



Implementasi Model Extreme Programming dalam Pengembangan Sistem Simulasi Tes TOEFL

Reyvi Rahman*, Ulfa Khaira, Zainil Abidin

Fakultas Sains dan Teknologi, Program Studi Sistem Informasi, Universitas Jambi, Jambi, Indonesia

Email: ^{1,*}reyvirahman@gmail.com, ²ulfakhaira@unja.ac.id, ³zainil.abidin@unja.ac.id

Email Penulis Korespondensi: reyvirahman@gmail.com

Abstrak—Kemampuan bahasa Inggris merupakan salah satu syarat penting dalam dunia akademik dan profesional, di mana TOEFL digunakan sebagai alat ukur standar. Namun, banyak calon peserta yang belum siap menghadapi ujian ini karena kurangnya pemahaman dan latihan yang menyerupai kondisi ujian sebenarnya. Yanto Tanjung English Academy merupakan lembaga kursus yang telah memberikan pembelajaran TOEFL secara daring, namun belum menyediakan fasilitas simulasi ujian secara menyeluruh. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem simulasi tes TOEFL berbasis web guna memberikan pengalaman ujian yang mendekati kondisi asli, sehingga peserta dapat lebih siap dan percaya diri. Metode yang digunakan dalam pengembangan sistem ini adalah Extreme Programming (XP), yang terdiri dari tahapan perencanaan, perancangan, pengkodean, dan pengujian. Pengujian dilakukan dengan pendekatan Test-Driven Development (TDD) untuk memastikan fungsionalitas dan kesesuaian sistem dengan kebutuhan pengguna. Hasilnya menunjukkan bahwa seluruh fitur utama dapat diakses dengan baik, yang dibuktikan dengan waktu eksekusi yang efisien pada seluruh fitur yang diuji, dengan rata-rata di angka 2.126 detik. Data performa ini mengonfirmasi bahwa pendekatan TDD berhasil menghasilkan sistem yang tidak hanya fungsional tetapi juga responsif bagi pengguna, serta memperoleh tingkat penerimaan yang tinggi dari stakeholder dan peserta kursus. Sistem ini diharapkan mampu meningkatkan efektivitas pembelajaran TOEFL secara mandiri dan fleksibel, serta menjadi solusi digital yang bermanfaat bagi peserta kursus maupun masyarakat umum yang ingin berlatih TOEFL secara online.

Kata Kunci: TOEFL; Simulasi Tes; Extreme Programming; Website; TDD

Abstract—English proficiency is a key requirement in both academic and professional settings, with TOEFL serving as a standardized measure of competency. However, many candidates are unprepared for this exam due to a lack of understanding and practice that simulates actual test conditions. Yanto Tanjung English Academy, a course provider offering online TOEFL instruction, currently lacks a comprehensive test simulation facility. Therefore, this research aims to develop a web-based TOEFL test simulation system to provide a realistic exam experience, enabling participants to be better prepared and more confident. The system was developed using the Extreme Programming (XP) method, which consists of planning, design, coding, and testing phases. Testing was conducted using a Test-Driven Development (TDD) approach to ensure the system's functionality and alignment with user requirements. The results show that all main features are accessible, as evidenced by an efficient average execution time of 2.126 seconds across all tested features. This performance data confirms that the TDD approach successfully yielded a system that is not only functional but also responsive for its users. Furthermore, the system received a high acceptance rate from both stakeholders and course participants. This system is expected to enhance the effectiveness of independent and flexible TOEFL learning and to serve as a beneficial digital solution for course participants and the general public wishing to practice TOEFL online.

Keywords: TOEFL; Test Simulation; Extreme Programming; Website; TDD

1. PENDAHULUAN

TOEFL adalah salah satu alat ukur kemampuan berbahasa Inggris yang penting, namun masih banyak orang yang tidak tahu akan hal itu. Bagi mereka yang belum tahu dengan istilah ini, TOEFL sering kali dianggap sulit dan menakutkan. Hal ini tidak hanya dialami oleh kalangan siswa, tetapi juga dialami oleh masyarakat umum. Akibatnya, ketidakpahaman mengenai TOEFL menjadi hambatan bagi banyak individu untuk mewujudkan impian mereka, seperti melanjutkan pendidikan ke luar negeri atau mendapatkan pekerjaan yang diinginkan. Rasa takut terhadap ujian ini sering menjadi penyebab kegagalan dalam meraih cita-cita (Sudewi et al., 2024).

Secara umum, peserta ujian TOEFL diharuskan mencapai standar skor tertentu agar dapat memenuhi persyaratan, baik untuk mendaftar beasiswa internasional, melanjutkan pendidikan, maupun melamar pekerjaan. Skor minimal yang seringkali dibutuhkan adalah 450 atau lebih, yang menuntut usaha keras dari peserta. Untuk mencapai skor tersebut, berbagai langkah dapat dilakukan, baik melalui program kursus intensif maupun belajar mandiri (Sudewi et al., 2024) Salah satu program kursus intensif yang dapat menjadi pilihan adalah Yanto Tanjung *English Academy*.

Yanto Tanjung *English Academy* adalah *platform* kursus *online* yang dirancang untuk membantu peserta mempersiapkan diri secara optimal dalam menghadapi ujian TOEFL. Dengan bimbingan dari tutor berpengalaman, peserta tidak hanya mempelajari strategi menjawab soal dengan efektif, tetapi juga meningkatkan kemampuan Bahasa Inggris secara keseluruhan, sehingga peluang mereka untuk mencapai skor minimal 450 atau lebih menjadi lebih besar. Pelaksanaan kursus dilakukan melalui Zoom Meeting, yang memungkinkan peserta untuk mengikuti pembelajaran dari mana saja tanpa perlu datang langsung ke tempat kursus. Pendekatan ini dirancang agar peserta dapat belajar dengan nyaman dan fleksibel, cukup dari rumah atau lokasi lain sesuai kebutuhan mereka.

Namun, hingga saat ini, Yanto Tanjung *English Academy* belum menyediakan fasilitas simulasi TOEFL secara khusus. Kursus ini lebih berfokus pada pengajaran cara menjawab soal dengan benar, termasuk strategi menghadapi berbagai tipe soal yang sering muncul dalam ujian TOEFL. Pendekatan ini membantu peserta memahami konsep dasar dan pola soal, namun belum sepenuhnya memberikan pengalaman seperti ujian sesungguhnya. Dengan demikian,



peserta masih memerlukan latihan tambahan secara mandiri untuk membiasakan diri dengan kondisi ujian yang sebenarnya, seperti pengelolaan waktu dan tekanan dalam mengerjakan soal dalam satu sesi penuh.

Oleh karena itu, diperlukan adanya sistem simulasi tes TOEFL yang dapat memberikan pengalaman ujian yang mendekati kondisi sebenarnya. Sistem ini akan membantu peserta memahami alur pengerjaan soal, mengelola waktu dengan lebih baik, serta meningkatkan kepercayaan diri saat menghadapi ujian resmi. Selain itu, simulasi tes berbasis web juga memungkinkan peserta untuk berlatih secara mandiri kapan saja dan di mana saja, sehingga lebih fleksibel dan sesuai dengan kebutuhan mereka. Kehadiran sistem ini tidak hanya akan melengkapi layanan Yanto Tanjung *English Academy*, tetapi juga memberikan nilai tambah yang signifikan bagi peserta dalam upaya mereka mencapai skor TOEFL yang diinginkan. Jumlah pengguna yang akan menggunakan sistem simulasi tes TOEFL ini diperkirakan sejalan dengan jumlah peserta kursus yang terdaftar di Yanto Tanjung *English Academy*. Setiap bulannya terdapat sekitar 10 hingga 25 peserta yang mengikuti program kursus. Dengan demikian, sistem ini memiliki potensi untuk digunakan secara rutin oleh puluhan peserta setiap bulan. Selain itu, sistem simulasi ini dirancang sebagai *platform* terbuka, sehingga tidak hanya terbatas bagi peserta internal kursus, tetapi juga dapat diakses oleh masyarakat umum yang ingin berlatih TOEFL secara mandiri.

Dalam pengembangan perangkat lunak, *Software Development Life Cycle* (SDLC) atau siklus hidup pengembangan perangkat lunak adalah serangkaian langkah sistematis yang dirancang untuk menciptakan dan memperbarui perangkat lunak, mencakup model dan metode yang digunakan dalam prosesnya. SDLC menawarkan berbagai pendekatan populer, seperti *Waterfall*, *Prototype*, *Rapid Application Development* (RAD), *Evolutionary*, *Agile*, *Rational Unified Process* (RUP), *Build and Fix*, *Extreme Programming* (XP), *V-Shape Model*, serta *Fountain*. Masing-masing pendekatan memiliki keunikan dan strategi tersendiri yang dapat diterapkan sesuai dengan kebutuhan spesifik suatu proyek perangkat lunak (Siva et al., 2023).

Pada penelitian ini, metode yang digunakan adalah *Extreme Programming* karena menurut (Faizal et al., 2021) *Extreme Programming* (XP) adalah salah satu metodologi dalam pengembangan perangkat lunak berbasis agile yang menitikberatkan pada pengkodean (*coding*) sebagai aktivitas utama dalam siklus pengembangan. Metode ini dikenal karena kemampuannya yang tinggi dalam merespons perubahan selama proses pengembangan. Salah satu keunggulan XP adalah kemampuannya untuk menyelesaikan proyek dalam waktu yang relatif singkat sambil tetap mempertahankan fokus pada tujuan yang ingin dicapai. Tahapan dalam metode XP meliputi perencanaan (*planning*), perancangan (*design*), pengkodean (*coding*), dan pengujian (*testing*).

Metode *Extreme Programming* telah diterapkan dalam berbagai penelitian sebelumnya, termasuk penelitian yang dilakukan oleh (Amdi Rizal et al., 2022) mengenai pengembangan Aplikasi *Inventory* Persediaan Barang Berbasis Web pada studi kasus *Esha 2 Cell*. Penelitian ini mengungkapkan bahwa metode XP memungkinkan proses pengembangan sistem yang lebih cepat serta memberikan fleksibilitas tinggi dalam menghadapi perubahan selama proses pembangunan perangkat lunak. Penelitian lainnya dilakukan oleh (Sarasvananda et al., 2021) mengenai Pendekatan Metode *Extreme Programming* untuk Pengembangan Sistem Informasi Manajemen Surat Menyurat pada LPIK STIKI. Hasil penelitian ini menunjukkan bahwa pendekatan XP yang digunakan dalam pengembangan sistem mampu mempercepat proses pembangunan serta terbukti efisien dalam meningkatkan kinerja sistem. Selanjutnya penelitian yang dilakukan oleh (Mutezar & Salamah, 2021) mengenai Pengembangan Sistem Manajemen *Event* Pameran Karya Mahasiswa Menggunakan Metode *Extreme Programming*. Hasil penelitian ini menunjukkan bahwa penerapan XP mampu menghasilkan sistem yang berkualitas, karena melibatkan pengguna secara aktif dalam memberikan umpan balik selama proses pengembangan. Metode *Extreme Programming* (XP) sangat cocok diterapkan dalam pembangunan sistem simulasi tes TOEFL karena pendekatannya yang fleksibel dan efisien. Dengan fokus pada pengkodean sebagai aktivitas utama, XP memungkinkan pengembang untuk lebih cepat mengimplementasikan fitur-fitur penting seperti tampilan antarmuka pengguna, mekanisme penilaian otomatis, serta pengelolaan waktu pengerjaan soal. *Responsivitas* XP terhadap perubahan juga memberikan keuntungan dalam menyesuaikan sistem dengan masukan pengguna atau kebutuhan baru yang mungkin muncul selama proses pengembangan. Selain itu, tahapan perencanaan dan pengujian yang terstruktur dalam XP membantu memastikan bahwa sistem simulasi yang dihasilkan dapat berfungsi dengan baik, memberikan pengalaman pengguna yang optimal, dan mendukung peserta dalam persiapan menghadapi ujian TOEFL.

Dalam pembangunan sistem simulasi tes TOEFL menggunakan metode *Extreme Programming* (XP), pengujian menjadi salah satu tahapan krusial untuk memastikan kualitas dan fungsionalitas sistem untuk itu pada penelitian ini metode pengujian yang diterapkan yaitu *Test-Driven Development*. Pengujian *Test-Driven Development* memiliki tujuan untuk menciptakan unit pemrograman yang dapat diuji secara mandiri, memastikan setiap unit bekerja dengan baik, dan menghasilkan aplikasi yang berfungsi tanpa *error* (Hidayat et al., 2024). Dengan TDD, pengembang dapat mengurangi potensi kesalahan dalam kode, membangun sistem yang lebih stabil, serta mempermudah proses identifikasi dan perbaikan masalah.

Beberapa penelitian terdahulu telah mengimplementasikan metode *Extreme Programming* (XP) dalam pengembangan sistem. Penelitian oleh Amdi Rizal et al. (2022) menghasilkan aplikasi *inventory* berbasis web yang cepat dikembangkan dengan XP. Sarasvananda et al. (2021) menerapkan XP pada sistem informasi manajemen surat menyurat dengan hasil efisiensi proses pengembangan. Mutezar & Salamah (2021) juga membuktikan efektivitas XP dalam pengembangan sistem manajemen event. Meskipun penelitian-penelitian tersebut menunjukkan keberhasilan XP, belum ada penelitian yang secara khusus menerapkan XP dalam pengembangan sistem simulasi TOEFL berbasis web. Selain itu, penelitian sebelumnya umumnya hanya menekankan aspek pengembangan sistem, sementara aspek

pengujian dengan Test-Driven Development (TDD) sebagai bagian integral dari XP masih jarang diuraikan secara mendalam.

2. METODOLOGI PENELITIAN

2.1 Objek Penelitian

Objek penelitian adalah sistem simulasi tes TOEFL berbasis web yang dikembangkan untuk mendukung proses pembelajaran di Yanto Tanjung English Academy. Sistem ini difokuskan pada pembuatan platform simulasi TOEFL ITP yang mendekati pengalaman ujian sesungguhnya.

2.2 TOEFL dan Simulasi Tes

TOEFL (Test of English as a Foreign Language) adalah salah satu tes yang paling umum digunakan untuk menilai kemampuan berbahasa Inggris bagi penutur non-native (Iswati, 2021). Pada penelitian ini, simulasi difokuskan pada TOEFL ITP yang mencakup tiga sesi utama, yaitu Listening Comprehension, Structure and Written Expression, serta Reading Comprehension. Simulasi tes dirancang agar peserta memperoleh pengalaman yang menyerupai ujian TOEFL sebenarnya. Fitur yang diimplementasikan antara lain: timer yang tidak bisa dimanipulasi, audio sekali putar untuk sesi Listening, serta scoring otomatis berdasarkan standar TOEFL ITP.

2.4 Extreme Programming

Extreme Programming merupakan metode pengembangan perangkat lunak yang menekankan kecepatan proses. Tahapannya meliputi perencanaan, perancangan, penulisan kode, dan pengujian. Selain itu, aspek pemeliharaan terhadap perangkat lunak yang dihasilkan juga menjadi bagian penting dalam pendekatan ini (Yoga et al., 2025).

2.5 Website

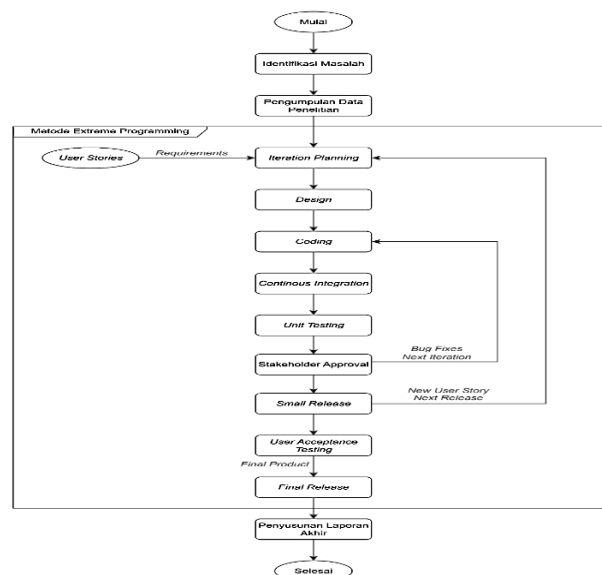
Website merupakan kumpulan halaman informasi yang dapat diakses melalui jaringan internet kapan pun dan di mana pun pengguna terhubung (Rahmi et al., 2023).

2.6 Test Driven Development (TDD)

Dalam setiap iterasi, pengujian dilakukan dengan pendekatan Test Driven Development (TDD). Prinsipnya adalah menulis tes terlebih dahulu, lalu menulis kode hingga tes berhasil. Dengan pendekatan ini, sistem dipastikan berjalan sesuai kebutuhan dan terjaga kualitas kodenya.

2.7 Kerangka Kerja Penelitian

Penelitian ini dilakukan di Yanto Tanjung *English Academy*, yang berlokasi di Kotabaru, Jambi, selama enam bulan, mulai dari Januari hingga Juni 2025. Penelitian ini berfokus pada pengembangan perangkat lunak simulasi tes TOEFL berbasis web. Dalam pengembangan perangkat lunak tersebut, digunakan pendekatan yang terstruktur melalui tahapan *Software Development Lifecycle* (SDLC). Salah satu model SDLC yang diterapkan dalam penelitian ini adalah *Extreme Programming* (XP), yang dikenal dengan pendekatan iteratif dan kolaboratifnya. Adapun tahapan-tahapan dalam model Extreme Programming yang diikuti selama proses pengembangan dijelaskan dalam kerangka kerja pada gambar 1 berikut.



Gambar 1. Kerangka Kerja Penelitian



Dari Gambar 1 Kerangka Kerja Penelitian tersebut, dapat diuraikan sebagai berikut :

1. Tahap awal dalam proses penelitian ini dimulai dengan mengidentifikasi masalah melalui analisis terhadap proses kursus yang sedang berjalan di Yanto Tanjung *English Academy*. Berdasarkan hasil analisis tersebut, ditemukan bahwa lembaga ini belum menyediakan fasilitas khusus untuk simulasi tes TOEFL. Selama ini, kegiatan pembelajaran lebih berfokus pada pemberian materi dan strategi menjawab soal, tanpa adanya sarana yang memungkinkan peserta merasakan pengalaman ujian TOEFL secara nyata. Hal ini menjadi hambatan utama dalam mempersiapkan peserta secara menyeluruh menghadapi ujian sesungguhnya..
2. Untuk memahami kebutuhan dan kendala secara lebih mendalam, peneliti melakukan pengumpulan data melalui tiga metode utama: observasi langsung di lokasi, wawancara semi-terstruktur dengan pemilik kursus, dan studi literatur dari berbagai sumber relevan..
3. Berdasarkan data yang telah dikumpulkan, peneliti menyusun *user stories* untuk menggambarkan peran, kebutuhan, dan aktivitas masing-masing aktor yang akan terlibat dalam sistem.
4. Hasil dari penyusunan *user stories* kemudian digunakan dalam proses perencanaan iterasi. Sistem dibagi menjadi dua bagian pengembangan utama, yaitu fitur untuk Admin dan fitur untuk User.
5. Pada tahap *design*, dilakukan pemodelan sistem berdasarkan hasil analisis kebutuhan yang telah diperoleh. *Unified Modeling Language* (UML) adalah bahasa pemodelan perangkat lunak standar yang digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan berbagai aspek sistem perangkat lunak. Secara sederhana, seperti arsitek yang membuat *blueprint* gedung, arsitek perangkat lunak menggunakan diagram UML sebagai panduan bagi *programmer* dalam mengembangkan perangkat lunak (Sumiati et al., 2021). Salah satu diagramnya adalah *Use Case Diagram*, yang menggambarkan kebutuhan fungsional sistem, termasuk siapa saja yang berinteraksi dengan sistem dan fungsi yang dapat mereka akses (Dharmawan, 2023). *Activity Diagram* digunakan untuk merancang alur aktivitas atau alur kerja sistem, mendefinisikan, dan mengelompokkan alur tampilan (Dharmawan, 2023). *Class Diagram* merepresentasikan hubungan antar kelas beserta detailnya, menggambarkan struktur program, aturan, tanggung jawab entitas, dan perilaku sistem, sekaligus memvisualisasikan alur *database*. Setiap kelas digambarkan dalam bentuk persegi yang memuat nama kelas, atribut, dan metode, sehingga menjadi elemen utama dalam desain berbasis objek (Ramdany et al., 2024).
6. *Coding* merupakan tahap di mana rancangan dan desain diubah menjadi kode yang membentuk aplikasi secara keseluruhan. Pada pengembangan sistem ini, React JS digunakan sebagai pustaka JavaScript *open-source* untuk membangun antarmuka pengguna (*User Interface*), khususnya pada aplikasi satu halaman (*single-page application*). Dengan konsep komponen yang dapat digunakan kembali, React memudahkan pembuatan desain antarmuka yang efisien dan konsisten di berbagai platform (Tri Sulistyorini et al., 2022). Untuk sisi *backend*, digunakan Express.js, yaitu kerangka kerja JavaScript yang menyediakan struktur, alat bantu, dan dukungan *middleware* untuk mempercepat serta mempermudah pengembangan aplikasi web modern (Rafi Nahjan et al., 2023). Proses pengujian kode dilakukan menggunakan Jest, sebuah *test runner* yang memastikan keakuratan kode JavaScript melalui penyusunan dan eksekusi pengujian otomatis (Rahmani, 2020). Sedangkan untuk pengelolaan basis data, sistem ini menggunakan MySQL, yaitu sistem manajemen basis data relasional (*RDBMS*) *open-source* yang cepat, andal, dan mendukung akses multi-pengguna secara bersamaan (Endang & Velia Susanti, 2021).
7. *Continuous Integration* merupakan praktik dalam pengembangan perangkat lunak yang memungkinkan pengembang secara rutin mengintegrasikan perubahan kode ke dalam repositori utama. Dalam penerapannya, digunakan GitHub sebagai platform manajemen proyek dan sistem *versioning code* yang juga berfungsi sebagai jaringan sosial khusus bagi para pengembang. GitHub merupakan salah satu penyimpanan kode daring terbesar di dunia yang mendukung kerja kolaboratif, dengan Git sebagai inti utamanya (Mujiyanto & Maukar, 2021). Selain sebagai sistem kontrol versi, GitHub juga berperan sebagai layanan *web hosting* yang menyediakan berbagai fasilitas untuk mempermudah kolaborasi tim secara efisien (Ferdiana Sari & Ekohariadi, 2021).
8. Setiap komponen sistem diuji secara terpisah melalui *Unit Testing* dengan pendekatan Test Driven Development (TDD). Metode ini menekankan penulisan tes sebelum implementasi kode, mengikuti siklus *Red, Green, Refactor* untuk memastikan kode tetap andal, modular, dan terdokumentasi dengan baik (Wijanarko & Subhiyanto, 2024).
9. Setelah pengembangan dan pengujian awal selesai, sistem diserahkan kepada *stakeholder* untuk dilakukan evaluasi dan validasi. Tahap *Stakeholder Approval* memastikan bahwa sistem yang telah dibangun benar-benar memenuhi kebutuhan pengguna dan sesuai dengan harapan yang telah ditetapkan sejak awal. Persetujuan dari *stakeholder* menjadi syarat sebelum sistem dirilis ke tahap selanjutnya.
10. Tahap *Small Release* yaitu rilis terbatas yang hanya dapat diakses oleh pengguna tertentu melalui tautan.
11. Tahap terakhir adalah *Final Release*, di mana sistem dirilis secara penuh dan dapat diakses secara publik.
12. Sebagai tahap akhir, peneliti menyusun laporan akhir berdasarkan keberhasilan sistem yang telah dikembangkan.

3. HASIL DAN PEMBAHASAN

3.1 User Stories

Pada tahap ini, dilakukan proses identifikasi aktor beserta hak aksesnya dalam sistem. Peneliti mengembangkan *user stories* untuk mendeskripsikan peran dan fungsi masing-masing aktor yang terlibat. Spesifikasi *user stories* diperoleh

melalui wawancara kebutuhan sistem dengan pengajar selaku pemilik kursus Yanto Tanjung *English Academy*. Hasil spesifikasi tersebut dirangkum dalam Tabel 1 berikut ini.

Tabel 1. User Stories

Aktor	User Stories
User	User memiliki hak akses untuk memulai dan mengerjakan Mini Test TOEFL yang terdiri dari bagian <i>Listening Comprehension</i> , <i>Structure and Written Expression</i> , serta <i>Reading Comprehension</i> . Selain itu, peserta dapat melihat Riwayat Mini Test yang telah dikerjakan. Peserta juga dapat mengikuti simulasi tes TOEFL secara penuh, mengerjakan setiap bagian ujian, serta mengakses Riwayat simulasi untuk mengevaluasi hasil dan perkembangan mereka.

Berdasarkan Tabel 1, kebutuhan utama sistem berfokus pada peran pengguna (*user*) yang menjadi aktor utama. User diberi hak akses penuh untuk mengerjakan *Mini Test TOEFL* maupun simulasi tes TOEFL secara menyeluruh. Fitur riwayat disediakan agar user dapat meninjau kembali hasil tes yang sudah dikerjakan, sehingga mempermudah proses evaluasi belajar. Penentuan *user stories* ini menjadi dasar dalam perancangan sistem, karena langsung mencerminkan kebutuhan pengguna terhadap aplikasi simulasi TOEFL.

3.2 Iteration Planning

Iteration Planning menentukan urutan pengerjaan *user stories* berdasarkan prioritas, dengan fitur bernilai bisnis tertinggi dikembangkan lebih dulu. Nilai prioritas ditetapkan melalui diskusi dengan *stakeholder*, mempertimbangkan urgensi, dampak, dan kemudahan implementasi, sehingga pengembangan berjalan bertahap, terfokus, dan dapat langsung diuji. Fitur-fitur tersebut dapat dilihat pada Tabel 2 berikut ini.

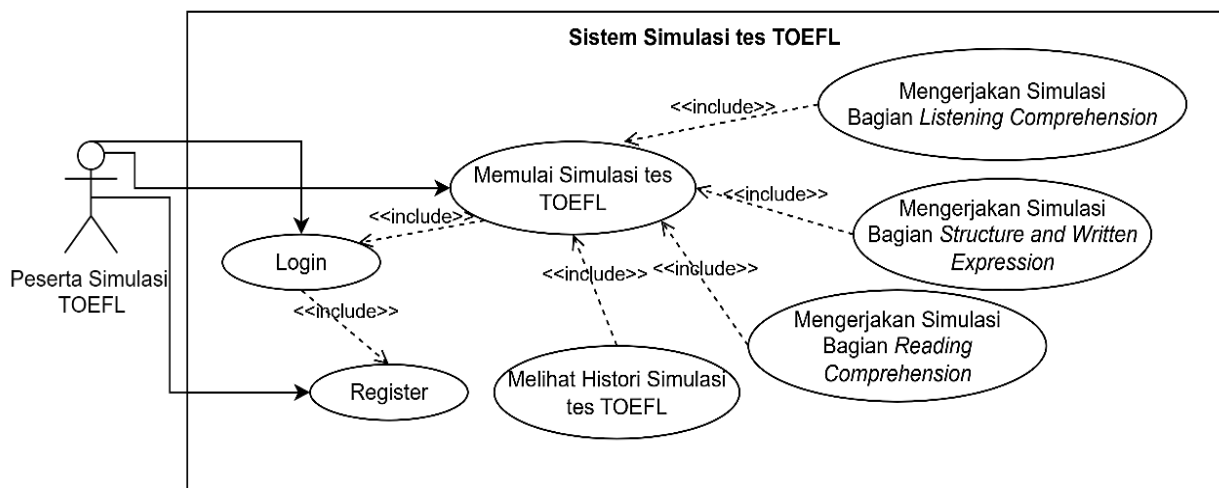
Tabel 2. Nilai Prioritas Fitur Iterasi

No	Nama Fitur	Halaman	Nilai Prioritas
1	Registrasi Akun	Halaman Register	5
2	Lihat Paket Soal	Halaman Paket Soal	4
3	Mengerjakan Simulasi TOEFL	Halaman Simulasi TOEFL	3
4	Lihat Riwayat Skor	Halaman Riwayat Simulasi	2

Berdasarkan Tabel 2 setiap fitur dengan nilai tinggi akan dimasukkan lebih awal dalam iterasi agar dapat segera diuji dan mendapatkan umpan balik dari pengguna. Dengan cara ini, sistem dikembangkan secara adaptif dan responsif terhadap kebutuhan.

3.2 Design

Pada tahap *design*, dilakukan pemodelan sistem berdasarkan hasil analisis kebutuhan yang telah diperoleh. Pemodelan sistem yang digunakan adalah *Use-Case Diagram* sebagaimana yang bisa dilihat pada gambar berikut ini.



Gambar 2. Rancangan *Use Case Diagram*

Gambar 2 Rancangan *Use Case Diagram* menunjukkan interaksi antara aktor dan sistem dalam sistem simulasi tes TOEFL. Aktor yang terlibat adalah peserta simulasi TOEFL. Peserta dapat melakukan *login* atau *register*, melihat histori simulasi tes, dan memulai simulasi TOEFL yang terdiri dari tiga bagian, yaitu *Listening Comprehension*, *Structure and Written Expression*, serta *Reading Comprehension*. Diagram ini juga menunjukkan hubungan *include* yang menandakan keterkaitan antar proses untuk memastikan alur sistem berjalan sesuai kebutuhan pengguna.

3.3 Iteration Implementation

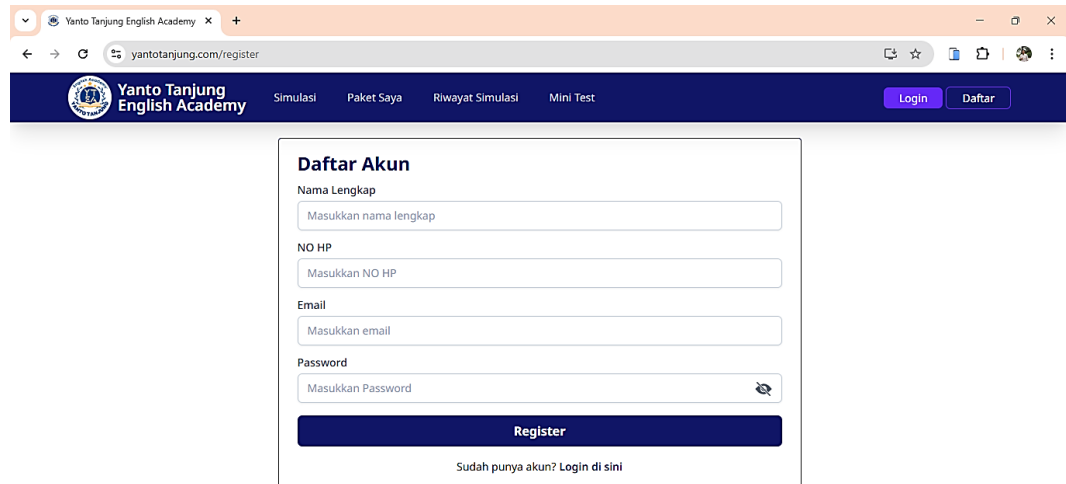
3.3.1 Iterasi Tahap 1 Admin

a. Coding.

Pada tahap ini, proses implementasi difokuskan pada pengembangan fitur utama yang dibutuhkan oleh peserta simulasi tes TOEFL untuk melakukan simulasi TOEFL. Implementasi dilakukan secara bertahap berdasarkan hasil dari *design user interface*. Berikut ini fitur yang di implementasikan:

1. Registrasi Akun.

Pada halaman registrasi, pengguna wajib mengisi nama lengkap, nomor HP, *email*, dan *password*. Setelah data *valid* dan terkirim ke *server*, informasi disimpan di *database*, akun dibuat, dan pengguna dapat langsung menuju halaman login. Gambar 3 berikut menunjukkan tampilan halaman untuk registrasi akun.



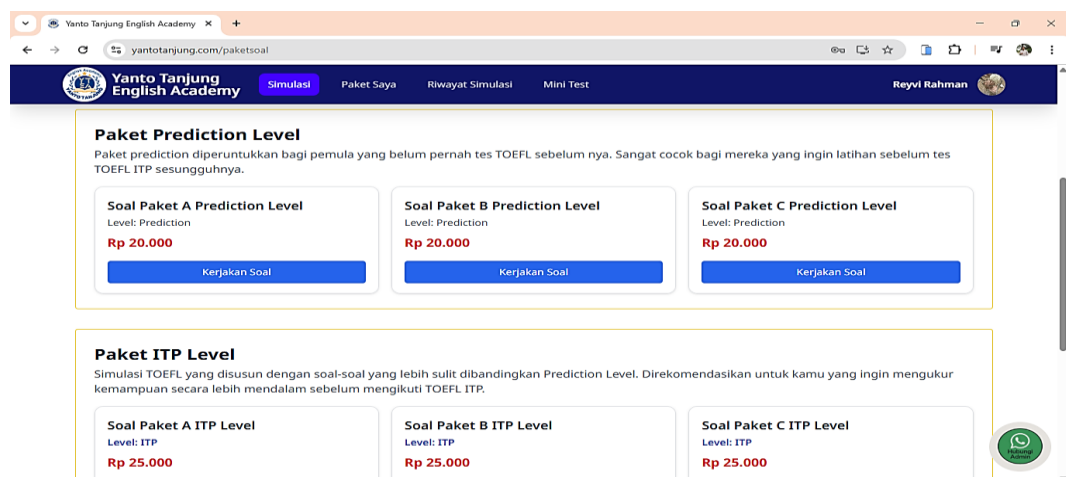
The screenshot shows a web browser window with the URL yantotanjung.com/register. The page title is "Daftar Akun". It features a dark blue header with the Yanto Tanjung English Academy logo and navigation links: "Simulasi", "Paket Saya", "Riwayat Simulasi", and "Mini Test". On the right side of the header are "Login" and "Daftar" buttons. The main content area is a white box with the following fields: "Nama Lengkap" (Masukkan nama lengkap), "NO HP" (Masukkan NO HP), "Email" (Masukkan email), and "Password" (Masukkan Password). A dark blue "Register" button is at the bottom of the form, with a link "Sudah punya akun? Login di sini" below it.

Gambar 3. Halaman Registrasi Akun

Gambar 3 menunjukkan tampilan halaman registrasi akun. Pada halaman ini pengguna wajib mengisi nama lengkap, nomor HP, email, dan password. Setelah data divalidasi dan tersimpan dalam database, akun berhasil dibuat dan pengguna dapat melanjutkan ke proses login.

2. Lihat Paket Soal

Pada halaman paket soal simulasi tes TOEFL, peserta dapat melihat seluruh paket soal yang tersedia dan memilih paket yang ingin dikerjakan dengan menekan tombol Kerjakan Soal. Halaman paket soal bisa dilihat pada Gambar 4 berikut ini.



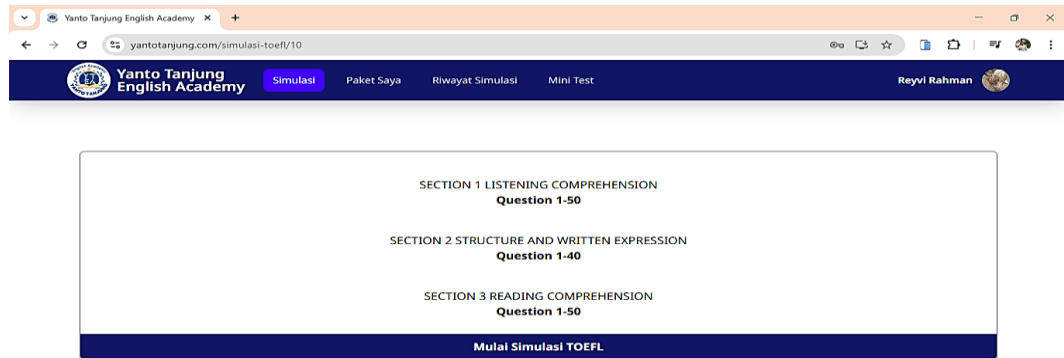
The screenshot shows a web browser window with the URL yantotanjung.com/paketsoal. The page title is "Paket Prediction Level". It features a dark blue header with the Yanto Tanjung English Academy logo and navigation links: "Simulasi", "Paket Saya", "Riwayat Simulasi", and "Mini Test". On the right side of the header is the user name "Reyvi Rahman". The main content area is a white box with the following sections: "Paket Prediction Level" (Paket prediction diperuntukkan bagi pemula yang belum pernah tes TOEFL sebelum nya. Sangat cocok bagi mereka yang ingin latihan sebelum tes TOEFL ITP sesungguhnya.) and "Paket ITP Level" (Simulasi TOEFL yang disusun dengan soal-soal yang lebih sulit dibandingkan Prediction Level. Direkomendasikan untuk kamu yang ingin mengukur kemampuan secara lebih mendalam sebelum mengikuti TOEFL ITP.). Each section contains three options (A, B, C) with a "Kerjakan Soal" button. The prices for each option are Rp 20.000 for Prediction Level and Rp 25.000 for ITP Level.

Gambar 4. Halaman Paket Soal

Gambar 4 menunjukkan tampilan halaman paket soal. Pada halaman ini pengguna bisa memilih berbagai macam paket soal yang bisa di kerjakan untuk latihan TOEFL.

3. Simulasi tes TOEFL

Pada halaman simulasi tes TOEFL peserta simulasi bisa melihat informasi terkait simulasi tes TOEFL tersebut seperti jumlah sesi dan jumlah soal nya. Di halaman simulasi tes TOEFL ini peserta bisa menekan tombol mulai simulasi tes TOEFL yang akan memulai sesi simulasi tes TOEFL. Gambar 5 berikut ini merupakan tampilan halaman simulasi tes TOEFL.

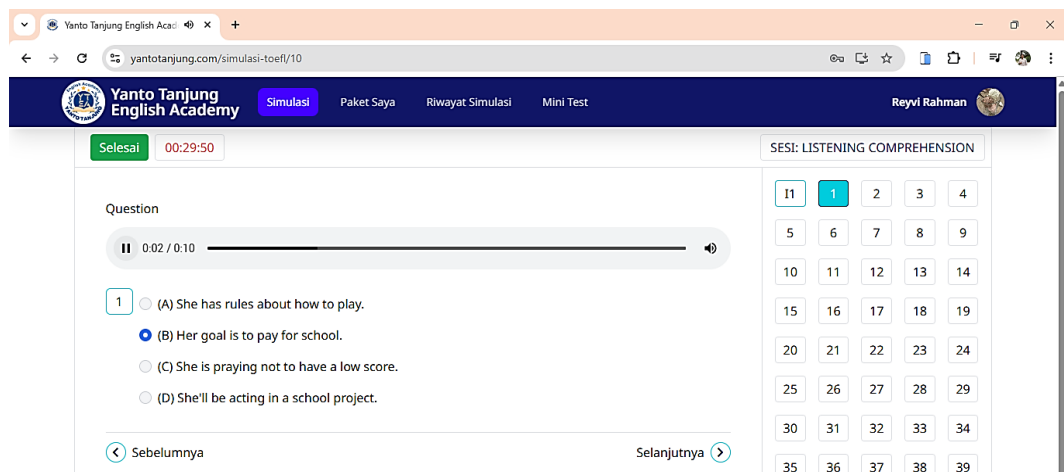


Gambar 5. Halaman Simulasi Tes TOEFL

Gambar 5 menunjukkan tampilan halaman simulasi tes TOEFL. Pada halaman ini pengguna dapat melihat informasi berupa jumlah sesi dan jumlah soal yang ada pada simulasi tes TOEFL. Pengguna bisa menekan tombol mulai simulasi TOEFL untuk mulai mengerjakan soal.

4. Simulasi Tes TOEFL Bagian *Listening Comprehension*

Pada halaman Simulasi Tes TOEFL *listening comprehension* peserta dapat melihat soal beserta pilihan ganda nya. Untuk sesi *listening* ini soal nya berbentuk *audio* yang bisa di klik peserta untuk memulai mendengarkan *audio* nya. Peserta bisa menjawab soal tersebut dengan klik pilihan ganda yang telah di sediakan pada halaman yang bisa di lihat pada Gambar 6 berikut ini.

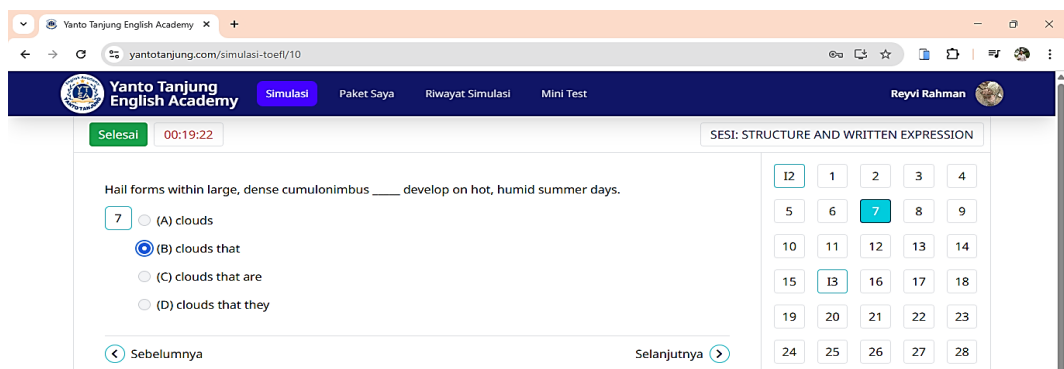


Gambar 6. Halaman Simulasi Tes TOEFL Bagian *Listening Comprehension*

Gambar 6 menunjukkan simulasi tes TOEFL bagian listening comprehension. Pada halaman ini pengguna dapat menjawab soal dengan memilih pilihan ganda yang telah tersedia pada soal nya. Tiap soal terdiri dari 4 pilihan ganda. Untuk soal listening peserta dapat mendengarkan audio terlebih dahulu lalu menjawab dengan jawaban yang sesuai.

5. Simulasi Tes TOEFL bagian *Structure*

Pada halaman Simulasi Tes TOEFL *structure* peserta kursus dapat melihat soal beserta pilihan ganda nya. Untuk sesi *structure* ini soal nya berbentuk teks. Pengguna bisa menjawab soal tersebut dengan klik pilihan ganda yang telah di sediakan pada halaman yang bisa di lihat pada gambar 7 berikut ini.



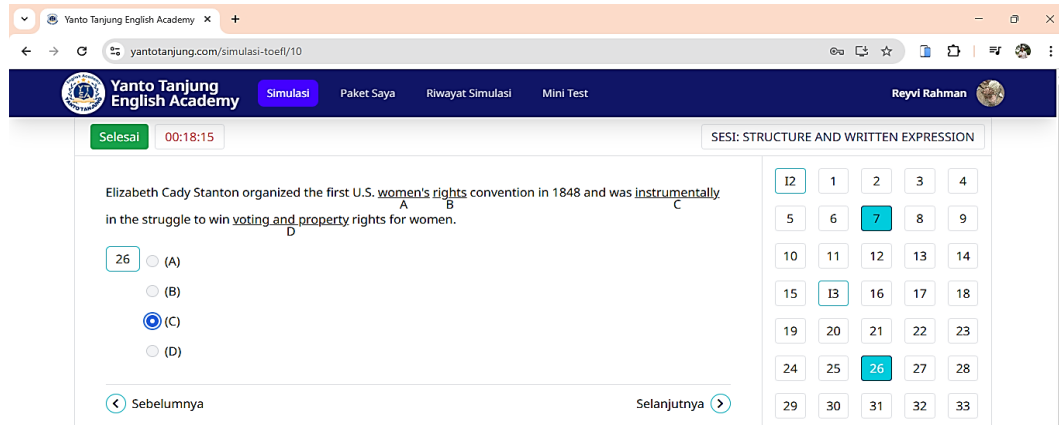
Gambar 7. Halaman Simulasi Tes TOEFL bagian *Structure*



Gambar 7 menunjukkan simulasi tes TOEFL bagian structure. Pada halaman ini pengguna dapat menjawab soal dengan memilih pilihan ganda yang telah tersedia pada soal nya. Tiap soal terdiri dari 4 pilihan ganda. Untuk soal structure peserta dapat membaca soal terlebih dahulu lalu menjawab dengan jawaban yang sesuai.

6. Simulasi Tes TOEFL bagian *Written Expression*

Pada halaman simulasi tes TOEFL *written expression* peserta kursus dapat melihat soal beserta pilihan ganda nya. Untuk sesi *written expression* ini soal nya berbentuk teks. Pengguna bisa menjawab soal tersebut dengan klik pilihan ganda yang telah di sediakan pada halaman yang bisa di lihat pada gambar 8 berikut ini.

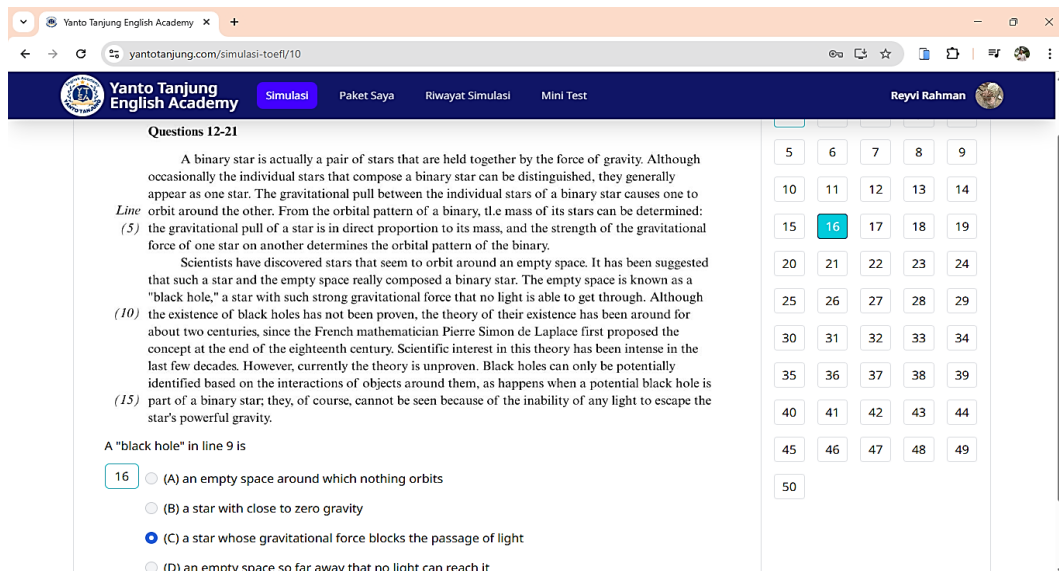


Gambar 8. Halaman Simulasi Tes TOEFL bagian *Written Expression*

Gambar 8 menunjukkan simulasi tes TOEFL bagian written expression. Pada halaman ini pengguna dapat menjawab soal dengan memilih pilihan ganda yang telah tersedia pada soal nya. Tiap soal terdiri dari 4 pilihan ganda. Untuk soal written expression peserta dapat membaca soal terlebih dahulu lalu menjawab dengan jawaban yang sesuai.

7. Simulasi Tes TOEFL Bagian *Reading Comprehension*

Pada halaman Simulasi Tes TOEFL *reading comprehension* peserta dapat melihat soal beserta pilihan ganda nya. Untuk sesi *reading comprehension* ini soal nya berbentuk gambar yang berisi soal dalam bahasa inggris. Pengguna bisa menjawab soal tersebut dengan klik pilihan ganda yang telah di sediakan pada halaman yang bisa di lihat pada Gambar 9 berikut ini.

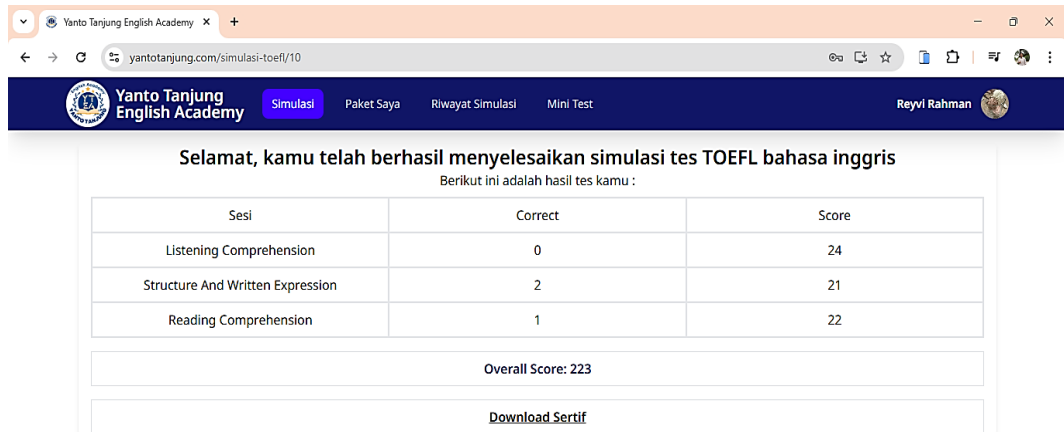


Gambar 9. Halaman Simulasi Tes TOEFL Bagian *Reading Comprehension*

Gambar 9 menunjukkan simulasi tes TOEFL bagian reading comprehension. Pada halaman ini pengguna dapat menjawab soal dengan memilih pilihan ganda yang telah tersedia pada soal nya. Tiap soal terdiri dari 4 pilihan ganda. Untuk soal written expression peserta dapat membaca soal terlebih dahulu lalu menjawab dengan jawaban yang sesuai.

8. *Score* Simulasi Tes TOEFL

Pada halaman *score* Simulasi Tes TOEFL peserta kursus dapat melihat *score* dari Simulasi Tes TOEFL yang telah di selesaikan. Di halaman ini juga ada tombol *download* sertif yang akan mengunduh sertifikat hasil simulasi tes TOEFL. Halaman ini bisa dilihat pada gambar 10 berikut ini.

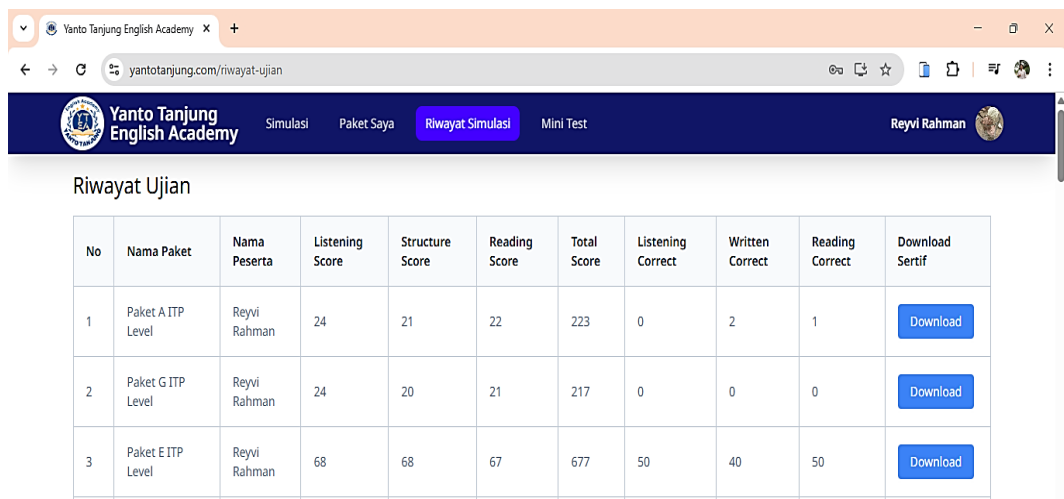


Gambar 10. Halaman Score Simulasi Tes TOEFL

Gambar 10 menunjukkan simulasi tes TOEFL bagian score simulasi tes toefl. Pada halaman ini pengguna dapat menjawab melihat hasil dari simulasi tes TOEFL yang telah dikerjakan.

9. Riwayat Simulasi Tes TOEFL

Pada halaman riwayat simulasi tes TOEFL ini peserta simulasi TOEFL bisa melihat seluruh riwayat ujian yang telah di lakukan sebelum nya. Di halaman ini juga tersedia tombol *download* untuk mendownload sertifikat simulasi tes TOEFL. Halaman Riwayat Simulasi Tes TOEFL dapat dilihat pada gambar 11 berikut ini.



Gambar 11. Halaman Riwayat Simulasi Tes TOEFL

Gambar 11 menunjukkan seluruh riwayat ujian yang telah dikerjakan pengguna dalam bentuk tabel. Pada tabel ini pengguna dapat melihat paket apa saja yang telah selesai di kerjakan.

b. *Continuous Integration*

Continuous Integration (CI) diterapkan untuk memastikan seluruh perubahan kode terdokumentasi dengan baik, tersimpan secara terpusat, dan dikelola secara sistematis. Implementasi CI menggunakan GitHub sebagai repositori utama proyek. Setiap selesai mengembangkan suatu fitur, dilakukan proses *commit* dan *push* ke GitHub agar kode tersimpan aman, riwayat perubahan tercatat rapi, serta memudahkan pemantauan dan pemeliharaan sistem.

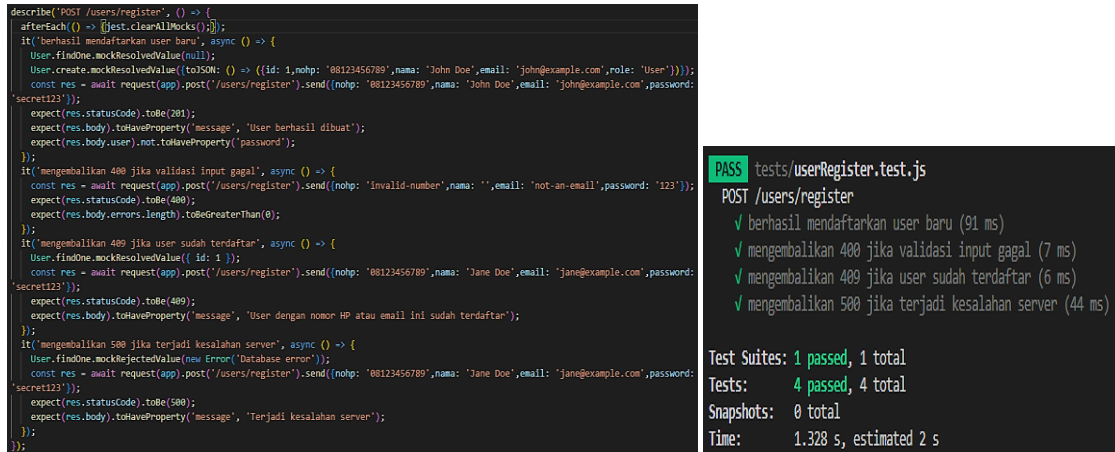
c. *Unit Testing*

Pengujian dilakukan secara otomatis dengan metode *Test Driven Development* yang dimulai dengan pembuatan tes dengan menuliskan skenario pengujian. Selanjutnya, tes dijalankan untuk melihat apakah hasilnya lulus atau gagal. Jika tes gagal atau menghasilkan *error*, maka dilakukan modifikasi atau *refactor* pada kode hingga pengujian berhasil tanpa *error*. Sebaliknya, jika tes berhasil, maka fitur dianggap siap untuk diimplementasikan.

1. Pengujian Fitur *Register*.

Pengujian dilakukan untuk memastikan proses registrasi pengguna berjalan dengan benar pada berbagai kondisi. Jika data valid dan belum pernah digunakan, sistem akan mendaftarkan pengguna baru dengan status 201. Apabila *input* tidak sesuai format, seperti nomor HP tidak *valid*, nama kosong, *email* tidak *valid*, atau *password* terlalu pendek, sistem akan menolak dengan status 400 dan menampilkan daftar *error*. Jika nomor HP atau *email* sudah terdaftar, sistem akan merespons dengan status 409. Sementara itu, apabila terjadi kesalahan pada *server*, seperti kegagalan *query database*, sistem akan memberikan respons status 500. Selanjut nya untuk menjalankan

pengujian, perintah `npm test` dieksekusi pada terminal dan menghasilkan pengujian yang sukses seperti yang bisa di lihat pada Gambar 12 berikut ini.



```
describe('POST /users/register', () => {
  afterEach(() => {jest.clearAllMocks();});
  it('berhasil mendaftarkan user baru', async () => {
    User.findOne.mockResolvedValue(null);
    User.create.mockResolvedValue({id: 1, nohp: '08123456789', nama: 'John Doe', email: 'john@example.com', role: 'User'});
    const res = await request(app).post('/users/register').send({nohp: '08123456789', nama: 'John Doe', email: 'john@example.com', password: 'secret123'});
    expect(res.statusCode).toBe(201);
    expect(res.body).toHaveProperty('message', 'User berhasil dibuat');
    expect(res.body.user).not.toHaveProperty('password');
  });
  it('mengembalikan 400 jika validasi input gagal', async () => {
    const res = await request(app).post('/users/register').send({nohp: 'invalid-number', nama: '', email: 'not-an-email', password: '123'});
    expect(res.statusCode).toBe(400);
    expect(res.body.errors.length).toBeGreaterThan(0);
  });
  it('mengembalikan 409 jika user sudah terdaftar', async () => {
    User.findOne.mockResolvedValue({ id: 1 });
    const res = await request(app).post('/users/register').send({nohp: '08123456789', nama: 'Jane Doe', email: 'jane@example.com', password: 'secret123'});
    expect(res.statusCode).toBe(409);
    expect(res.body).toHaveProperty('message', 'User dengan nomor HP atau email ini sudah terdaftar');
  });
  it('mengembalikan 500 jika terjadi kesalahan server', async () => {
    User.findOne.mockRejectedValue(new Error('Database error'));
    const res = await request(app).post('/users/register').send({nohp: '08123456789', nama: 'Jane Doe', email: 'jane@example.com', password: 'secret123'});
    expect(res.statusCode).toBe(500);
    expect(res.body).toHaveProperty('message', 'Terjadi kesalahan server');
  });
});
```

```
PASS tests/userRegister.test.js
  POST /users/register
    ✓ berhasil mendaftarkan user baru (91 ms)
    ✓ mengembalikan 400 jika validasi input gagal (7 ms)
    ✓ mengembalikan 409 jika user sudah terdaftar (6 ms)
    ✓ mengembalikan 500 jika terjadi kesalahan server (44 ms)

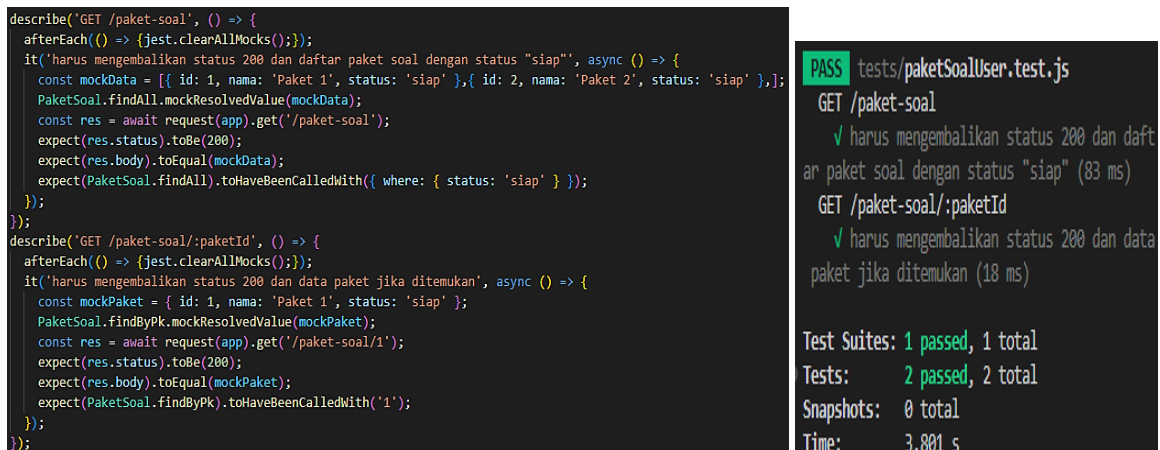
Test Suites: 1 passed, 1 total
Tests: 4 passed, 4 total
Snapshots: 0 total
Time: 1.328 s, estimated 2 s
```

Gambar 12. Skenario dan Hasil Pengujian Fitur Register

Gambar 12 menunjukkan hasil testing fitur yang telah di lakukan. Semua fungsi berhasil lolos uji dan tidak ada error yang ditemukan.

2. Pengujian Fitur Paket Soal

Pengujian pertama pada fungsi `GET /paket-soal` dilakukan untuk memastikan sistem hanya mengambil dan menampilkan paket soal berstatus `siap`. Contoh data yang diuji berisi dua paket, yaitu “Paket 1” dan “Paket 2”. Hasil yang diharapkan adalah respons status 200 dengan isi sesuai data contoh, serta verifikasi bahwa `query` ke `database` memang memfilter berdasarkan status `siap`. Pengujian kedua pada fungsi `GET /paket-soal/:paketId` bertujuan memeriksa pengambilan detail satu paket soal berdasarkan ID. Contoh pengujian menggunakan paket dengan ID 1 dan nama “Paket 1”. Hasil yang diharapkan adalah status 200 dengan data sesuai, serta pemanggilan metode `findByIdPk` menggunakan ID yang benar. Selanjut nya untuk menjalankan pengujian, perintah `npm test` dieksekusi pada terminal dan menghasilkan pengujian yang sukses seperti yang bisa di lihat pada gambar 13 berikut ini.



```
describe('GET /paket-soal', () => {
  afterEach(() => {jest.clearAllMocks();});
  it('harus mengembalikan status 200 dan daftar paket soal dengan status "siap", async () => {
    const mockData = [{ id: 1, nama: 'Paket 1', status: 'siap' }, { id: 2, nama: 'Paket 2', status: 'siap' }];
    PaketSoal.findAll.mockResolvedValue(mockData);
    const res = await request(app).get('/paket-soal');
    expect(res.status).toBe(200);
    expect(res.body).toEqual(mockData);
    expect(PaketSoal.findAll).toHaveBeenCalledWith({ where: { status: 'siap' } });
  });
});
describe('GET /paket-soal/:paketId', () => {
  afterEach(() => {jest.clearAllMocks();});
  it('harus mengembalikan status 200 dan data paket jika ditemukan', async () => {
    const mockPaket = { id: 1, nama: 'Paket 1', status: 'siap' };
    PaketSoal.findByIdPk.mockResolvedValue(mockPaket);
    const res = await request(app).get('/paket-soal/1');
    expect(res.status).toBe(200);
    expect(res.body).toEqual(mockPaket);
    expect(PaketSoal.findByIdPk).toHaveBeenCalledWith('1');
  });
});
```

```
PASS tests/paketSoalUser.test.js
  GET /paket-soal
    ✓ harus mengembalikan status 200 dan daftar paket soal dengan status "siap" (83 ms)
  GET /paket-soal/:paketId
    ✓ harus mengembalikan status 200 dan data paket jika ditemukan (18 ms)

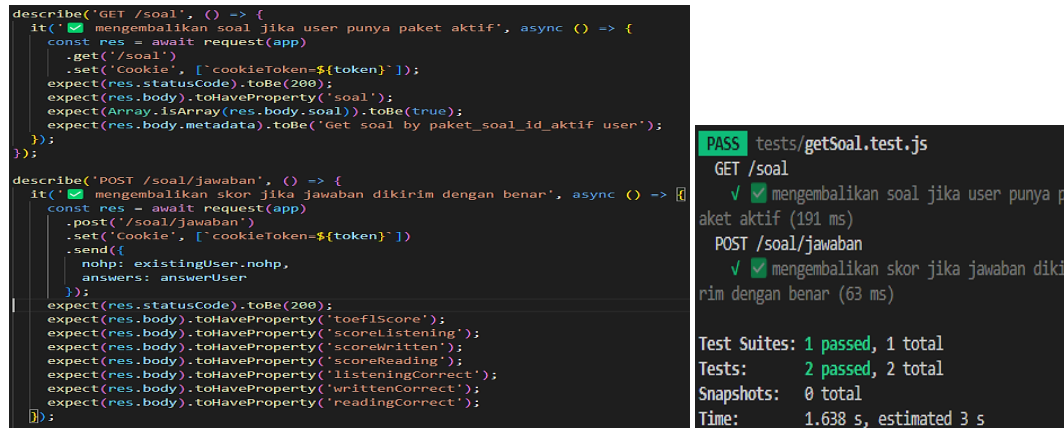
Test Suites: 1 passed, 1 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 3.801 s
```

Gambar 13. Skenario dan Hasil Pengujian Fitur Paket Soal

Gambar 13 menunjukkan hasil testing fitur yang telah di lakukan. Semua fungsi berhasil lolos uji dan tidak ada error yang ditemukan.

3. Pengujian Fitur Simulasi Tes TOEFL

Pengujian `GET /soal` dilakukan dengan menggunakan `user` terdaftar dan token autentikasi yang dikirim sebagai `cookie cookieToken` untuk membuat sesi login. Sistem diharapkan merespons dengan status 200, mengembalikan data soal dalam bentuk `array`, serta menyertakan properti metadata bernilai `"Get soal by paket_soal_id_aktif user"` yang menandakan soal diambil dari paket aktif milik user. Pengujian fungsi `POST /soal/jawaban` memeriksa proses pengiriman jawaban user dan perhitungan skor TOEFL. Respons yang diharapkan adalah status 200 dengan properti `toeflScore`, `scoreListening`, `scoreWritten`, `scoreReading`, serta jumlah jawaban benar (`listeningCorrect`, `writtenCorrect`, `readingCorrect`) untuk tiap bagian tes. Pengujian ini memastikan mekanisme penilaian dan autentikasi `server` berjalan sesuai harapan. Selanjut nya untuk menjalankan pengujian, perintah `npm test` dieksekusi pada terminal dan menghasilkan pengujian yang sukses seperti yang bisa di lihat pada Gambar 14 berikut ini.

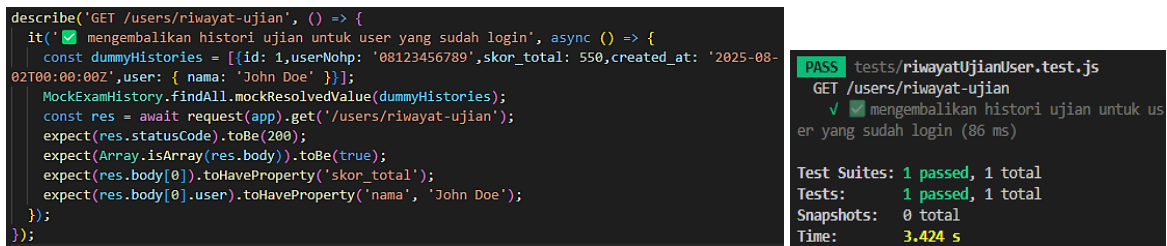


Gambar 14. Skenario dan Hasil Pengujian Fitur Simulasi Tes TOEFL

Gambar 14 menunjukkan hasil testing fitur yang telah dilakukan. Semua fungsi berhasil lolos uji dan tidak ada error yang ditemukan.

4. Pengujian Fitur Riwayat Simulasi *User*

Pengujian fungsi *GET /users/riwayat-ujian* memastikan sistem dapat mengembalikan riwayat ujian peserta dengan benar. Menggunakan *MockExamHistory.findAll*, diuji pengembalian *array* berisi satu entri dengan properti *id*, *userNohp*, *skor_total*, *created_at*, serta data user yang memuat nama pengguna. Hasil yang diharapkan adalah respons status 200 dengan *array* berisi setiap item yang memiliki *skor_total* dan user sesuai. Selanjutnya untuk menjalankan pengujian, perintah *npm test* dieksekusi pada terminal dan menghasilkan pengujian yang sukses seperti yang bisa dilihat pada Gambar 15 berikut ini.



Gambar 15. Skenario dan Hasil Pengujian Fitur Riwayat Simulasi *User*

Gambar 15 menunjukkan hasil testing fitur yang telah dilakukan. Semua fungsi berhasil lolos uji dan tidak ada error yang ditemukan.

d. *Stakeholder Approval*

Stakeholder approval melibatkan stakeholder utama yang bertugas mengevaluasi apakah fitur-fitur yang telah dikembangkan telah berfungsi sesuai dengan kebutuhan user. Evaluasi ini juga menjadi dasar untuk menentukan apakah pengembangan dapat dilanjutkan ke tahap *small release*, atau apakah diperlukan penambahan fitur baru sesuai masukan dari stakeholder. Selain itu, proses ini juga memastikan bahwa sistem yang dikembangkan tetap relevan dengan tujuan awal proyek. Daftar fitur yang diperlihatkan kepada stakeholder disajikan pada tabel 3 berikut ini.

Tabel 3. *Stakeholder Approval*

<i>User Story</i>	Fitur	Proses	Hasil	Catatan
Sebagai user, saya ingin membuat akun agar bisa mengikuti simulasi	Registrasi Akun	<i>User</i> mengisi data diri dan menekan tombol register	Akun berhasil dibuat	Disetujui
Sebagai user, saya ingin mengerjakan Simulasi <i>Listening</i>	Simulasi <i>Listening</i>	<i>User</i> membuka halaman simulasi dan memulai tes <i>Listening</i>	Soal tampil, jawaban bisa dikirim	Disetujui
Sebagai user, saya ingin mengerjakan Simulasi <i>Structure</i>	Simulasi <i>Structure</i>	<i>User</i> membuka halaman simulasi dan memulai tes <i>Structure</i>	Soal tampil, jawaban bisa dikirim	Disetujui
Sebagai <i>user</i> , saya ingin mengerjakan Simulasi <i>Reading</i>	Simulasi <i>Reading</i>	<i>User</i> membuka halaman simulasi dan memulai tes <i>Reading</i>	Soal tampil, jawaban bisa dikirim	Disetujui
Sebagai <i>user</i> , saya ingin melihat skor terakhir simulasi saya	Skor Simulasi	<i>User</i> membuka halaman skor simulasi	Skor terakhir tampil	Disetujui
Sebagai <i>user</i> , saya ingin melihat	Riwayat Skor	<i>User</i> membuka halaman	Daftar skor	Disetujui

User Story	Fitur	Proses	Hasil	Catatan
seluruh riwayat skor saya	Simulasi	riwayat skor	seluruh tes tampil	

Berdasarkan hasil pada Tabel 3, seluruh fitur telah memperoleh status disetujui. Capaian ini menunjukkan bahwa sistem telah berjalan sesuai dengan harapan *stakeholder* serta mampu memenuhi kebutuhan dari sisi peserta simulasi TOEFL. Selain itu, selama proses evaluasi, pihak *stakeholder* tidak memberikan permintaan penambahan fitur. Hal ini mengindikasikan bahwa fitur-fitur yang telah dikembangkan sudah dianggap lengkap dan sesuai dengan kebutuhan peserta tes simulasi TOEFL. Dengan demikian, proses pengembangan sistem pada tahap ini dinyatakan berhasil dan dapat dilanjutkan ke tahap *Small Release*.

e. *Small Release*

Setelah seluruh fitur pada iterasi berhasil melewati tahap *Stakeholder Approval*, proses selanjutnya adalah *Small Release*, yang dilakukan sebagai bentuk deploy awal dalam skala terbatas. Sistem telah diunggah ke server dan dapat diakses secara daring, namun belum dibuka untuk umum karena belum diindeks oleh mesin pencari seperti Google. Artinya, hanya pengguna yang menerima tautan secara langsung yang dapat mengakses dan menggunakan sistem ini. Tujuan utama dari *Small Release* ini adalah untuk memastikan bahwa sistem dapat beroperasi dengan baik dalam lingkungan nyata, sekaligus memfasilitasi uji coba langsung oleh pengguna tanpa menimbulkan eksposur publik yang luas. Gambar 16 berikut ini merupakan percobaan untuk mengakses web simulasi tes TOEFL yang sudah di deploy dalam skala kecil.

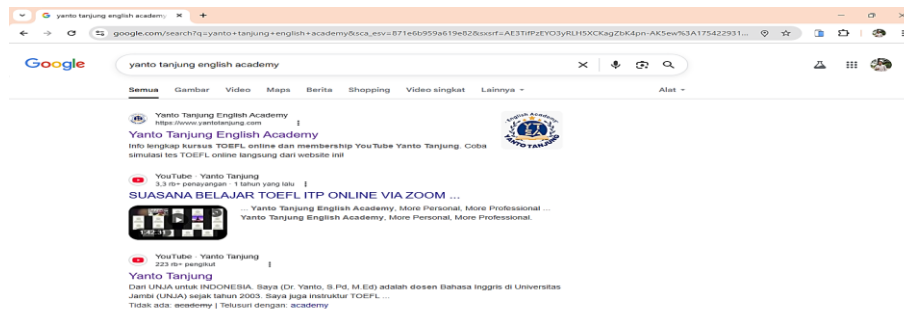


Gambar 16. Tahapan *Small Release*

Gambar 16 menunjukkan bahwa website sudah di-deploy dalam skala kecil. Website dapat diakses dengan baik dan tidak terlihat adanya error, sehingga menandakan proses deployment berjalan lancar. Hal ini juga menjadi indikasi awal bahwa sistem siap untuk diuji lebih lanjut oleh pengguna. Melalui tahap ini, tim pengembang dapat memantau performa sistem secara langsung di lingkungan nyata. Selain itu, hasil uji coba awal ini akan menjadi bahan evaluasi penting sebelum sistem dirilis dalam skala yang lebih luas.

f. *Final Release*

Setelah melalui seluruh tahapan pengujian, sistem simulasi tes TOEFL dinyatakan siap untuk dirilis secara penuh (*final release*). Seluruh fungsi utama telah berjalan dengan baik dan tidak ditemukan kendala kritis dalam penggunaannya. Sistem telah memenuhi aspek-aspek penting dalam pengembangan perangkat lunak, seperti fungsionalitas, kenyamanan penggunaan, dan keandalan. Dalam tahap rilis final, sistem dibuka untuk publik dan diatur agar dapat diindeks oleh mesin pencari seperti Google. Hal ini bertujuan agar sistem lebih mudah ditemukan oleh pengguna yang membutuhkan layanan simulasi tes TOEFL secara online. Gambar 17 berikut menunjukkan hasil pencarian di Google yang menampilkan sistem telah tersedia dan dapat diakses oleh masyarakat secara luas.

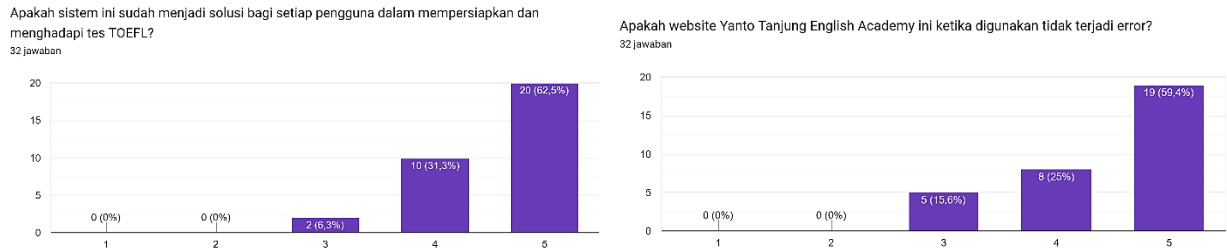


Gambar 17. *Final Release* Website Simulasi Tes TOEFL

Gambar 17 menunjukkan bahwa tahapan *Final Release* telah berhasil dilakukan. Setelah sistem masuk ke tahap *Final Release*, penulis melakukan validasi dengan menyebarkan kuesioner kepada sejumlah peserta kursus yang ditunjuk sebagai pengguna uji coba. Kuesioner ini berisi pertanyaan terkait kemudahan akses, tampilan, serta fungsionalitas website.

Instrumen kuesioner menggunakan skala Likert 1–5, di mana nilai 1 menyatakan *sangat tidak setuju* dan nilai 5 menyatakan *sangat setuju*. Dengan demikian, semakin tinggi skor yang dipilih peserta, semakin positif pula penilaian mereka terhadap sistem.

Para peserta kemudian mencoba langsung mengakses website, menggunakan fitur yang tersedia, lalu mengisi kuesioner sesuai pengalaman mereka. Jawaban dari kuesioner tersebut dikumpulkan dan diolah untuk mendapatkan gambaran objektif mengenai performa sistem. Hasil pengisian kuesioner ditampilkan pada Gambar 18 dalam bentuk diagram/grafik, sehingga terlihat dengan jelas tingkat kepuasan dan tanggapan pengguna.



Gambar 18. Hasil Uji Coba Pengguna

Dari hasil uji coba pada Gambar 18 terlihat bahwa semua pertanyaan memperoleh penilaian yang cenderung positif. Pada pertanyaan “Apakah sistem ini sudah menjadi solusi bagi setiap pengguna dalam mempersiapkan dan menghadapi tes TOEFL?” mayoritas responden memberikan nilai tinggi, yaitu 62,5% memilih skala 5 (sangat setuju) dan 31,3% memilih skala 4 (setuju). Hanya sebagian kecil (6,3%) yang memilih skala 3, sementara tidak ada responden yang memilih skala 1 maupun 2. Hal ini menunjukkan bahwa sebagian besar peserta menilai sistem benar-benar membantu mereka dalam persiapan tes TOEFL.

Pada pertanyaan “Apakah website Yanto Tanjung English Academy ini ketika digunakan tidak terjadi error?” hasilnya juga sangat baik. Sebanyak 59,4% responden memilih skala 5 (sangat setuju) dan 25% memilih skala 4 (setuju). Sementara itu, 15,6% memilih skala 3, dan tidak ada yang memberikan penilaian 1 atau 2. Dengan demikian, dapat disimpulkan bahwa sistem mampu berjalan dengan lancar, minim error, dan dapat diakses dengan aman oleh pengguna.

Secara keseluruhan, grafik pada Gambar 18 memperlihatkan bahwa sistem yang dikembangkan telah memenuhi kebutuhan pengguna, baik dari sisi fungsi sebagai solusi persiapan TOEFL maupun dari sisi stabilitas website yang aman digunakan.

4. KESIMPULAN

Berdasarkan hasil penelitian mengenai Implementasi Model Extreme Programming dalam Pengembangan Sistem Simulasi Tes TOEFL, dapat disimpulkan bahwa sistem berhasil dikembangkan sesuai kebutuhan dan mampu menjawab permasalahan utama, yaitu ketiadaan fasilitas simulasi TOEFL yang terstruktur dan interaktif. Seluruh tahapan XP, mulai dari *User Stories*, *Iteration Planning*, *Design*, *Coding*, *Continuous Integration*, *Unit Testing*, *Stakeholder Approval*, *Small Release* hingga *Final Release* dapat dilaksanakan dengan baik. Penelitian ini melibatkan iterasi yang menghasilkan fitur untuk peserta simulasi TOEFL. Proses *Test Driven Development* (TDD) juga berjalan optimal, mencakup empat skenario pengujian dengan seluruhnya *passed* tanpa kegagalan fungsi, sehingga sistem dinyatakan siap digunakan. Dalam penelitian terkait implementasi sistem simulasi tes TOEFL berbasis web untuk Yanto Tanjung English Academy, penulis menyadari adanya sejumlah keterbatasan yang disebabkan oleh faktor waktu, tenaga, serta keterbatasan sumber daya. Oleh karena itu, pada pengembangan selanjutnya disarankan untuk melakukan penyempurnaan dan optimalisasi sistem, salah satunya dengan melengkapi fitur pembahasan soal yang dapat diakses oleh peserta setelah menyelesaikan ujian. Fitur ini diharapkan memberikan umpan balik edukatif, memungkinkan peserta meninjau kembali soal yang telah dijawab, melihat jawaban yang benar, serta memahami alasan di balik setiap jawaban tersebut. Dengan demikian, peserta tidak hanya mengetahui skor akhir, tetapi juga memperoleh wawasan tambahan yang mendukung pembelajaran mandiri dan mempersiapkan mereka lebih baik dalam menghadapi tes TOEFL sesungguhnya.

REFERENCES

- Amdri Rizal, M., Ahmad, I., Damayanti, Aftirah, N., & Lestari, W. (2022). Aplikasi Inventory Persediaan Barang Berbasis Web Menggunakan Metode Extreme Programming (Studi Kasus: Esha 2 Cell). *TELEFORTECH: Journal of Telematics and Information Technology*, 3(2), 2774–5384. <https://doi.org/10.33365/tft.v3i2.2513>
- Dharmawan, E. A. (2023). Perancangan Sistem Informasi Geografis Penyebaran Daerah Zonasi Mangrove Di Pulau Ambon. *Jurnal ELKO (Elektrikal Dan Komputer)*, 4(1), 283–290. <https://doi.org/10.54463/je.v4i1.75>
- Endang, & Velia Susanti, M. (2021). Penerapan Metode Waterfall Pada Rancang Bangun Aplikasi P2db Di Smk



- Dewantara Cikarang Menggunakan Php Dan Mysql. *Jurnal Informatika SIMANTIK*, 6(2), 37–45. <https://doi.org/10.22141/2224-0721.16.4.2020.208486>
- Faizal, I., Nanda, I., Ariestiandy, D., & Ernawati, T. (2021). Pengembangan Sistem Informasi Pengelolaan Keuangan Bagi Usaha Mikro Kecil dan Menengah (UMKM). *Jurnal Sistem Komputer Dan Informatika (JSON)*, 3(2), 81–86. <https://doi.org/10.30865/json.v3i2.3590>
- Ferdiana Sari, E., & Ekohariadi. (2021). Penerapan Github Sebagai Media E-Learning Untuk Mengetahui Keefektifan Kolaborasi Project Pada Mata Pelajaran Pemrograman Web Dan Perangkat Bergerak Di Smk Negeri 2 Surabaya. *It-Edu*, 06(3), 14–22. <https://doi.org/doi.org/10.26740/it-edu.v6i3.43427>
- Hidayat, W. S., Ridwan, M., & Rismaningsih, F. (2024). Perancangan Sistem Absensi Menggunakan Qr Code Dan Geolokasi Berbasis Android Pada Cv Fahren Jaya. *Jutis (Jurnal Teknik Informatika Unis)*, 12(1), 56–68. <https://doi.org/10.33592/jutis.v12i1.4653>
- Iswati, L. (2021). Pelatihan TOEFL secara virtual bagi anggota Nasyyiatul ‘Aisyiah Cabang Mantrijeron Yogyakarta. *KACANEGARA Jurnal Pengabdian Pada Masyarakat*, 4(2), 247–254. <https://doi.org/10.28989/kacanegara.v4i2.959>
- Mujiyanto, T. R., & Maukar. (2021). Implementasi Aplikasi Pusat Pertukaran Data Pada Sistem Peradilan Pidana Terpadu Berbasis Teknologi Informasi (SPPT-TI). *Journal of Information System, Applied, Management, Accounting and Research.*, 5(2), 307–320. <https://doi.org/10.52362/jisamar.v5i2.428>
- Mutezar, A. A., & Salamah, U. (2021). Pengembangan Sistem Manajemen Event Pameran Karya Mahasiswa Menggunakan Metode Extreme Programming. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(4), 809–819. <https://doi.org/10.29207/resti.v5i4.3249>
- Rafi Nahjan, M., Ali Ridha, A., Heryana, N., & Voutama, A. (2023). Rancang Bangun Website Pencarian Informasi Berita Dan Cuaca Daerah Di Indonesia Menggunakan Api Dan Express.Js. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(5), 3309–3313. <https://doi.org/10.36040/jati.v7i5.7382>
- Rahmani, A.-. (2020). Studi Algoritma Babelrts Untuk Pelaksanaan Regression Test Selection (RST). *Jurnal Ilmiah Ilmu Dan Teknologi Rekayasa*, 3(2), 13–22. <https://doi.org/10.31962/jiitr.v3i2.108>
- Rahmi, E., Yumami, E., & Hidayasari, N. (2023). Analisis Metode Pengembangan Sistem Informasi Berbasis Website: Systematic Literature Review. *Remik: Riset Dan E-Jurnal Manajemen Informatika Komputer*, 7(1), 821–834. <https://doi.org/doi.org/10.33395/remik.v7i1.12177>
- Ramdany, S. W., Kaidar, S. A., Aguchino, B., Putri, C. A. A., & Anggie, R. (2024). Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web. *Journal of Industrial and Engineering System*, 5(1). <https://doi.org/10.31599/2e9afp31>
- Sarasvananda, I. B. G., Wiguna, I. K. A. G., & Styawati. (2021). Pendekatan Metode Extreme Programming untuk Pengembangan Sistem Informasi Manajemen Surat Menyurat pada LPIK STIKI. *Jurnal Informatika Universitas Pamulang*, 6(2), 258–267. <https://doi.org/10.32493/informatika.v6i2.9482>
- Siva, F., Assegaf, S. M. U., Pahlevi, S. A., & Yaqin, M. A. (2023). Survei Metode-Metode Software Development Life Cycle dengan Metode Systematic Literature Review. *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, 5(2), 36–52. <https://doi.org/10.28926/ilkomnika.v5i2.447>
- Sudewi, P. W., Imansari, N., & Putri, A. M. J. (2024). Pelatihan TOEFL ITP Untuk Meningkatkan Kemampuan Bahasa Inggris Siswa Sekolah Menengah Atas. *Jurnal Abdi Insani*, 11(1), 667–676. <https://doi.org/10.29303/abdiinsani.v11i1.1431>
- Sumiati, M., Abdillah, R., & Cahyo, A. (2021). Pemodelan UML untuk Sistem Informasi Persewaan Alat Pesta. *Jurnal Fasilkom*, 11(2), 79–86. <https://doi.org/10.37859/jf.v11i2.2673>
- Tri Sulistyorini, Sova, E., & Ramadhan, R. (2022). Pemantauan Kasus Penyebaran Covid-19 Berbasis Website Menggunakan Framework React Js Dan Api. *Jurnal Ilmiah Multidisiplin*, 1(04), 01–13. <https://doi.org/10.56127/jukim.v1i04.137>
- Wijanarko, S., & Subhiyanto. (2024). Penerapan Test Driven Development (TDD) pada Laravel Menggunakan PHPUnit. *Jurnal Sistem Informasi Stmik Antar Bangsa*, XIII(02), 60–63. <https://doi.org/10.51998/jsi.v13i2.574>
- Yoga, I. P. A. T., Wijaya, I. N. Y. A., & Putra, A. A. G. A. M. (2025). Rancang Bangun Sistem Informasi Pemetaan UMKM Menggunakan Metode Extreme Programming. *TIN: Terapan Informatika Nusantara*, 6(2), 75–84. <https://doi.org/10.47065/tin.v6i2.7839>