

Penerapan *Continuous Integration* Pada Sistem Manajemen Keanggotaan Gym Menggunakan Metode *Extreme Programming* (Studi Kasus: *Fortuna Fitness Sidoarjo*)

1st Elang Dewa Samudra
Sistem Informasi
Universitas Telkom Surabaya
Surabaya, Indonesia
dewaclangsamudra@gmail.com

2st Mochamad Nizar Palefi Ma'ady,
S.Kom., M.Kom., M.IM
Sistem Informasi
Universitas Telkom Surabaya
Surabaya, Indonesia
mnizarpm@telkomuniversity.ac.id

3st Purnama Anaking, S.Kom., M.Kom.
Sistem Informasi
Universitas Telkom Surabaya
Surabaya, Indonesia
purnamaanaking@telkomuniversity.ac.id

Abstrak—Dalam beberapa tahun terakhir, meningkatnya kesadaran masyarakat Indonesia terhadap pentingnya kesehatan dan kebugaran tercermin dari tingginya minat untuk berolahraga di pusat kebugaran seperti gym. Tren ini ditandai dengan bertambahnya jumlah gym di berbagai kota. Namun, sebagian besar pengelola gym, termasuk *Fortuna Fitness* masih menggunakan sistem manual dalam pengelolaan keanggotaan, pendaftaran, dan manajemen kelas, yang berpotensi menimbulkan risiko kesalahan pencatatan data dan ketidakakuratan laporan. Oleh karena itu, penelitian ini bertujuan untuk mengatasi permasalahan tersebut dengan melakukan pengembangan sistem menggunakan *Continuous Integration* (CI) dan *Extreme Programming* (XP). Metode ini dapat membantu proses pengembangan sistem yang terstruktur, dan adaptif terhadap perubahan kebutuhan. Penggunaan *Continuous Integration* memungkinkan kode diintegrasikan secara berkala untuk mengurangi risiko kesalahan. Dalam pengembangan perangkat lunak, *Extreme Programming* bermanfaat untuk memastikan perangkat lunak yang dihasilkan lebih responsif terhadap perubahan kebutuhan dengan pengembangan yang efisien. Pengujian dilakukan menggunakan metode *Blackbox Testing* yang menunjukkan bahwa seluruh fungsi sistem berjalan sesuai harapan. Kemudian, dilakukan *User Acceptance Testing* (UAT) dengan tiga indikator utama: fungsionalitas sistem (93%), pengalaman pengguna dan antarmuka (92%), serta efisiensi dan produktivitas (91%). Hasil pengujian menunjukkan bahwa sistem diterima dengan baik oleh pengguna. Penelitian ini diharapkan dapat membantu *Fortuna Fitness* dalam mengoptimalkan proses bisnis melalui penerapan sistem informasi yang lebih efektif dan efisien.

Kata kunci—*Continuous Integration*, Manajemen Keanggotaan Gym, Sistem Manajemen.

I. PENDAHULUAN

Dalam beberapa tahun terakhir, kesadaran akan pentingnya kesehatan semakin berkembang di Indonesia untuk pencegahan penyakit dan meningkatkan kualitas hidup secara keseluruhan dengan berolahraga yang dapat menjaga kesehatan, kebugaran, serta meningkatkan imunitas [1]. Seiring meningkatnya kesadaran ini, banyak masyarakat Indonesia mulai mengadopsi gaya hidup sehat dengan rutin berolahraga di gym, dimana berdasarkan laporan Statista

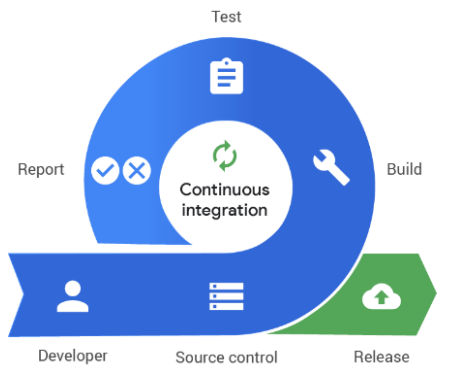
(2020), sebanyak 48,59% anggota gym secara rutin mengunjungi fasilitas beberapa kali dalam seminggu dan 31,12% lainnya hadir beberapa kali dalam sebulan. Kesadaran masyarakat yang tinggi terhadap kesehatan mendorong pertumbuhan bisnis gym di Indonesia secara signifikan karena semakin banyak orang menyadari pentingnya berolahraga untuk kesehatan fisik, mental, dan psikis, sehingga dengan banyaknya masyarakat yang rutin berolahraga di gym membuat manajemen keanggotaan menjadi kompleks dan diperlukan sistem otomatisasi pencatatan dan pengelolaan data agar gym dapat berjalan lebih efektif dan terstruktur.

Fortuna Fitness merupakan pusat kebugaran yang belum mengimplementasikan sistem untuk melakukan kegiatan operasional dimana proses pencatatan keanggotaan gym, pendaftaran, serta pengelolaan kelas masih dilakukan secara manual dengan 127 member, 9 kelas *fitness*, dan dikelola oleh 11 karyawan. Kondisi ini meningkatkan risiko kesalahan dalam pencatatan data serta ketidaktepatan laporan karena *human error* yang kerap muncul dalam proses penyusunan laporan keuangan saat melakukan rekapitulasi data transaksi dan kurangnya kemampuan penyusunan laporan keuangan oleh admin [2] serta membuat pemantauan dan pengelolaan jadwal kelas menjadi kurang efektif dan rawan kesalahan penjadwalan atau hilangnya data penting. Seiring perkembangan teknologi dan meningkatnya ekspektasi konsumen yang mengutamakan digitalisasi seperti pembayaran *online* dan penjadwalan daring [3], *Fortuna Fitness* perlu melakukan transformasi digital menggunakan sistem manajemen berbasis web yang telah menjadi standar industri [4]. Penelitian ini bertujuan menyelesaikan permasalahan tersebut menggunakan metode *Continuous Integration* dan *Extreme Programming* untuk mempercepat pengembangan dan menjamin pengujian otomatis yang dapat mengurangi waktu pembuatan dan pengujian [5] dimana XP diakui sebagai metode responsif terhadap perubahan yang memungkinkan proses iterasi berulang sesuai kebutuhan [6]. dan menerapkan pendekatan pengembangan sistem secara bertahap untuk merespons perubahan kebutuhan pelanggan dengan lebih efisien [7].

II. KAJIAN TEORI

Adapun teori – teori yang digunakan dalam penelitian ini sebagai pendukung penyelesaian masalah adalah sebagai berikut:

A. Continuous Integration (CI)



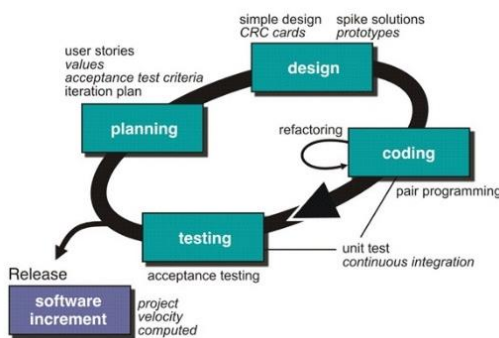
GAMBAR 1
Continuous Integration

Continuous Integration merupakan fase penting dalam pengembangan sistem, dimana proses pembangunan dan pengujian aplikasi dilakukan secara otomatis setiap kali ada pembaruan pada repositori guna memastikan integrasi perangkat lunak dan proses operasional berjalan lancar dan stabil. Penerapan *Continuous Integration* mendukung kelancaran pengembangan sistem dengan mengurangi bug dan masalah integrasi antar anggota tim, serta memberikan umpan balik cepat agar pengembang dapat segera melakukan perbaikan. Dengan demikian, metode ini terbukti efisien dalam meningkatkan mutu perangkat lunak serta mendorong sinergi antar anggota tim pengembang.

B. Jenkins

Jenkins merupakan *tools* yang sering digunakan dalam metode *Continuous Integration* dan *Continuous Delivery*. *Jenkins* adalah sebuah *server* otomatisasi sumber terbuka yang dirancang untuk melakukan otomatisasi terhadap aktivitas dalam *Continuous Integration and Delivery* perangkat lunak [8]. Kemampuan *Jenkins* dalam mendukung integrasi dan delivery berkelanjutan menjadikannya solusi yang efektif untuk mengoptimalkan dan mempermudah workflow pengembangan aplikasi [9]

C. Extreme Programming



GAMBAR 2
Extreme Programming

Extreme Programming adalah metode dalam pengembangan perangkat lunak agile yang berfokus pada

aktivitas pengkodean sebagai inti siklus pengembangan, dengan tujuan membentuk tim berukuran kecil hingga menengah yang mampu merespons ketidakpastian dan perubahan yang terjadi secara cepat dan dinamis. [6]. [10]. Pendekatan ini menekankan iterasi singkat, umpan balik berkelanjutan, serta keterlibatan intensif antara tim pengembang dan pengguna untuk menghasilkan perangkat lunak yang adaptif dan berkualitas tinggi. Langkah-langkah dalam *XP* meliputi:

1. Planning

Tahapan awal mencakup penelusuran kebutuhan sistem melalui perolehan informasi yang menjelaskan alur proses dan ekspektasi pengguna terhadap fungsi utama serta keluaran sistem

2. Design

Tahap perancangan melibatkan penyusunan gambaran sistem berdasarkan hasil analisis kebutuhan, dengan tujuan membuat hubungan antar data, merancang antarmuka yang sederhana, dan mudah dipahami untuk mendukung keberhasilan pengembangan sistem.

3. Coding

Tahapan selanjutnya adalah proses pengkodean, yaitu fase di mana pengembang mulai merealisasikan rancangan sistem ke dalam bentuk kode program yang terstruktur, berfokus pada kolaborasi tim, kualitas kode, serta adaptasi terhadap perubahan, guna menghasilkan prototipe perangkat lunak yang fungsional dan sesuai dengan tujuan yang telah ditetapkan.

4. Testing

Untuk memastikan perangkat lunak berjalan sesuai harapan, tahap pengujian dilakukan dengan meninjau fungsionalitas dan kinerja sistem melalui berbagai pendekatan untuk mendeteksi kesalahan sejak dini dan memastikan kesesuaian dengan spesifikasi yang telah ditentukan.

D. Framework Laravel

Laravel adalah salah satu *framework* yang dibangun dengan bahasa pemrograman *PHP*, yang dapat digunakan untuk mempermudah proses pengembangan situs web [11]. *Framework* ini dirancang untuk meningkatkan efektivitas dan produktivitas pengembang dengan menerapkan arsitektur *Model-View-Controller (MVC)* yang memisahkan logika aplikasi menjadi tiga komponen utama. *Model* sebagai pengelola data, *View* sebagai penyaji antarmuka pengguna, dan *Controller* sebagai penghubung antar keduanya.

E. Blackbox Testing

Pengujian *BlackBox* merupakan pendekatan evaluasi yang dilakukan berdasarkan fungsi yang diharapkan dari perangkat lunak tanpa melibatkan pengetahuan tentang struktur dalam atau alur logika internal sistem tersebut. Pengujian ini dilakukan dengan menjalankan program untuk mendeteksi adanya *error* dalam sistem yang selanjutnya diperbaiki agar sistem dapat dianggap layak digunakan [12].

F. User Acceptance Testing (UAT)

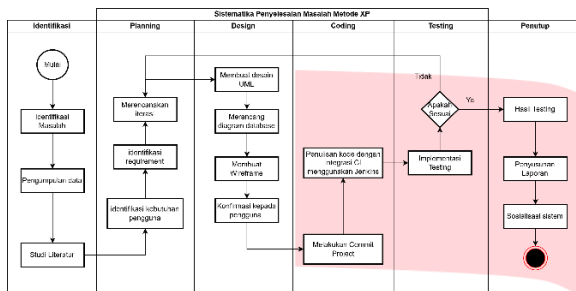
User Acceptance Testing dilakukan untuk memastikan perangkat lunak memenuhi kebutuhan pengguna, mengidentifikasi masalah yang belum terdeteksi sebelumnya, dan memberikan persetujuan akhir sebelum sistem diterapkan. UAT memusatkan perhatian pada tiga faktor

utama dalam menilai kelayakan sistem, yaitu: (1) fungsionalitas sistem, (2) kinerja sistem, dan (3) pengalaman serta desain antarmuka pengguna [13].

G. Unified Model Language

Unified Modeling Language (UML) adalah metode pemodelan visual yang digunakan untuk merancang sistem berorientasi objek dan pertama kali diperkenalkan dengan versi awal pada tahun 1997 oleh *Object Management Group*. Model diagram proses yang dapat dikembangkan oleh *UML* meliputi *use Case diagram*, *activity diagram*, dan lain – lain.

III. METODE



GAMBAR 3 Sistematisa Metode Penyelesaian

A. Identifikasi

Pada tahap identifikasi, peneliti menggali permasalahan melalui wawancara langsung, yang mengungkap bahwa *Fortuna Fitness* belum memiliki sistem manajemen keanggotaan terintegrasi, lalu dilanjutkan dengan penelusuran literatur guna menghimpun dasar teori yang mendukung perancangan sistem.

B. Planning

Pada tahap perencanaan, kebutuhan pengguna digali melalui wawancara dan penyusunan user story, kemudian dianalisis menjadi requirement fungsional yang diurai menjadi fitur-fitur prioritas berdasarkan urgensi, lalu dirancang ke dalam iterasi pengembangan sistem.

C. Design

Pada tahap perancangan, pengembangan perangkat lunak difokuskan pada pembuatan diagram *UML* dan rancangan antarmuka pengguna untuk memvisualisasikan alur sistem dan interaksi secara menyeluruh, membentuk kerangka kerja yang terstruktur sebagai dasar pengembangan tahap selanjutnya.

D. Coding

Pada tahap pengkodean, pengembang memulai dengan inisiasi proyek, lalu mengimplementasikan desain sistem melalui penulisan kode menggunakan *Laravel* untuk struktur *backend* yang efisien dan *Bootstrap* untuk tampilan antarmuka yang responsif, serta memastikan aplikasi ramah perangkat *mobile*. Proses ini juga didukung dengan penerapan *Continuous Integration* menggunakan *Jenkins* agar pengujian dan integrasi kode berlangsung otomatis dan konsisten.

E. Testing

Pada tahap pengujian, sistem diuji untuk memastikan seluruh fitur berfungsi sesuai harapan pengguna, menggunakan metode *blackbox testing* yang berfokus pada keluaran dari input tertentu. Selain itu, dilakukan *User Acceptance Testing (UAT)* oleh pengguna akhir guna menilai kesesuaian sistem dengan kebutuhan bisnis, dan hasil

akhirnya berupa persetujuan pengguna sebagai tanda bahwa aplikasi siap diterapkan sepenuhnya.

F. Penutup

Tahap penutup merupakan fase akhir dari penelitian. Setelah pengembangan dan pengujian aplikasi selesai, penulis menyusun laporan yang merangkum seluruh proses, mulai dari identifikasi masalah hingga implementasi dan pengujian. Selain itu, disiapkan video panduan sebagai media pengenalan sistem untuk memudahkan pemahaman pengguna terhadap cara kerja aplikasi yang telah dikembangkan.

IV. HASIL DAN PEMBAHASAN

Perancangan sistem manajemen keanggotaan dan layanan kebugaran di *Fortuna Fitness* diawali dengan pengumpulan data melalui wawancara kepada *stakeholder* utama. Proses ini bertujuan untuk memahami alur kerja serta kebutuhan informasi dari masing-masing peran, yang selanjutnya dirangkum sebagai dasar perancangan sistem.

A. Pengumpulan Data

Untuk merancang sistem yang mendukung pengelolaan keanggotaan dan layanan kebugaran di *Fortuna Fitness*, dilakukan pengumpulan data secara terarah dari pihak-pihak terkait. Data diperoleh melalui wawancara untuk memahami alur kerja dan kebutuhan informasi tiap peran, lalu dirangkum menjadi kebutuhan masing-masing aktor sebagai perancangan sistem manajemen *Fortuna Fitness*.

TABEL 1
Pengumpulan Data

No.	Aktor	Data yang Diperlukan	Keterangan
1.	User	- Informasi membership - Jadwal kelas <i>fitness</i> - Daftar <i>personal trainer</i>	Member dapat melihat jadwal kelas, status keanggotaan, dan daftar <i>personal trainer</i>
2.	Admin	- Data pengguna (ID, nama, <i>email</i>, status <i>membership</i>) - Data kelas kebugaran (jadwal, jenis, kapasitas) - Data <i>personal trainer</i> dan jadwal	Admin dapat mengelola informasi member, mengatur jadwal dan jenis kelas kebugaran, serta mengelola data dan penjadwalan <i>personal trainer</i>
3.	Owner	- Data admin - Data jumlah member, kelas <i>fitness</i>, dan <i>personal trainer</i>	Owner dapat mengelola akun admin, melihat data operasional secara menyeluruh.

B. Pengolahan Data

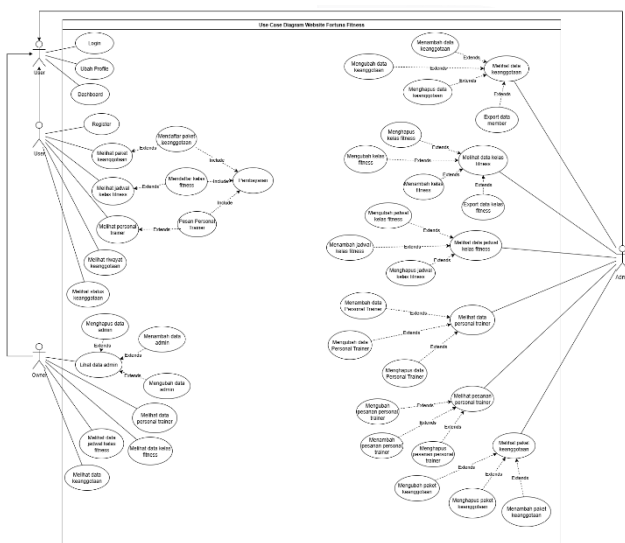
Pada tahapan pengolahan data yang dilakukan setelah proses wawancara dengan pihak-pihak terkait yaitu *Admin*, *Owner*, dan *User*, diperoleh berbagai informasi penting yang menggambarkan kebutuhan sistem serta permasalahan yang dihadapi selama proses operasional aplikasi yang sedang berjalan untuk menjamin bahwa desain sistem yang dihasilkan memenuhi kebutuhan dari seluruh peran yang ada. Berdasarkan hasil wawancara tersebut, dilakukan identifikasi kebutuhan sistem untuk setiap

pengguna yaitu *Admin*, *Owner*, dan *User* yang kemudian dituangkan dalam bentuk *user story* untuk menggambarkan kebutuhan fungsional sistem dari perspektif pengguna akhir dan dijadikan sebagai dasar dalam merancang iterasi pengembangan website yang terstruktur dan sesuai dengan metodologi *Extreme Programming* yang diterapkan

C. Perancangan Sistem

Bab ini mengulas tahap perancangan sistem berdasarkan hasil analisis kebutuhan sebelumnya. Proses dimulai dengan pembuatan *use case diagram* untuk menunjukkan hubungan aktor dengan fitur sistem. Kemudian, *Entity Relationship Diagram* digunakan untuk memetakan struktur data dan keterkaitannya. *Sequence diagram* turut disusun guna menggambarkan alur komunikasi antar objek secara runtut sesuai urutan waktu, membantu visualisasi proses dalam skenario tertentu.

Use case diagram digunakan untuk memetakan keterkaitan aktor dengan fungsi utama sistem secara visual. Representasi ini membantu tim pengembang memahami kebutuhan pengguna serta menyusun dokumen fungsional yang terarah. Selain menjadi panduan dalam menentukan prioritas fitur, diagram ini juga berfungsi sebagai media komunikasi efektif antar tim karena mampu menggambarkan alur interaksi secara ringkas namun informatif. Diagram berikut menunjukkan keseluruhan skenario *use case* dalam sistem



GAMBAR 3
Use Case Diagram

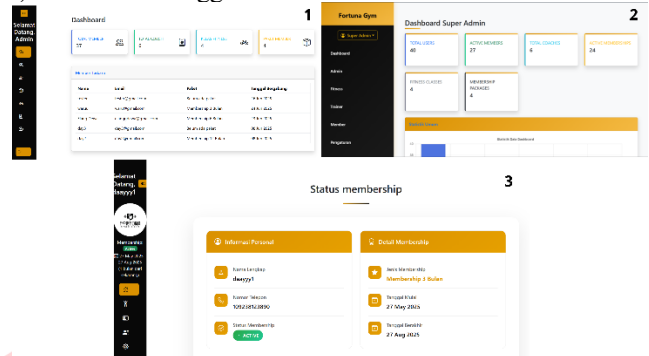
Gambar 3 memvisualisasikan *use case diagram* yang mendeskripsikan interaksi kompleks antaraktor dalam ekosistem aplikasi. Diagram ini mengrepresentasikan berbagai skenario fungsional seperti proses registrasi pengguna, mekanisme pembayaran transaksional, administrasi keanggotaan, serta pengelolaan privilege akun administrator.

D. Coding

Proses *coding* dilakukan berdasarkan hasil perancangan sistem yang telah disusun sebelumnya. Tahapan ini dimulai dengan membangun struktur folder proyek, konfigurasi *database*, dan implementasi antarmuka pengguna. Seluruh pengembangan disesuaikan dengan rancangan yang telah

dituangkan dalam *use case diagram*, *class diagram*, dan *sequence diagram*. Rangkaian fitur yang berhasil diimplementasikan dijabarkan sebagai berikut:

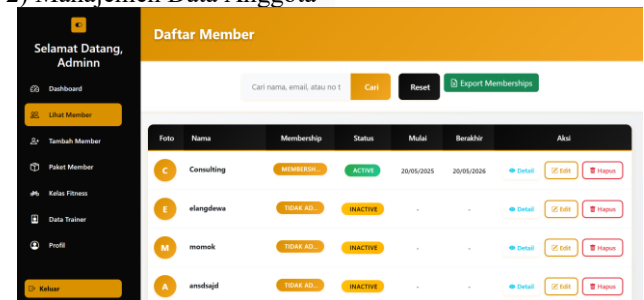
1) Dashboard Pengguna



GAMBAR 4
Tampilan Dashboard

Gambar 4 merupakan antarmuka *dashboard* yang disesuaikan dengan peran masing-masing. Admin berperan dalam pengelolaan data anggota, pelatih, dan paket keanggotaan. Owner memperoleh visual terkait keanggotaan dan kelas. Penyajian data melalui grafik dinamis mendukung pemantauan operasional dan strategi keputusan secara menyeluruh. User memiliki akses melihat status membership.

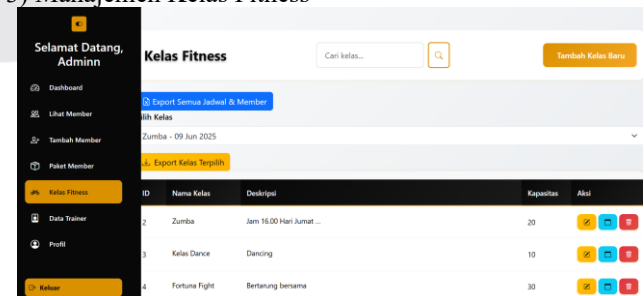
2) Manajemen Data Anggota



GAMBAR 5
Manajemen Data Anggota

Gambar 5 merupakan tampilan halaman manajemen data member yang dirancang khusus untuk admin. Melalui tampilan ini, admin memiliki akses untuk melakukan penambahan, perubahan, dan penghapusan data.

3) Manajemen Kelas Fitness

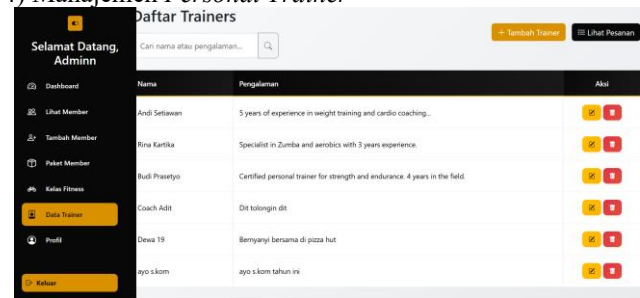


GAMBAR 6
Manajemen Kelas Fitness

Gambar 6 representasi antarmuka pengelolaan kelas fitness yang dikhususkan untuk admin, memberikan otoritas penuh dalam manajemen data kelas. Interface ini memfasilitasi admin untuk membuat kelas baru,

memodifikasi informasi kelas, serta menghapus kelas yang tidak diperlukan.

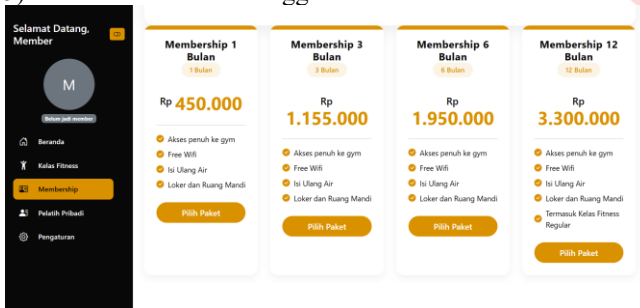
4) Manajemen *Personal Trainer*



GAMBAR 7
Manajemen Personal Trainer

Gambar 7 menampilkan antarmuka halaman pengelolaan personal trainer khusus untuk Admin. Pada halaman ini, Admin diberikan akses untuk menambahkan data pelatih baru, memperbarui informasi pelatih yang sudah ada, serta menghapus data jika diperlukan.

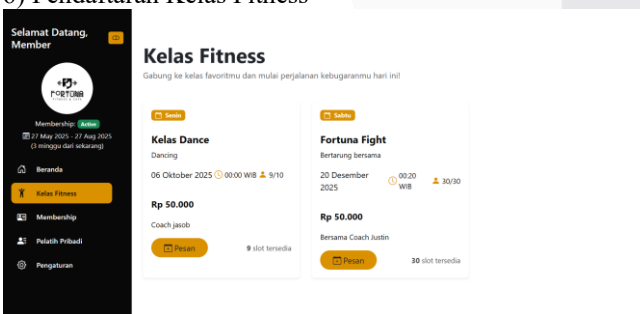
5) Pendaftaran Paket keanggotaan



GAMBAR 8
Pendaftaran Paket Keanggotaan

Pada Gambar 8 merupakan visualisasi *interface* keanggotaan *user* yang memfasilitasi pengguna untuk menyeleksi paket *membership*, melakukan registrasi, serta mengeksekusi transaksi pembayaran.

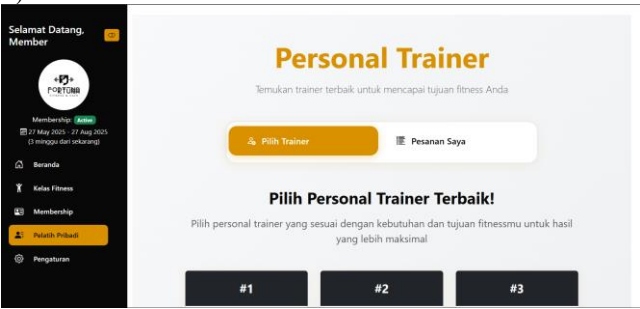
6) Pendaftaran Kelas Fitness



Gambar 9
Pendaftaran Kelas *Fitness*

Gambar 9 menampilkan antarmuka halaman User yang dirancang untuk memfasilitasi pemilihan kelas *fitness* sesuai minat, melakukan pembayaran secara langsung, serta meninjau riwayat kelas yang telah diikuti.

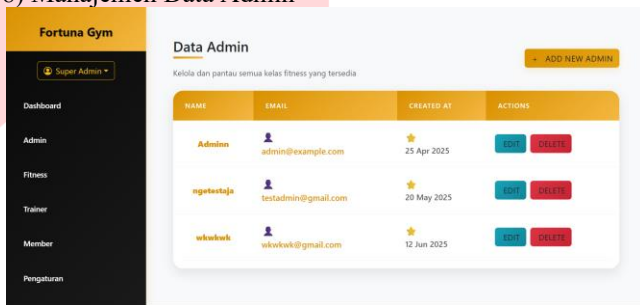
7) Pemesanan *Personal Trainer*



GAMBAR 10
Pemesanan *Personal Trainer*

Pada Gambar 10 tervisualisasi interface pemesanan personal trainer yang diperuntukkan bagi user, memfasilitasi mereka untuk menyeleksi instruktur, mengkonfigurasi jadwal temporal, serta mengeksekusi reservasi sesi pelatihan.

8) Manajemen Data Admin



Gambar 11
Manajemen Data Admin

Gambar 11 menampilkan *interface* pengelolaan data Admin yang diperuntukkan bagi Owner, memberikan otoritas komprehensif untuk menambah, mengubah, atau menghilangkan *record* Admin berdasarkan keperluan operasional sistem.

E. *BlackBox Testing*

Pada tahapan ini *blackbox testing* dilakukan dengan mengimplementasikan test case scenarios untuk memverifikasi fungsionalitas sistem tanpa mengakses kode internal dimana setiap komponen dievaluasi berdasarkan *input-output* untuk memastikan kesesuaian dengan kebutuhan pengguna yang telah ditetapkan sebelumnya.

TABEL 2
BlackBox Testing

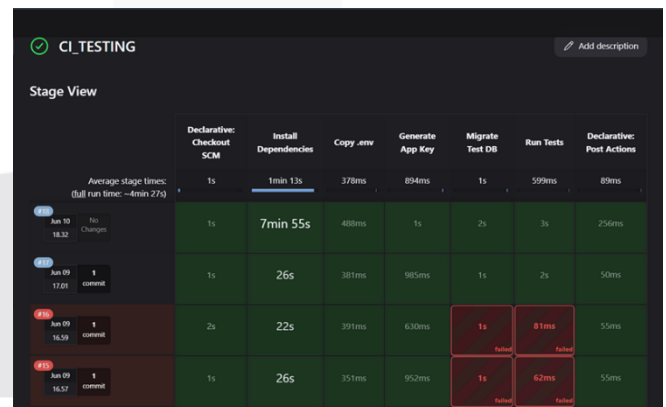
No.	Fitur	Test Case	Hasil Diharapkan	Hasil
1.	Login	Pengguna mencoba melakukan login menggunakan email dan password.	Berhasil Melakukan login menggunakan email dan password.	Valid
2.	Dashboar d Admin	Admin mengakses halaman dashboard dan melihat ringkasan informasi anggota, kelas fitness, dan personal trainer	Berhasil menampilkan ringkasan informasi anggota, kelas fitness, dan personal trainer.	Valid

3.	<i>Dashboard Owner</i>	Owner mengakses halaman <i>dashboard</i> dan melihat ringkasan informasi anggota, kelas <i>fitness</i> , dan <i>personal trainer</i>	Berhasil menampilkan ringkasan informasi anggota, kelas <i>fitness</i> , dan <i>personal trainer</i> .	Valid
4.	<i>Dashboard User</i>	Pengguna mengakses halaman <i>dashboard</i> dan melihat informasi terkait keanggotaan, kelas <i>fitness</i> , dan <i>personal trainer</i>	Halaman <i>dashboard</i> tampil dengan informasi yang benar dan sesuai dengan data	Valid
5.	Mendaftar Keanggotaan	Pengguna mendaftar untuk paket keanggotaan yang dipilih dan dilanjutkan dengan pembayaran	Pengguna berhasil melakukan pembayaran dan keanggotaan aktif	Valid
6.	Melihat Data Keanggotaan	Pengguna mengakses data anggota yang terdaftar dengan paket keanggotaan.	Data keanggotaan anggota berhasil ditampilkan dengan informasi yang benar.	Valid
7.	Menambah Data Keanggotaan	Admin menambah data keanggotaan baru untuk anggota dengan memilih paket yang sesuai.	Data keanggotaan berhasil ditambahkan dan terhubung dengan anggota.	Valid
8.	Mengubah Data Keanggotaan	Admin mengubah informasi keanggotaan anggota yang terdaftar, seperti status atau paket.	Data keanggotaan anggota berhasil diperbarui sesuai perubahan yang dilakukan.	Valid
9.	Menghapus Data Keanggotaan	Admin menghapus data keanggotaan anggota tertentu.	Data keanggotaan anggota berhasil dihapus dari sistem.	Valid
10.	Melihat Data <i>personal trainer</i>	Pengguna melihat daftar <i>personal trainer</i> yang terdaftar.	Daftar <i>personal trainer</i> berhasil ditampilkan dengan data yang benar.	Valid
11.	Menambah Data <i>personal trainer</i>	Admin menambah <i>personal trainer</i> baru dengan informasi seperti nama dan pengalaman	Data <i>personal trainer</i> baru berhasil ditambahkan ke dalam sistem.	Valid
12.	Mengubah Data <i>personal trainer</i>	Admin mengubah data <i>personal trainer</i> yang sudah ada (misalnya pengalaman).	Data <i>personal trainer</i> berhasil diperbarui sesuai perubahan yang dilakukan.	Valid

13.	Menghapus Data <i>personal trainer</i>	Admin menghapus data <i>personal trainer</i> yang terdaftar.	Data <i>personal trainer</i> berhasil dihapus dari sistem.	Valid
14.	Pesan <i>personal trainer</i>	Pengguna pesan <i>personal trainer</i> dan melakukan pembayaran	Melakukan pembayaran dan pesanan <i>personal trainer</i> berhasil	Valid
15.	Mendaftar kelas <i>fitness</i>	Pengguna mendaftar kelas <i>fitness</i> dan melakukan pembayaran untuk mengikuti kelas tersebut	Pengguna berhasil membayar dan berhasil mendaftar	Valid
16.	Melihat data admin	Owner dapat melihat daftar data admin yang terdaftar di sistem	Daftar admin ditampilkan dengan benar	Valid
17.	Menambah data admin	Owner dapat menambahkan data admin baru ke sistem	Data admin baru berhasil ditambahkan	Valid
18.	Menghapus data admin	Owner dapat menghapus data admin yang terdaftar di sistem	Data admin berhasil dihapus	Valid

F. Continuous Integration Pipeline

Pada tahapan awal konfigurasi, definisi *pipeline* ditentukan sebagai *Pipeline script from SCM* yang menginstruksikan *Jenkins* untuk menjalankan alur kerja otomatis berdasarkan skrip eksternal yang disimpan dalam repositori *Git* dengan *URL* mengarah ke proyek di *GitHub*. Skrip yang digunakan adalah *Jenkinsfile* yang terdokumentasi di *GitHub* pada branch *main*, mencakup proses *build*, *test*, dan *deploy*. Dengan konfigurasi ini, *Jenkins* dapat menarik skrip *pipeline* secara otomatis dari repositori eksternal setiap kali terjadi perubahan pada branch *main* sehingga seluruh proses integrasi dan pengujian dilakukan secara otomatis



GAMBAR 12
Hasil Continuous Integration

Pada Gambar diatas mempresentasikan proses penerapan alur *Continuous Integration (CI)*, di mana setiap langkah berkontribusi penting untuk menjamin aplikasi dapat dibangun dan diuji secara otomatis sebelum dijalankan di lingkungan lokal. Keberhasilan setiap tahapan menunjukkan bahwa pipeline mendukung otomatis pengembangan perangkat lunak yang andal.

G. User Acceptance Testing (UAT)

Dalam uji kelayakan sistem, diterapkan metode *User Acceptance Test (UAT)* dengan partisipasi langsung dari 30 pengguna akhir. Para partisipan memberikan tanggapan melalui instrumen survei berbasis skala Likert 1–5 guna menilai keselarasan fungsionalitas sistem dengan ekspektasi dan kebutuhan yang telah ditetapkan.

TABEL 3
Skala Likert

Bobot	Kode	Keterangan
1	TS	Tidak Setuju
2	KS	Kurang Setuju
3	CS	Cukup Setuju
4	S	Setuju
5	SS	Sangat Setuju

Penilaian pada angket kuesioner Mencakup tiga variabel yang digunakan dalam proses pengujian, yaitu fungsionalitas sistem, pengalaman & tampilan antarmuka sistem, dan efisiensi & produktivitas sistem. Tiga variabel yang digunakan dinilai cukup representatif untuk mengevaluasi keberfungsian dan kenyamanan sistem bagi pengguna.

Data yang diperoleh dari kuesioner dianalisis dengan menghitung rata-rata skor berdasarkan jawaban yang diberikan oleh setiap responden. Perhitungan bobot dilakukan dengan mengalikan jumlah jawaban dengan bobot penilaian pada skala likert. Berikut hasil perhitungan bobot untuk setiap pertanyaan

TABEL 4
Bobot Fungsionalitas Sistem

Pertanyaan	SS	S	CS	KS	TS	Jumlah
A1	105	36	0	0	0	141
A2	90	48	0	0	0	138
A3	90	48	0	0	0	138
A4	95	44	0	0	0	139
A5	105	36	0	0	0	141

TABEL 5
Pengalaman & Tampilan Antarmuka Sistem

Pertanyaan	SS	S	CS	KS	TS	Jumlah
B1	95	44	0	0	0	139
B2	90	48	0	0	0	138
B3	100	40	0	0	0	140
B4	90	48	0	0	0	138
B5	90	48	0	0	0	138

TABEL 6
Efisiensi & Produktivitas Sistem

Pertanyaan	SS	S	CS	KS	TS	Jumlah
C1	85	52	0	0	0	137
C2	70	64	0	0	0	134
C3	95	44	0	0	0	139
C4	65	68	0	0	0	133

C5	110	32	0	0	0	142
----	-----	----	---	---	---	-----

TABEL 7
Hasil Akhir Perhitungan UAT

No.	Variabel	Nilai Bobot (%)	Keterangan
1.	Persentase Fungsionalitas Sistem	93%	Sangat Baik
2.	Pengalaman & Tampilan Antarmuka Sistem	92%	Sangat Baik
3.	Efisiensi & Produktivitas Sistem	91%	Sangat Baik

Hasil penilaian terhadap sistem oleh 30 partisipan menunjukkan pencapaian yang optimal di tiga indikator utama. Pada indikator kapabilitas fungsional, sistem memperoleh skor sebesar 93%, yang mengindikasikan bahwa seluruh fitur bekerja selaras dengan ekspektasi pengguna. Selanjutnya, aspek kenyamanan visual dan kemudahan navigasi memperoleh nilai 92%, menandakan antarmuka sistem telah dirancang dengan pendekatan yang intuitif. Terakhir, efisiensi operasional serta kontribusi terhadap produktivitas pengguna mendapatkan skor 91%, merefleksikan performa sistem yang mendukung proses kerja secara optimal dan minim hambatan.

V. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat ditarik kesimpulan bahwa pengembangan sistem manajemen keanggotaan *gym Fortuna Fitness* berhasil dilakukan menggunakan framework *Laravel* dengan metode *Extreme Programming* yang memungkinkan proses pengembangan iteratif melalui tahapan perencanaan, desain, pengkodean, hingga pengujian berulang untuk menghasilkan sistem yang sesuai kebutuhan operasional manajemen *gym*. Implementasi *Continuous Integration* menggunakan *Jenkins* sebagai tools otomatisasi berperan penting dalam membangun *pipeline* integrasi untuk pengujian otomatis dan deteksi kesalahan sejak dini yang mampu meningkatkan efisiensi pengembangan serta memastikan setiap perubahan kode dapat diintegrasikan berkelanjutan tanpa mengganggu sistem utama. Pengujian sistem melalui *Blackbox Testing* dan *User Acceptance Testing (UAT)* dengan fokus pada fungsionalitas sistem, pengalaman antarmuka, serta efisiensi produktivitas menunjukkan hasil bahwa sistem telah memenuhi ekspektasi pengguna dan dapat diimplementasikan secara efektif dalam operasional manajemen keanggotaan *gym*.

REFERENSI

- [1] I. Zubaida, R. A. Fernanda, and W. W. N. Firdaus, "OLAHRAGA KESEHATAN : MEMASYARAKATKAN OLAHRAGA UNTUK PENINGKATAN KESEHATAN," *Journal of Sport Science and Tourism Activity (JOSITA)*, vol. 1, no. 1, pp. 15–21, 2022, doi: <http://dx.doi.org/10.62870/josita.v1i1.15422>.
- [2] H. Tehuayo, A. Martani, and A. L. Perdana, "Perancangan Sistem Data Keuangan dan Pendaftaran Member pada Vanlino GYM Menggunakan Visual Basic 2010," *Jurnal Multidisiplin Madani (MUDIMA)*, vol. 2, no. 1, pp.

- 443–456, 2022, [Online]. Available: <https://journal.y3a.org/index.php/mudima/index>
- [3] A. S. Khot, S. R. Lankal, V. N. Timmanagoudar, and M. V. S. Patil, “GYM MANAGEMENT SYSTEM,” *INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (INJRD)*, no. 10, Oct. 2024, [Online]. Available: www.ijnrd.org
- [4] S. Hadawale, A. Pawar, P. Bhajrawale, and S. Patange, “The Role of Web-Based Gym Management Systems in Enhancing User Experience,” *International Journal of Research Publication and Reviews Journal homepage: www.ijrpr.com*, vol. 6, no. 2, pp. 3448–3453, Feb. 2025, [Online]. Available: www.ijrpr.com
- [5] W. Widianara, G. P. Megayasa, L. Putu, and R. Noviana, “RANCANG BANGUN SISTEM CONTINUOUS INTEGRATION (CI) DAN CONTINUOUS DELIVERY (CD) MENGGUNAKAN JENKINS PADA APLIKASI BERBASIS WEB,” *Jurnal Manajemen dan Teknologi Informasi (JMTI)*, vol. 14, no. 1, pp. 36–39, 2024, doi: 10.59819.
- [6] M. Siahaan, “KLIK: Kajian Ilmiah Informatika dan Komputer Pengembangan Helpdesk Ticketing System berbasis Website dengan menggunakan metode SDLC, XP, dan Scrum,” *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 4, no. 3, pp. 1399–1410, 2023, doi: 10.30865/klik.v4i3.1400.
- [7] S. B. Atim, “Permodelan Sistem Informasi Penjualan Barang Berbasis Website Menggunakan Metode Agile,” *Journal of Artificial Intelligence and Technology Information (JAITI)*, vol. 2, no. 1, 2024, doi: 10.58602/jaiti.v2i1.104.
- [8] R. A. Putra, “ANALISA IMPLEMENTASI ARSITEKTUR MICROSERVICES BERBASIS KONTAINER PADA KOMUNITAS PENGEMBANG PERANGKAT LUNAK SUMBER TERBUKA (OPENDAYLIGHT DEVOPS COMMUNITY),” *Jurnal Sistem Informasi, Teknologi Informasi Dan Komputer*, vol. 9, 2019, [Online]. Available: <https://jurnal.umj.ac.id>
- [9] U. Sa’adah *et al.*, “FRAMEWORK TESTING OTOMATIS BERBASIS SERENITY DAN JENKINS AUTOMATED BUILD AUTOMATIC TESTING FRAMEWORK BASED ON SERENITY AND JENKINS AUTOMATED BUILD,” *JUTI: Jurnal Ilmiah Teknologi Informasi*, vol. 19, no. 2, p. 102, 2021, doi: 10.12962/j24068535.v19i2.a1017.
- [10] W. Rahmawanti and A. Rosemalatriasari, “IMPLEMENTASI METODE AGILE PADA PEMBUATAN E-REPORT LITTLE CLOUD BERBASIS WEB MENGGUNAKAN FRAMEWORK CODEIGNETER,” *JUIT*, vol. 2, no. 2, 2023.
- [11] R. I. Melyani, Rosita, and S. Aji, “Pengembangan Sistem Informasi Penggajian Berbasis Web Menggunakan Framework Laravel dengan Metode Agile Software Development,” *Jurnal Sistem Informasi Akuntansi*, vol. 03, no. 01, 2023, [Online]. Available: <http://jurnal.bsi.ac.id/index.php/jasika>
- [12] B. A. Maulana, E. Mawarni, M. Y. Hidayattuloh, V. Suryawijaya, and A. Saifudin, “Pengujian Black Box pada Sistem Informasi Barang Berbasis Web Menggunakan Metode Boundary Value Analysis,” *OKTAL : Jurnal Ilmu Komputer dan Science*, vol. 2, no. 6, 2023.
- [13] Aliyah, N. Hartono, and A. A. Muin, “Penggunaan User Acceptance Testing (UAT) Pada Pengujian Sistem Informasi Pengelolaan Keuangan Dan Inventaris Barang,” *Switch: Jurnal Sains dan Teknologi Informasi*, vol. 3, no. 1, pp. 84–100, 2025, doi: 10.62951/switch.v3i1.330.