi di lebih dari 50 negara [3], dan sering dijadikan syarat dalam seleksi beasiswa nasional maupun internasional seperti LPDP, Fulbright, dan MEXT [4]. Dalam perspektif modal manusia menurut Becker [5], skor TOEFL ITP mencerminkan investasi dalam kompetensi individu yang dapat membuka berbagai peluang karier dan pendidikan.

Telkom University Language Center (LaC), menyelenggarakan kursus TOEFL ITP secara *blended learning*. Namun, proses pendaftaran dan manajemen peserta saat ini masih manual, mengandalkan Google Form terpisah, email belajar.id khusus, dan konfirmasi WhatsApp. Metode ini menciptakan keterbatasan akses karena periode pendaftaran yang triwulanan dan keharusan penggunaan email belajar.id Telkom University untuk akses LMS, membatasi partisipasi masyarakat umum dan menjadikan proses kurang fleksibel. Selain itu, kurangnya efisiensi dan akurasi menjadi isu krusial. Penggunaan platform eksternal berupa Google Form yang terpisah untuk pendaftaran dan unggah bukti pembayaran menyebabkan pemindahan data manual yang rentan *human error*. Ketergantungan pada konfirmasi manual via WhatsApp juga dinilai kurang efisien dalam proses pembayaran. Selain itu, tidak adanya integrasi dengan sistem pembelajaran mandiri menimbulkan kebutuhan untuk membuat proses pendaftaran hingga pembayaran terintegrasi dengan sistem pembelajaran mandiri. Permasalahan ini semakin krusial dengan adanya lonjakan jumlah peserta yang signifikan dari 6 pada 2023 menjadi 80 pada 2024, dan 47 dalam empat bulan pertama 2025, meningkatkan beban administratif dan potensi *human error*.

Oleh karena itu, penelitian ini bertujuan untuk merancang dan mengembangkan sistem terintegrasi yang mampu mengelola seluruh proses manajemen kursus yang berfokus pada proses persiapan pembelajaran kursus

sebelum proses pembelajaran kursus TOEFL ITP di LaC dimulai yang mencakup proses pendaftaran hingga pembayaran, serta melakukan pengujian untuk memastikan kesesuaian terhadap kebutuhan pengguna. Sistem ini diharapkan dapat mengatasi keterbatasan yang ada, inefisiensi, mencegah *human error*, dan memfasilitasi integrasi dengan pembelajaran mandiri sehingga diharapkan mampu meningkatkan efisiensi operasional LaC.

II. KAJIAN TEORI

A. TOEFL ITP

Berdasarkan ETS (Educational Testing Service), TOEFL ITP (Institutional Testing Program) adalah alat penilaian kemampuan bahasa Inggris yang diakui secara global, dirancang untuk mengukur keterampilan bahasa Inggris siswa dalam konteks akademik. Level kecakapan TOEFL ITP terdiri dari beberapa tingkatan yaitu Level C1 (mahir) memiliki skor minimal 620, dengan skor minimal 62 untuk listening comprehension, 64 untuk structure and written expression, dan 60 untuk reading comprehension. Level B2 (menengah atas) memiliki skor minimal 543, dengan skor minimal 55 untuk listening comprehension, 53 untuk structure and written expression, dan 55 untuk reading comprehension. Level B1 (menengah bawah) memiliki skor minimal 433, dengan skor minimal 46 untuk listening comprehension, 43 untuk structure and written expression, dan 41 untuk reading comprehension. Level A2 (dasar atas) memiliki skor minimal 343, dengan skor minimal 38 untuk listening comprehension, 32 untuk structure and written expression, dan 33 untuk reading comprehension [3].

B. Back-End

Back-End adalah komponen dari sistem informasi atau aplikasi yang bertanggung jawab terhadap jalannya berbagai proses di balik layar, seperti menambahkan, mengubah, dan menghapus data. Umumnya, bagian ini mengelola aspek teknis yang tidak langsung diakses oleh pengguna, termasuk pengaturan server dan manajemen basis data [6].

Back-end mencakup tahapan *evaluative research* dan *post-design*. Tahap *evaluative research* bertujuan untuk mengevaluasi sejauh mana desain yang dibuat efektif serta menemukan kendala yang muncul selama proses pengembangan. Sementara itu, tahap *post-design* berfokus pada analisis terhadap cara pengguna berinteraksi dengan produk dan bagaimana pengalaman mereka saat menggunakannya dalam situasi nyata [7].

C. REST API

Back-End REST (Representational State Transfer) adalah sebuah arsitektur yang digunakan untuk merancang layanan yang dapat diakses lintas platform dan lingkungan sistem. RESTful API merupakan implementasi dari konsep ini, di mana layanan disediakan melalui berbagai endpoint yang merepresentasikan fungsi bisnis spesifik biasanya menggunakan protokol HTTP dengan metode yang standar seperti *GET*, *POST*, *PUT*, dan *DELETE*. Untuk menjamin konsistensi dan keandalan dokumentasi, RESTful API umumnya menggunakan format yang terstandar seperti OpenAPI *Specification*, JSON, atau XML [8].

D. Situs Web

Situs Web adalah sekumpulan halaman yang tersedia secara *online* dan dapat diakses melalui alamat URL. Secara umum, situs web dibedakan menjadi dua jenis, yaitu statis dan dinamis. Situs web statis menyajikan informasi yang tidak mengalami perubahan, sedangkan situs web dinamis memungkinkan konten di dalamnya diperbarui sesuai kebutuhan [9].

E. Kerangka Kerja Laravel

Berdasarkan Laravel memiliki fitur lengkap seperti routing fleksibel, autentikasi bawaan, dan migrasi database yang mudah. Keunggulan yang ada pada Eloquent ORM memudahkan pengelolaan database dengan sintaks PHP. Selain itu, Blade membantu membuat tampilan lebih efisien, dan fitur lain seperti notifikasi, antrian, dan caching turut mendukung kinerja aplikasi [10].

F. Iterative Incremental

Metode *iterative incremental* memungkinkan pengembangan perangkat lunak secara bertahap. Setiap iterasi menghasilkan bagian dari perangkat lunak yang bisa berfungsi sehingga pengguna diberikan kesempatan untuk dapat memberikan masukan atau umpan balik lebih awal [11].

Proses dalam *iterative incremental* terdiri dari beberapa tahapan seperti *planning*, *Design*, *Implementation/Development*, *Testing*, dan *Evaluation*. Meskipun tahapan dalam metode ini dijalankan berurutan, keseluruhan prosesnya dibuat untuk meningkatkan fleksibilitas dan menyesuaikan perubahan selama masa pengembangan [12].

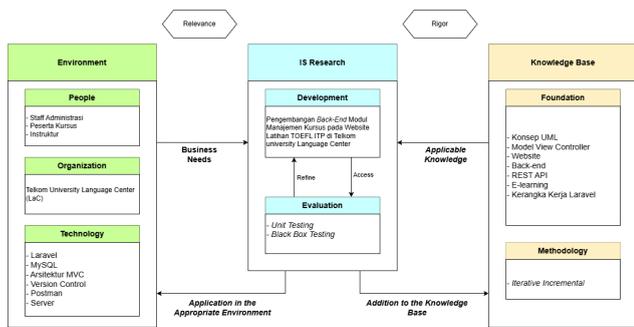
G. UML

Unified Modelling Languages (UML) merupakan bahasa yang digunakan untuk memvisualisasikan gambaran maupun spesifikasi dalam pengembangan dan dokumentasi dari pengembangan sistem yang berorientasi objek berupa diagram atau gambar. UML terdiri dari kumpulan diagram yang menggambarkan masalah dan solusinya. UML menyediakan standar untuk membuat cetak biru sistem meliputi konsep proses bisnis, pembuatan *class diagram*, *activity diagram*, *use case*, perancangan basis data, dan berbagai komponen yang diperlukan dalam pengembangan sistem [13]. Pada penelitian ini digunakan beberapa UML yaitu *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.

III. METODE

A. Model Konseptual

Model konseptual yang digunakan dalam membantu memahami dan mengevaluasi penelitian dalam sistem informasi dilakukan dengan menghubungkan dua paradigma, yaitu *behavioral science* dan *design science*. *Behavioral science* berfokus pada pengembangan dan verifikasi teori yang menjabarkan dan memprediksi perilaku manusia dan organisasi. *Design science* berfokus pada penciptaan artefak teknologi yang inovatif guna memecahkan masalah organisasi [14].

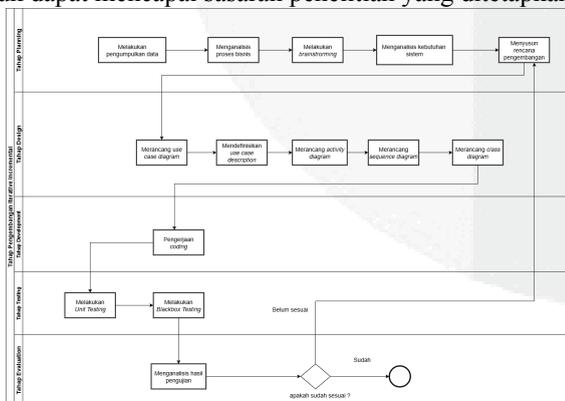


Gambar III. 1 Model Konseptual

Gambar III. 1 menjelaskan bahwa model konseptual terdiri dari tiga elemen, yaitu Lingkungan, Penelitian SI, dan Dasar Ilmu. Elemen Lingkungan mencakup aspek *people*, *organization*, dan *technology*. Aspek *people* berfokus pada staff administrasi, peserta kursus, dan instruktur LaC sebagai pengguna aplikasi. Aspek *organization* merujuk pada Telkom University Language Center (LaC) sebagai entitas yang membutuhkan solusi. Sementara itu, aspek *technology* menguraikan penggunaan Laravel, MySQL, arsitektur MVC, version control, Postman, dan server yang digunakan. Selanjutnya, elemen Penelitian SI menjelaskan kegiatan inti pengembangan, yaitu pengembangan *backend* modul manajemen kursus pada website latihan TOEFL ITP di Telkom University Language Center. Pengujian *backend* dilakukan menggunakan metode *unit testing* dan metode *blackbox testing* melalui Postman yang secara spesifik menguji respons dan fungsionalitas setiap endpoint API untuk memastikan kesesuaian dengan kebutuhan yang telah ditetapkan. Terakhir, elemen Dasar Ilmu (*Knowledge Base*) terdiri dari aspek *foundation* dan *methodology*. Aspek *foundation* mencakup konsep-konsep dasar seperti UML, MVC, *website*, *backend*, REST API, e-learning, dan framework Laravel. Sementara pada aspek *methodology*, penelitian ini menerapkan metode *iterative incremental*, yang dipilih karena fleksibilitasnya dalam mengakomodasi fitur yang berkembang dan kebutuhan penyesuaian berkelanjutan berdasarkan evaluasi.

B. Sistematisasi Penyelesaian Masalah

Sistematisasi penyelesaian masalah merupakan tahapan atau tata cara yang digunakan dalam merancang penelitian untuk dapat mencapai sasaran penelitian yang ditetapkan.



Gambar III. 2 Sistematisasi Penyelesaian Masalah

Penelitian ini menerapkan metode *iterative incremental*. Gambar III.2 menggambarkan sistematisasi penyelesaian

masalah pada bagian pengembangan yang terdiri dari lima tahap sebagai berikut:

1. Tahap *Planning*

Pada tahap ini, data dikumpulkan melalui wawancara dengan pihak-pihak yang terlibat dalam pengelolaan kursus TOEFL ITP di LaC, seperti administrasi, instruktur, peserta, dan Kepala Pusat Bahasa. Hasil wawancara digunakan untuk menganalisis proses bisnis saat ini, mengidentifikasi masalah, serta kesenjangan antara kondisi yang ada dan kebutuhan yang diharapkan. Selanjutnya, dilakukan sesi *brainstorming* dengan tim untuk merumuskan solusi berdasarkan permasalahan yang ditemukan, dengan pendekatan kreatif dan berfokus pada kebutuhan pengguna. Ide-ide yang terkumpul kemudian diseleksi berdasarkan urgensi dan relevansinya. Dari ide yang disepakati, dilakukan analisis kebutuhan sistem yang mencakup aspek fungsional dan non-fungsional, sekaligus menentukan rencana pengembangan fitur dan batasan pengembangan yang dilakukan pada tahap awal iterasi.

2. Tahap *Design*

Pada tahap desain, perancangan fungsionalitas sistem yang dilakukan mencakup cara kerja dan komponen yang akan dikembangkan. Proses ini melibatkan penyusunan diagram UML seperti *Use Case*, *Activity*, *Sequence*, dan *Class Diagram* untuk memvisualisasikan alur atau tahapan dan fungsionalitas sistem secara rinci. Penyusunan artefak ini bertujuan untuk mempermudah proses pengembangan sistem secara terstruktur dan sesuai dengan kebutuhan.

3. Tahap *Development*

Pada tahap *development*, dilakukan implementasi kode untuk mengembangkan aplikasi sesuai dengan rancangan kebutuhan sistem. Pengembangan difokuskan pada *back-end* modul manajemen kursus untuk website latihan TOEFL ITP, menggunakan framework Laravel dengan penerapan konsep Model View Controller (MVC) dan basis data MySQL.

4. Tahap *Testing*

Pada tahap *testing*, aplikasi yang telah dikembangkan diuji untuk memastikan kesesuaiannya dengan rancangan awal. Jika hasilnya sesuai, tahap ini dianggap selesai; namun jika perlu penyesuaian, dilakukan perbaikan terlebih dahulu. Pengujian dilakukan menggunakan metode *unit testing* dengan menguji komponen atau fungsi yang digunakan pada kode internal program yang memanfaatkan PHPUnit dan *blackbox testing* dengan fokus menguji respons API menggunakan Postman untuk memastikan fungsionalitas berjalan sesuai kebutuhan.

5. Tahap *Evaluation*

Pada tahap evaluasi, fitur-fitur yang telah lolos pengujian sebelumnya ditinjau kembali untuk menilai mana yang perlu ditingkatkan atau diperbaiki. Sementara, proses *iterative incremental* akan terus berulang hingga seluruh kebutuhan pengguna terpenuhi dan seluruh fitur dapat digunakan sesuai dengan tujuan yang ditetapkan.

C. Pengumpulan Data

Pada proses ini dilakukan wawancara sebagai metode pengumpulan data. Wawancara dilakukan terhadap staff administrasi, peserta kursus, dan instruktur dari kursus TOEFL ITP di LaC untuk menggali kondisi saat ini, kebutuhan, dan permasalahan yang dihadapi. Selain itu, penelitian ini melakukan pengujian dengan metode *unit testing* menggunakan PHPUnit dan metode *blackbox testing* dengan fokus menguji respons API.

IV. PENYELESAIAN PERMASALAHAN

A. Analisis Kebutuhan Pengguna

Dalam menentukan kebutuhan pengguna dilakukan analisis kesenjangan untuk mengidentifikasi solusi dan kebutuhan sesuai kondisi saat ini dan diharapkan dapat berubah sesuai kondisi yang diharapkan. Berikut merupakan analisis kesenjangan yang dilakukan terhadap permasalahan proses pembayaran saat ini yang masih dilakukan melalui google form untuk mengunggah bukti pembayaran. Proses pembayaran tersebut tidak terintegrasi dalam satu sistem dengan proses pendaftaran sehingga membuat proses verifikasi hingga konfirmasi masih dilakukan secara manual oleh staff administrasi dan dinilai kurang efisien.

Tabel IV. 1 Analisis Kesenjangan

Kondisi saat ini	N	P	F	Kebutuhan	Akar masalah
Staff administrasi memverifikasi bukti pembayaran secara manual dan mengirimkan konfirmasi pembayaran satu per satu melalui WhatsApp.	v			Membutuhkan sistem yang memfasilitasi verifikasi bukti pembayaran yang lebih efisien, terintegrasi, dan aman, serta konfirmasi pembayaran yang efisien.	Belum tersedianya sistem yang memfasilitasi verifikasi bukti pembayaran dan mengirimkan konfirmasi pembayaran secara efisien

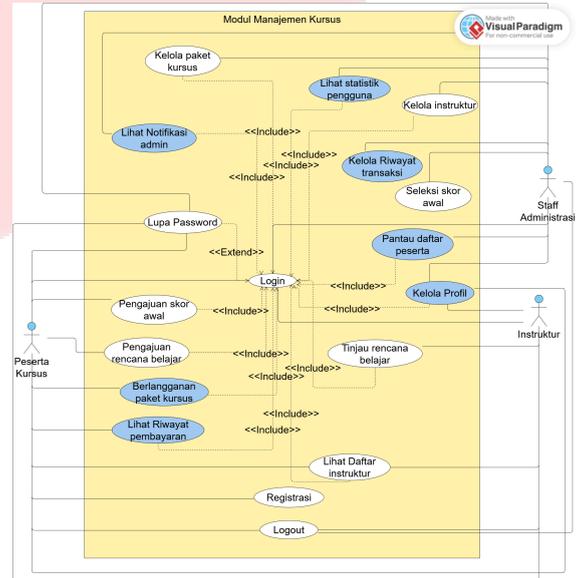
Sementara kebutuhan sistem ditentukan berdasarkan hasil *brainstorming* dengan tim untuk merumuskan solusi berdasarkan permasalahan yang ditemukan, dengan pendekatan kreatif dan berfokus pada kebutuhan pengguna. Kemudian, kebutuhan yang telah ditentukan menjadi dasar untuk perencanaan pengembangan yang dilakukan sebanyak dua iterasi.

B. Perancangan Sistem

Perancangan sistem dilakukan berdasarkan kebutuhan yang telah didefinisikan. Rancangan dibuat berdasarkan standar UML (*Unified Modelling Language*).

1. Use Case Diagram

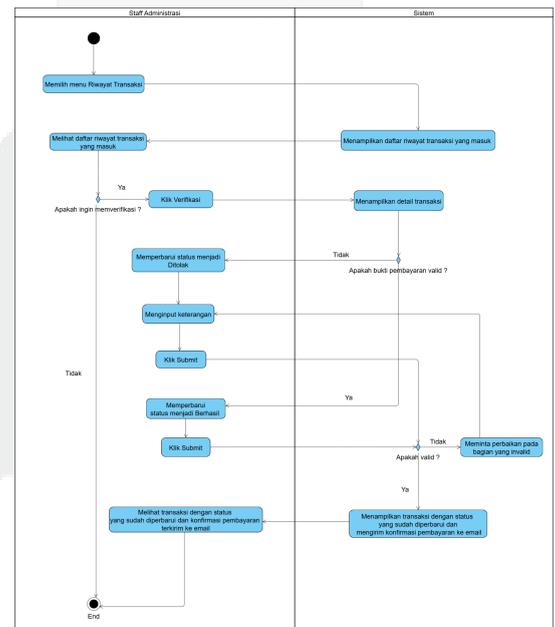
Berikut merupakan use case diagram yang menunjukkan alur fungsionalitas sistem. *Use case* berwarna putih menunjukkan fungsionalitas yang dikembangkan pada iterasi pertama, sementara yang berwarna biru menunjukkan fungsionalitas yang dikembangkan pada iterasi kedua.



Gambar IV. 1 Use Case Diagram

2. Activity Diagram

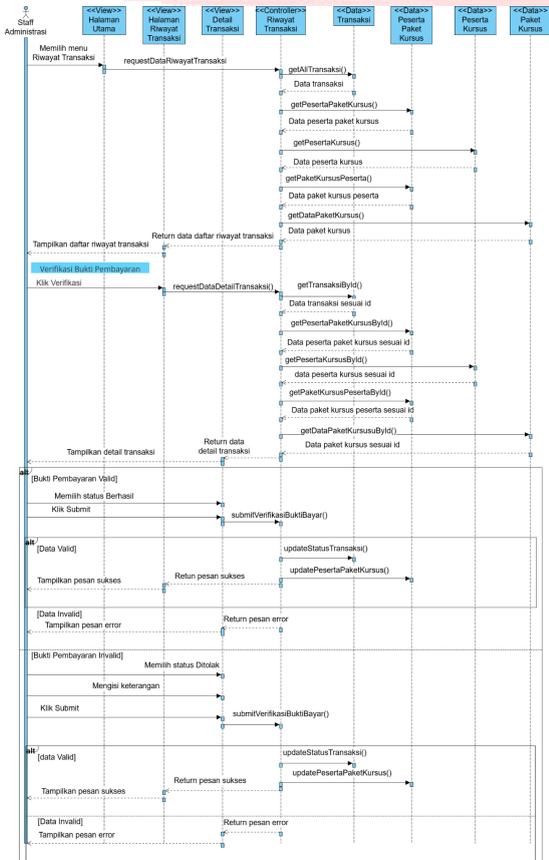
Berikut merupakan activity diagram yang menunjukkan alur aktivitas antara aktor staff administrasi dengan sistem pada *use case* kelola riwayat transaksi.



Gambar IV. 2 Activity Diagram Kelola Riwayat Transaksi

3. Sequence Diagram

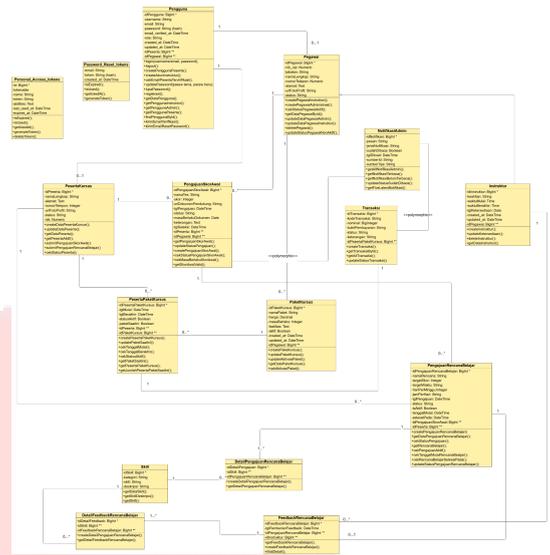
Berikut merupakan sequence diagram yang menunjukkan alur objek di dalam sistem untuk *use case* kelola riwayat transaksi. Terjadi interaksi untuk melihat transaksi dan memverifikasi pembayaran. Pada *sequence diagram* di bawah, ditampilkan bahwa tampilan/*boundary* yang terlibat ada pada halaman utama, halaman riwayat transaksi, dan detail transaksi, dengan controller riwayat transaksi, sementara data yang terlibat berasal dari data peserta paket kursus, data peserta kursus, data transaksi, dan data paket kursus. Pada *sequence diagram* tersebut juga diterapkan berbagai kondisi alternatif yang ditandai dengan *alternative fragment* untuk memvalidasi inputan dan menangani bukti pembayaran valid atau invalid.



Gambar IV. 3 Sequence Diagram Kelola Riwayat Transaksi

4. Class Diagram

Berikut merupakan *class diagram* yang menunjukkan class apa saja yang digunakan dalam sistem termasuk atribut dan metodenya. Antara satu class dengan class yang lain terhubung melalui relasi antar class.



Gambar IV. 4 Class Diagram

C. Pengembangan

Pengembangan dilakukan dengan menyediakan API yang dibutuhkan pada setiap iterasi. Berikut merupakan spesifikasi API yang dikembangkan untuk menangani kelola riwayat transaksi.

Tabel IV. 2 Spesifikasi API untuk Kelola Riwayat Transaksi

Kebutuhan	Sub-Kebutuhan	Autorisasi	Method	URL
Kelola Riwayat Transaksi	Lihat Riwayat transaksi	v	GET	/api/admin/transaksi
	Lihat detail transaksi	v	GET	/api/admin/transaksi/{id}
	Verifikasi pembayaran	v	PATCH	/api/admin/transaksi/{id}/verifikasi

V. VALIDASI, ANALISI HASIL, DAN IMPLIKASI

Pengujian dan evaluasi dilakukan untuk memastikan bahwa sistem mendukung kebutuhan pengguna pada modul manajemen kursus di website Latihan TOEFL ITP. Verifikasi dilakukan oleh pengembang *backend* untuk memastikan setiap API berjalan sesuai rancangan. Sementara *frontend* melakukan validasi terhadap tampilan dan interaksi pengguna, lalu memberi umpan balik jika ada penyesuaian yang diperlukan. Evaluasi dilakukan untuk menilai apakah layanan *backend* sudah mendukung fungsi

modul dengan baik, sehingga sistem berjalan optimal melalui antarmuka pengguna. Pada iterasi pertama, didapatkan seluruh fungsionalitas pada iterasi tersebut berhasil mendapat status passed dalam pengujian unit dan sebagian besar API telah berjalan dengan baik, ditandai dengan status code 200 OK dan 201 Created berdasarkan pengujian API. Namun, masih ada beberapa kekurangan berdasarkan hasil pengujian API, seperti fitur lupa password yang belum berhasil mengirim email, serta API untuk pengajuan rencana belajar dan tinjau rencana belajar yang belum otomatis mengubah status menjadi selesai dan membatasi untuk tidak perlu meninjau ulang rencana belajar yang sudah selesai pada tinjau rencana belajar. Kekurangan ini diperbaiki pada iterasi kedua. Berikut merupakan hasil pengujian untuk iterasi kedua. Pada setiap API kecuali pada verifikasi email ketika lupa password dan reset password telah dilengkapi dengan auth:sanctum sebagai otorisasi untuk melindungi rute (*route*) API agar hanya bisa diakses oleh pengguna yang sudah terautentikasi.

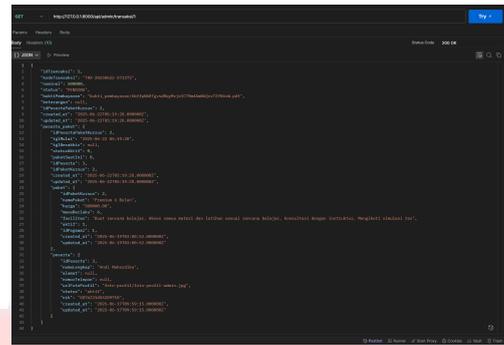
Tabel V. 1 Hasil Pengujian API Iterasi 2

Sub-Kebudayaan	Method	Status Code	URL
Verifikasi Email Ketika lupa password	POST	200 OK	/api/forgot-password
Reset Password	POST	200 OK	/api/reset-password
Admin			
Lihat daftar peserta kursus	GET	200 OK	/api/admin/pantau-peserta
Search peserta kursus tertentu	GET	200 OK	/api/admin/pantau-peserta?search=value
Lihat statistik pengguna	GET	200 OK	/api/admin/dashboard-admin
Lihat riwayat transaksi	GET	200 OK	/api/admin/transaksi
Lihat detail transaksi	GET	200 OK	/api/admin/transaksi/{id}

Verifikasi pembayaran	PATCH	200 OK	/api/admin/transaksi/{id}/verifikasi
Lihat daftar notifikasi terbaru	GET	200 OK	/api/admin/notifikasi/terbaru
Lihat semua notifikasi	GET	200 OK	/api/admin/notifikasi
Lihat notifikasi sesuai status BACA/BELUM TERBACA	GET	200 OK	/api/admin/notifikasi?status=value
Tandai notifikasi terbaca	POST	200 OK	/api/admin/notifikasi/tandai/{id}
Peserta Kursus			
Lihat daftar rencana belajar (termasuk yang sudah selesai)	GET	200 OK	/api/peserta/rencana-belajar
Lihat daftar paket kursus yang tersedia	GET	200 OK	/api/peserta/paket-kursus
Lihat detail paket kursus	GET	200 OK	/api/paket/{id}
Cek peserta kursus memenuhi syarat untuk membeli paket	GET	200 OK	/api/paket/{id}/eligibility

Lihat informasi pembayaran	GET	200 OK	/api/paket/{id}/beli
Mengunggah bukti pembayaran	POST	200 OK	/api/paket/{id}/beli
Lihat daftar riwayat pembayaran	GET	200 OK	/api/pembayaran/riwayat
Instruktur			
Lihat daftar pengajuan rencana belajar (termasuk yang sudah selesai otomatis terupdate statusnya)	GET	200 OK	/api/pengajuan-rencana-belajar
Beri feedback rencana belajar (termasuk menambahkan batasan tidak perlu meninjau ulang rencana belajar yang sudah selesai)	POST	200 OK	/api/pengajuan-rencana-belajar/{id}/feedback

Hasil respons API pada Postman untuk API yang menangani sub-kebutuhan lihat detail transaksi ditunjukkan pada Gambar V. 1. Sementara respons untuk API yang menangani sub-kebutuhan verifikasi pembayaran ditunjukkan pada Gambar V. 2.



Gambar V. 1 Respons API Lihat Detail Transaksi



Gambar V. 2 Respons API Verifikasi Pembayaran

Pengujian *blackbox* terhadap seluruh API pada modul manajemen kursus menunjukkan bahwa seluruh endpoint telah berfungsi sesuai kebutuhan. Selain itu, seluruh komponen yang diuji pada kode internal menggunakan PHPUnit juga berhasil berjalan dengan baik dan memperoleh status *passed*. Validasi dan penanganan error telah diterapkan secara konsisten untuk menjaga integritas data dan mendukung pengalaman pengguna. Backend yang stabil dan terdokumentasi ini mendukung integrasi penuh dengan frontend, termasuk integrasi dengan sistem yang memungkinkan pembelajaran mandiri.

Bagi Telkom University Language Center (LaC), sistem backend ini menjadi solusi atas kendala administratif seperti proses manual dan keterbatasan akses peserta. Proses pendaftaran hingga pembayaran yang efisien dan terintegrasi dengan pembelajaran mandiri mendorong peningkatan layanan serta memperluas jangkauan peserta dengan beban administratif yang lebih ringan. Secara lebih luas, sistem ini turut mendukung pencapaian SDG 4 (Pendidikan Berkualitas) dengan membuka akses pelatihan TOEFL ITP yang terstruktur dan inklusif sebagai bentuk pendidikan nonformal.

VI. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengembangkan back-end modul manajemen kursus untuk website Latihan TOEFL ITP di Telkom University Language Center (LaC) dengan menggunakan metode *iterative incremental*. Sistem yang dibangun mampu menjawab permasalahan administratif, seperti keterbatasan akses peserta, proses pendaftaran dan pembayaran tidak terintegrasi yang rentan *human error* dan kurang efisien, serta disintegrasi dengan sistem yang memungkinkan pembelajaran mandiri. Fitur yang dikembangkan mencakup kebutuhan pengguna dari berbagai peran, seperti peserta, instruktur, dan admin. Pengujian dilakukan melalui *unit testing* dengan PHPUnit dan *blackbox testing* terhadap API yang dikembangkan menunjukkan bahwa sistem backend telah berfungsi dengan baik serta memberikan dukungan teknis terhadap validasi

antarmuka dan alur aplikasi yang dilakukan pihak frontend bersama stakeholder.

Untuk pengembangan lebih lanjut, disarankan agar sistem dilengkapi dengan fitur keamanan yang lebih kuat, seperti pengujian kerentanan secara menyeluruh guna mengantisipasi ancaman siber. Selain itu, integrasi dengan sistem pembayaran otomatis seperti Midtrans atau Xendit dapat mempercepat dan mempermudah proses transaksi. Penerapan caching juga perlu dipertimbangkan guna meningkatkan performa aplikasi, khususnya pada data yang sering diakses, sehingga dapat memberikan pengalaman pengguna yang lebih responsif dan efisien.

REFERENSI

- [1] Ethnologue, "Top 10 most spoken languages," 2024. [Online]. Tersedia: <https://www.ethnologue.com>.
- [2] E. Y. Ernawati, Widiastuti, Y. Harun, dan F. N. Biduri, "THE EFFECT OF USING ENGLISH ON PERFORMANCE IN THE OFFICE FIELD," *International Journal of Social Service and Research*, vol. 4, no. 10, 2024.
- [3] IIEF, "TOEFL ITP Assessment Series Testing & Certification," *IIEF*. [Online]. Tersedia: <https://www.iief.or.id/toefl-family-of-assessments/toefl-ity-a-ssessment-series>.
- [4] Kompas, "5 Beasiswa Jenjang S1-S3 dengan Syarat Skor TOEFL Kurang dari 550," *Kompas.com*, 30 November 2023. [Online]. Tersedia: <https://www.kompas.com/edu/read/2023/11/30/164947171/5-beasiswa-jenjang-s1-s3-dengan-syarat-skor-toefl-kurang-dari-550?page=all>.
- [5] G. Becker, *Human Capital: A Theoretical and Empirical Analysis, with Special Reference to Education*, 3rd ed. University of Chicago Press, 1993.
- [6] R. Pangestika dan R. T. Dirgahayu, "Pengembangan Back-end Sistem Informasi Pendataan Sekolah Desa Komunitas Pendar Foundation Yogyakarta," 2021.
- [7] J. X. Xiao, M. J. Luo, dan W. Li, "From the fuzzy front end to the back end: A participatory action research approach to co-design promoting sustainable behaviour," *Design Journal*, vol. 26, no. 1, pp. 32–51, 2023, doi: 10.1080/14606925.2022.2113262.
- [8] A. Ehsan, M. A. M. E. Abuhaliqa, C. Catal, dan D. Mishra, "RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions," *Applied Sciences (Switzerland)*, vol. 12, no. 9, 2022, doi: <https://doi.org/10.3390/app12094369>.
- [9] A. Yudhistira, L. D. Pangesti, G. Isran, R. Bagus, B. Sumantri, dan R. Suryani, "Perancangan dan Implementasi Sistem Informasi Perpustakaan Berbasis Web," *Jurnal Sistem Informasi dan Komputerisasi Akuntansi*, 2023, pp. 2579–4477.
- [10] F. N. Saidah, R. R. Isnanto, dan I. F. H. Alim, "Usaha Koperasi Siswa SMK Negeri 2 Semarang Menggunakan Kerangka Kerja-Laravel," *Jurnal Teknik Komputer*, vol. 3, no. 1, pp. 25–31, 2024, doi: 10.14710/jtk.v3i1.44284.
- [11] N. Dwivedi, D. Katiyar, dan G. Goel, "A Comparative Study of Various Software Development Life Cycle (SDLC) Models," *International Journal of Research in Engineering, Science and Management*, vol. 5, no. 3, pp. 2581–5792, 2022. [Online]. Tersedia: <https://www.ijresm.com>.
- [12] I. M. Ibrahim, O. F. Nonyelum, dan I. R. Saidu, "ITERATIVE AND INCREMENTAL DEVELOPMENT ANALYSIS STUDY OF VOCATIONAL CAREER INFORMATION SYSTEMS," *International Journal of Software Engineering & Applications (IJSEA)*, vol. 11, no. 5, pp. 13–24, 2020.
- [13] S. Narulita, A. Nugroho, dan M. Z. Abdillah, "Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS)," *Bridge : Jurnal publikasi Sistem Informasi dan Telekomunikasi*, vol. 2, no. 3, pp. 244–256, 2024, doi: <https://doi.org/10.62951/bridge.v2i3.174>.
- [14] Hevner, March, Park, and Ram, "Design Science in Information Systems Research," *MIS Quarterly*, vol. 28, no. 1, p. 75, 2004, doi: 10.2307/25148625.