

Rancang Bangun Sistem Penggalangan Dana Berbasis *Website* Menggunakan Metode *Extreme Programming*

1st Arif Al Hakimi Rifai
Universitas Telkom Purwokerto, Indonesia
arifalhakimi@student.telkomuniversity.ac.id

2nd Alon Jala Tirta Segara, S.Kom., M.Kom.
Universitas Telkom Purwokerto, Indonesia
alon@telkomuniversity.ac.id

Abstrak — Panti Asuhan Masjid Gelora Indah masih menyebarkan kotak donasi ke masjid-masjid dalam proses penggalangan dana. Hal ini menyebabkan keterbatasan jangkauan dan informasi, serta menyulitkan calon donatur atau dermawan dalam berdonasi. Sistem yang diusulkan bertujuan untuk mempermudah proses donasi donatur untuk berpartisipasi terhadap panti asuhan Masjid Gelora Indah dan juga menyebarkan informasi panti asuhan Masjid Gelora Indah. Penggalangan dana secara digital menjadi solusi yang relevan di era teknologi saat ini, di mana masyarakat menginginkan kemudahan dan kecepatan dalam melakukan donasi. Namun, belum banyak panti asuhan yang memanfaatkan teknologi informasi secara optimal. Kondisi ini menciptakan kesenjangan antara potensi penggunaan teknologi dan praktik yang berjalan saat ini. Solusi yang dikembangkan berupa sistem penggalangan dana berbasis *website* yang dibangun menggunakan Vite untuk *frontend* dan Express.js untuk *backend*. Sistem ini dilengkapi dengan fitur registrasi dan *login* pengguna, donasi online menggunakan Midtrans sebagai *payment gateway*, riwayat transaksi, *formulir* kontak, serta dashboard admin untuk mengelola data pengguna, donasi, pesan, dan ekspor data dalam bentuk Excel. Pengembangan dilakukan dengan metode *Extreme Programming* dan pengujian sistem menggunakan *Blackbox Testing* untuk memastikan semua fungsi berjalan dengan baik. Hasil pengujian menunjukkan bahwa seluruh fitur sistem berfungsi sesuai kebutuhan pengguna dan berjalan sebagaimana mestinya.

Kata kunci — Panti asuhan, *extreme programming*(XP), *website*

I. PENDAHULUAN

Saat ini di Indonesia masih banyak panti asuhan yang memiliki banyak keterbatasan finansial, fasilitas, dan afeksi mengingat panti asuhan merupakan tempat anak yang ditinggalkan satu bahkan kedua orang tuanya[1]. Selain itu, keluarga yang tidak mampu juga sering menitipkan anak-anak mereka di panti asuhan. Oleh karena itu, dukungan donasi dari berbagai pihak sangatlah penting untuk memenuhi kebutuhan anak-anak asuh, seperti makanan, pendidikan, biaya utilitas asrama, dan lain sebagainya. Di sini pembinaan anak sebagai bagian dari proses sosialisasi yang paling penting dan mendasar karena fungsi utama pembinaan anak adalah mempersiapkan anak menjadi warga masyarakat yang mandiri[2]. Tidak jarang, beberapa panti asuhan berupaya untuk menjadi lebih mandiri dalam menyokong anak-anak asuh mereka dan mengajarkan mereka untuk tidak selalu bergantung pada donasi. Salah satu cara yang dilakukan adalah melalui upaya kewirausahaan, baik yang

dilakukan secara individu maupun dengan bekerja sama dengan pihak lain.

Banyak panti asuhan di Indonesia salah satu sumber dana penting bagi panti asuhan adalah melalui situs-situs donasi online yang tersedia di internet. Sayangnya, saat ini, banyak dari situs-situs tersebut kurang transparan dan kurang efektif dalam melibatkan para dermawan secara aktif, sehingga mengurangi partisipasi dalam donasi. Para dermawan yang ingin memberikan sumbangan seringkali harus melakukan penelusuran sendiri tentang organisasi sosial atau panti asuhan tertentu, yang tentu saja mengurangi minat mereka untuk menyumbangkan waktu dan uang. Solusi yang efektif adalah sistem donasi yang lebih transparan. Dengan sistem ini, para dermawan tidak hanya akan memiliki akses ke informasi tentang penggunaan dana donasi, siapa yang menerima manfaatnya, dan dampaknya, tetapi juga akan diberikan *platform* interaktif yang memungkinkan mereka untuk tetap terlibat dan terhubung dengan panti asuhan secara langsung.

Sistem penyaluran donasi di Panti Asuhan Masjid Gelora Indah sebelumnya masih dilakukan secara konvensional, yaitu dengan menyalurkan kotak-kotak donasi di sekitar masjid. Berdasarkan hasil wawancara dengan Ketua Panti Asuhan Masjid Gelora Indah, Bapak Achmad Mulyono, S.H, ditemukan beberapa permasalahan utama, yaitu terbatasnya penyampaian informasi terkait profil dan kebutuhan panti asuhan dan juga terbatasnya pilihan metode pembayaran yang ada. Sehingga kondisi ini menyulitkan masyarakat untuk mengetahui informasi panti asuhan Masjid Gelora Indah dan juga sulit dalam melakukan donasi.

Untuk menjawab permasalahan tersebut, penelitian ini mengusulkan pembangunan sistem informasi berbasis *website* sebagai media informasi dan sarana donasi. Metode pengembangan perangkat lunak yang digunakan adalah *Extreme Programming* (XP), karena metode ini memiliki kelebihan dalam mengakomodasi perubahan kebutuhan secara fleksibel, berfokus pada keterlibatan pengguna secara langsung, serta cocok untuk tim kecil dengan sumber daya terbatas[3].

II. KAJIAN TEORI

A. Rancang Bangun

Rancang bangun merupakan suatu kegiatan yang bertujuan untuk merancang sistem baru yang mampu menyelesaikan permasalahan yang dihadapi oleh pengguna, berdasarkan hasil pemilihan sistem yang dianggap paling tepat. Istilah "bangun" merujuk pada proses pembangunan, yakni kegiatan menciptakan sistem baru atau melakukan perbaikan terhadap sistem yang sudah ada, baik secara menyeluruh maupun sebagian. Dengan demikian, rancang bangun dapat diartikan sebagai proses menerjemahkan hasil analisis ke dalam bentuk perangkat lunak yang terintegrasi, yang kemudian diwujudkan melalui pembuatan sistem baru atau penyempurnaan sistem yang telah ada sebelumnya[4].

B. Website

Website adalah sekumpulan informasi yang disajikan dalam bentuk halaman-halaman web yang saling terhubung satu sama lain. Informasi ini dapat disediakan oleh individu, kelompok, maupun organisasi untuk berbagai tujuan, baik informatif, komersial, maupun sosial[5]. Halaman-halaman yang berisi informasi dalam bentuk teks, gambar, video, atau elemen multimedia lainnya yang dapat diakses oleh pengguna melalui browser web.

C. Java Script

JavaScript adalah bahasa yang terdiri dari kumpulan kode yang berfungsi untuk dieksekusi pada dokumen HTML[6]. Pada awalnya JavaScript dikembangkan untuk berjalan di sisi klien pada halaman web. Kini JavaScript juga sudah bisa berjalan di sisi server dengan bantuan platform seperti nodejs.

D. MySQL

MySQL adalah sebuah sistem manajemen basis data (DBMS) yang menggunakan bahasa SQL (*Structured Query Language*) sebagai bahasa utama untuk mengelola dan memanipulasi data. MySQL dirancang sebagai perangkat lunak yang mendukung akses multi-user dan mampu menjalankan banyak proses secara bersamaan (*multithreaded*), sehingga memungkinkan pengelolaan data secara efisien dalam lingkungan yang kompleks dan dinamis[7]. Sistem Manajemen Basis Data dengan MySQL adalah solusi yang populer untuk mengelola dan menyimpan data. MySQL adalah sistem manajemen basis data open-source yang sangat fleksibel dan aman[8].

E. Hyper Text Markup Language(HTML)

HTML (*HyperText Markup Language*) merupakan bahasa dasar dalam web scripting yang bersifat *client-side*, digunakan untuk menampilkan informasi dalam bentuk teks, grafik, maupun multimedia di halaman web. Selain itu, HTML juga berfungsi untuk menghubungkan antar halaman web melalui *hyperlink*. Meskipun bukan merupakan protokol, HTML bekerja bersama dengan protokol HTTP untuk mentransfer data atau dokumen dari web server ke browser. Dengan adanya HTML, pengguna dapat menjelajahi berbagai halaman di internet dan menampilkan konten web secara visual di browser[9].

F. Bootstrap

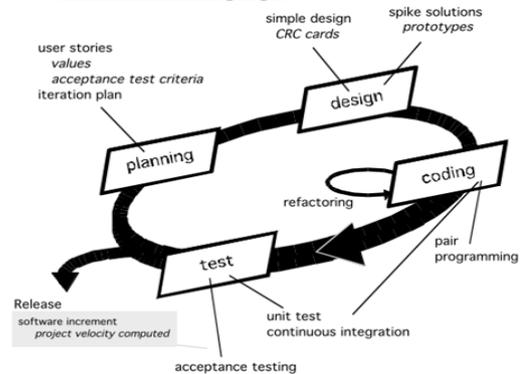
Bootstrap adalah platform CSS (*Cascading Style Sheet*) yang digunakan untuk mendesain situs web. Bootstrap adalah alat hebat yang dapat digunakan pemrogram saat membangun situs web[10].

G. Unified Modelling Language(UML)

Unified Modelling Language (UML) merupakan sebuah "bahasa" yang telah diakui sebagai standar industri untuk memvisualisasikan, merancang, serta mendokumentasikan sistem perangkat lunak. UML menyediakan standar dalam pembuatan model suatu sistem. Dengan UML, kita dapat membuat model untuk berbagai jenis aplikasi perangkat lunak, baik yang berjalan di berbagai perangkat keras, sistem operasi, maupun jaringan, serta ditulis dalam bahasa pemrograman apapun. Salah satu komponen dari Unified Modelling Language (UML) adalah Use Case dan Activity Diagram[11].

H. Extreme Programming (XP)

Extreme Programming (XP) adalah salah satu metodologi rekayasa perangkat lunak yang populer digunakan oleh para pengembang untuk mengembangkan aplikasi. Meskipun XP terlihat seperti kumpulan ide sederhana dan dianggap tidak memberikan dampak besar pada pengembangan aplikasi, metode ini memiliki keunggulan utama yaitu sangat cocok untuk proyek dengan kebutuhan yang dinamis atau persyaratan yang belum jelas dari klien. XP menekankan pada pengkodean yang sederhana, proses yang transparan, umpan balik konstan, dan kerja tim yang erat, sehingga memungkinkan pengembangan perangkat lunak yang cepat, efisien, dan minim risiko[12].



GAMBAR 1
(PROSES METODE EXTREME PROGRAMMING (XP))

I. Postman

Postman adalah alat perangkat lunak yang kuat yang digunakan oleh para pengembang untuk menguji dan memvalidasi kinerja serta stabilitas API. Alat ini berfungsi dengan mengirimkan permintaan ke API yang diarahkan ke server situs web, dan kemudian menunggu respons dari server[13].

J. Vite

Vite adalah alat bantu moderen yang digunakan untuk merancang pengembangan aplikasi web dengan lebih cepat[14]. Karena kecepatannya dalam waktu startup dan kemudahan konfigurasi, Vite menjadi pilihan populer di

kalangan pengembang *frontend*, terutama untuk proyek yang menggunakan *framework* seperti *Vue*, *React*, dan *Svelte*.

K. Express.JS

Express.js adalah sebuah *framework web* yang dirancang untuk memudahkan pengembangan aplikasi *backend* berbasis node.js[15]. Kesederhanaan dan fleksibilitasnya menjadi salah satu *framework backend* yang paling sering digunakan untuk pengembangan aplikasi yang ditulis dalam bahasa pemrograman *JavaScript*.

L. NodeJS

NodeJS adalah *platform runtime* yang didesain untuk mengembangkan aplikasi berbasis *website* yang ditulis dengan bahasa pemrograman *javascript*[16].

M. Midtrans Sandbox

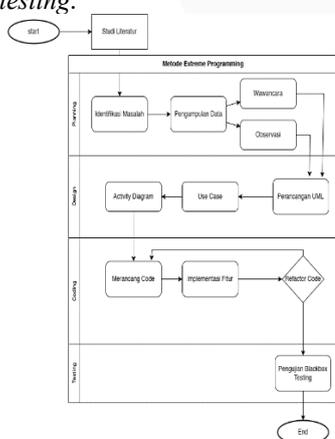
Midtrans Sandbox adalah lingkungan uji coba yang disediakan oleh Midtrans, sebuah layanan *payment gateway* di Indonesia, yang memungkinkan pengembang untuk menguji dan mensimulasikan proses transaksi pembayaran secara nyata tanpa menggunakan uang asli. Fungsi utama dari midtrans sandbox untuk Memungkinkan pengembang *web/aplikasi* untuk mensimulasikan transaksi pembayaran, Membantu pengembang untuk melakukan pengujian dan integrasi dengan sistem dan Mengizinkan pengembang untuk memvalidasi data transaksi yang dikirimkan ke midtrans dan respon yang diterima midtrans sesuai yang diharapkan oleh pengembang dan juga pengguna[17].

III. METODE

Proses penelitian mengenai rancang bangun sistem penggalangan dana berbasis *website* menggunakan metode *extreme programming*.

A. Diagram Alir Penelitian

Pada bagian ini menjelaskan tahapan yang dilakukan pada penelitian ini. Tahapan penelitian ini di susun secara sistematis untuk memudahkan peneliti dalam mencapai tujuan penelitian. Penelitian ini dimulai dari *planning*, *design*, *coding*, *testing*.



GAMBAR 2
(DIAGRAM ALIR PENELITIAN INI)

1. Planning

Pada tahap ini, akan dilakukan indentifikasi masalah serta pengumpulan data yang akan menjadi dasar dalam

pengembangan *website* panti asuhan Masjid Gelora Indah. Yang dimana melakukan identifikasi masalah dan mengumpulkan data dengan melakukan observasi dan wawancara.

2. Design

Setelah tahap perencanaan akan dilanjutkan dengan tahap design untuk memvisualisasikan sistem penggalangan dana panti asuhan Masjid Gelora Indah sebelum masuk tahap *coding*. Desain mencakup perancangan *use case diagram*, *activity diagram*, *class diagram*, *sequence diagram* dan *wireframe*.

3. Coding

Setelah tahap desain selesai, proses dilanjutkan dengan pengkodean, di mana setiap kode dirancang dan diimplementasikan sesuai fitur yang telah direncanakan. Jika terdapat perubahan, kode akan diperbaiki dan ditingkatkan kualitasnya agar memenuhi standar yang ditetapkan. Tahapan *coding* ini mencakup pengembangan *frontend*, *backend*, implementasi fitur, *continuous integration*, dan *refactor code*.

4. Testing

Setelah tahap *coding*, proses dilanjutkan dengan tahap testing untuk memastikan *website* berfungsi dengan baik dan sesuai kebutuhan. Pengujian dilakukan menggunakan Postman untuk menguji API serta metode *blackbox testing* yang berfokus pada pengujian fungsional sistem.

IV. HASIL DAN PEMBAHASAN

Penelitian ini bertujuan untuk membangun sebuah sistem penggalangan dana untuk Masjid Gelora Indah, menggunakan metode *extreme Programming*. Dalam penelitian ini terjadi 2 kali iterasi yaitu sebagai berikut:

A. Iterasi 1

Pada iterasi pertama, pengembangan difokuskan pada fitur-fitur utama yang menjadi dasar sistem penggalangan dana, seperti *login*, registrasi, lupa sandi, donasi, *payment gateway*, laporan donasi, laporan pengguna, dan *logout*. Pemilihan fitur ini didasarkan pada prioritas kebutuhan pengguna dan perannya dalam alur utama sistem.

1. Planning

Pada tahap perencanaan iterasi pertama, dilakukan identifikasi kebutuhan utama agar sistem dapat berfungsi dengan baik bagi pengguna. Kebutuhan ini diperoleh melalui observasi dan wawancara dengan pihak panti asuhan Masjid Gelora Indah, yang mencakup fitur serta proses inti yang harus tersedia dalam sistem. Adapun kebutuhan fungsionalnya adalah sebagai berikut:

TABEL 1
(TABEL KEBUTUHAN)

No	Nama Kebutuhan	Deskripsi
1	<i>Login</i>	Pengguna dan admin dapat masuk ke sistem
2	Register	Pengguna dan admin dapat mendaftar ke sistem

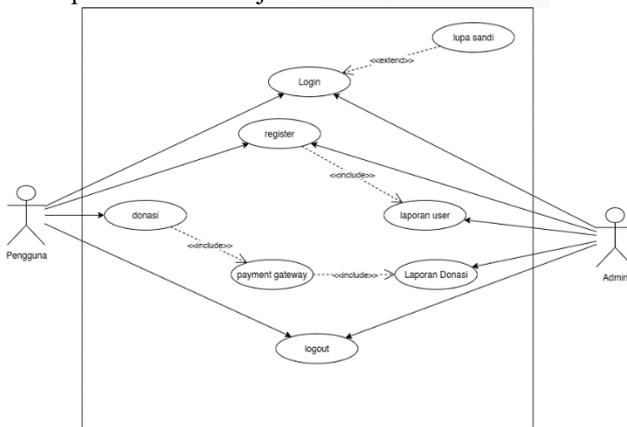
No	Nama Kebutuhan	Deskripsi
3	Lupa Sandi	Sistem dapat merubah kata sandi pengguna dan admin
4	Donasi	Dapat melakukan donasi secara online dengan berbagai metode pembayaran.
5	Laporan Donasi	Admin dapat melihat laporan donasi pengguna
6	Payment Gateway	Sistem dapat memproses pembayaran secara otomatis dengan midtrans sanbox.
.	Laporan user	Admin dapat melihat daftar seluruh anggota
8	logout	Pengguna dan admin dapat keluar dari akun

2. Design

Tahap desain pada iterasi pertama difokuskan pada kebutuhan fungsional dasar sebagai fondasi utama sistem donasi panti asuhan. Perancangan ini bertujuan untuk memvisualisasikan sistem sebelum memasuki tahap pengkodean, melalui pembuatan *use case diagram*, *activity diagram*, *class diagram*, *sequence diagram*, dan *wireframe* sesuai kebutuhan fungsional.

a) Use Case

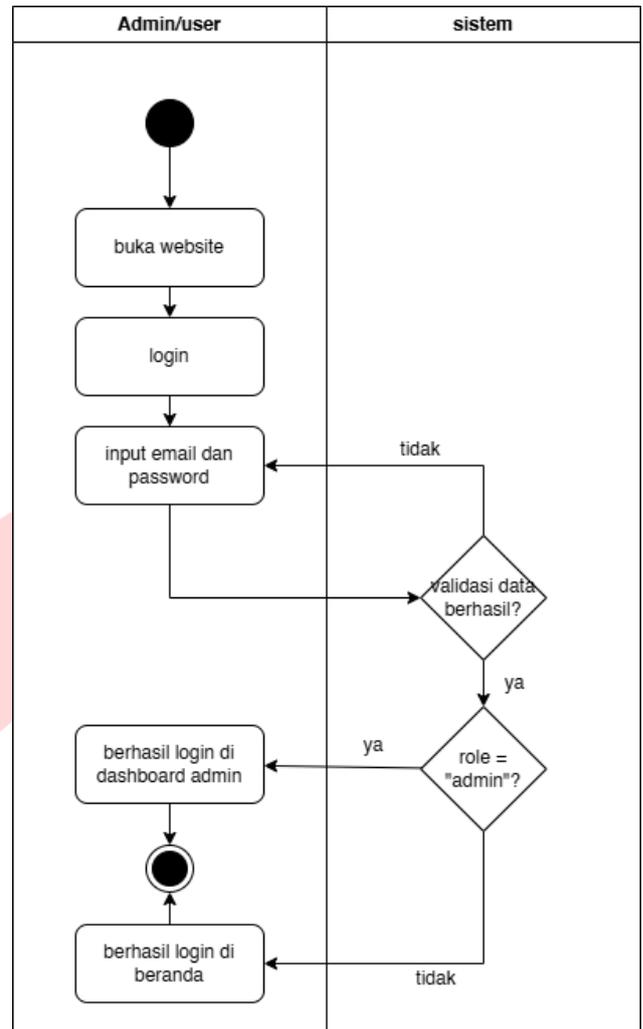
Diagram Use Case disusun sebagai langkah awal untuk merancang bagaimana pengguna berinteraksi dengan sistem donasi panti asuhan Masjid Gelora Indah.



GAMBAR 3
(USE CASE DIAGRAM ITERASI 1)

b) Activity Diagram

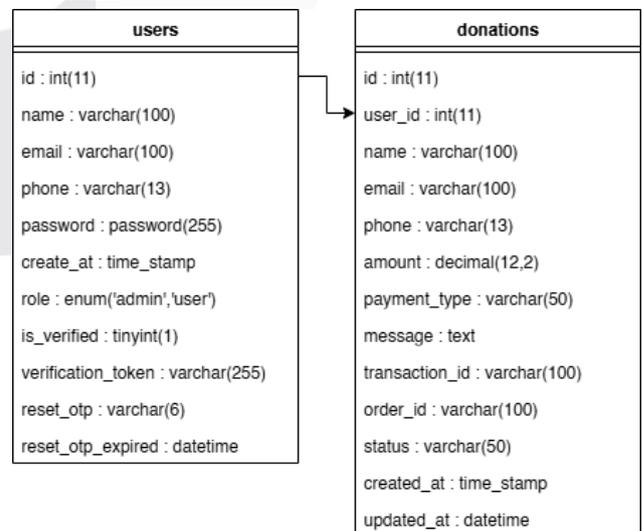
Setelah perancangan use case, tahap selanjutnya adalah membuat *activity diagram* untuk memvisualisasikan alur aktivitas dalam sistem. Berikut adalah *activity diagram* yang disusun:



GAMBAR 4
(ACTIVITY DIAGRAM)

c) Class Diagram

Class diagram digunakan untuk menggambarkan struktur data dasar pada basis data, yang mencakup tabel *Users* dan *Donations*.



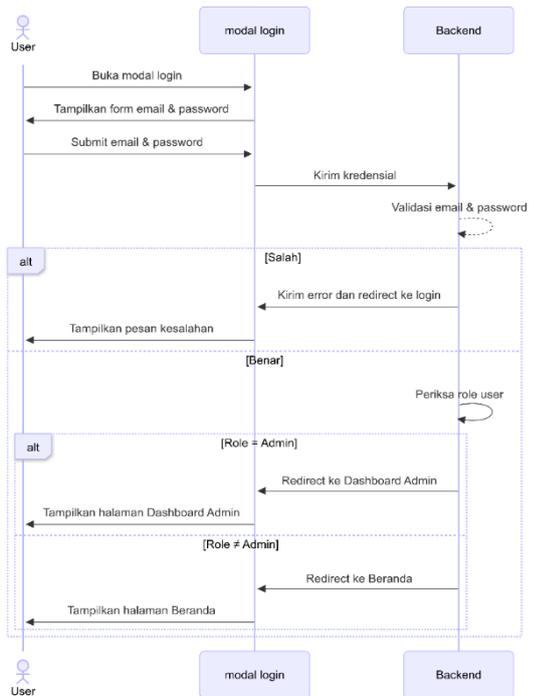
GAMBAR 5
(CLASS DIAGRAM)

Class Users menyimpan data lengkap pengguna, seperti ID unik, informasi kontak (nama, email, nomor telepon), kata

sandi, peran, status verifikasi akun (*is_verified*, *verification_token*), serta data untuk *reset* kata sandi (*reset_otp*, *reset_otp_expired*), dan tanggal pembuatan akun (*created_at*). Sementara itu, *class Donations* mencatat transaksi donasi, mencakup ID donasi, nominal, metode pembayaran, pesan, ID transaksi eksternal, status, serta waktu pembuatan dan pembaruan data (*created_at*, *updated_at*). *Class Donations* terhubung ke *Users* melalui *user_id*, yang menandakan bahwa setiap donasi dilakukan oleh pengguna tertentu.

d) *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan interaksi antara aktor dalam suatu skenario, dengan menampilkan alur komunikasi antara pengguna, sistem, dan komponen lain secara berurutan sesuai waktu.



GAMBAR 6 (SEQUENCE DIAGRAM ITERASI 1)

e) *Wireframe*

Wireframe low-fidelity digunakan pada tahap awal perancangan antarmuka untuk menggambarkan secara kasar tata letak elemen pada *website* donasi Panti Asuhan Masjid Gelora Indah. Perancangan ini dibuat sederhana tanpa detail visual seperti warna, gambar, atau tipografi, sehingga fokus pada struktur, fungsi, dan alur navigasi. Tujuannya adalah memberikan gambaran awal sebelum tahap *coding*. Pada iterasi pertama, *wireframe* menampilkan halaman beranda sebagai tampilan awal, halaman donasi dengan integrasi Midtrans sandbox, serta fitur *login*, registrasi, dan lupa sandi. Admin juga dapat mengakses laporan pengguna dan laporan donasi.



GAMBAR 7 (WIREFRAME ITERASI 1)

3. *Coding*

Pada iterasi pertama, pengkodean dilakukan dengan menulis *code* berdasarkan design yang telah dibuat sebelumnya.

a) Pengembangan *Frontend*

Pengembangan *frontend* menggunakan *React.js* sebagai *framework* utama untuk membangun antarmuka pengguna. Bahasa pemrograman yang digunakan adalah *JavaScript*, dengan bantuan CSS dan *framework* *Bootstrap* untuk memastikan desain yang responsif dan konsisten di berbagai perangkat. Diantaranya ada fitur *login*, fitur *register*, fitur lupa sandi, fitur donasi, *payment gateway*, laporan donasi, laporan *user* dan *logout*.

```

const DonasiPage = () => {
  return (
    <div className="donasiPage">
      <div className="donasi min-vh-100">
        <Container>
          <Row className="image">
            <div className="image-donasi text-center mb-4">
              <img src={ImageDetail} alt="Donasi" className="donasi-image" />
            </div>
          </Row>
          <Row className="nominal">
            <div className="text-center mb-4">
              <h1 className="donasi-title">Donasi</h1>
            </div>
            <Col>
              <div className="donasi-form-container">
                <div className="mb-4">
                  <h4 className="form-title">
                    <FaHeart className="text-danger me-2" />
                    Masukkan Nominal Donasi
                  </h4>
                </div>
              </div>
            </Col>
          </Row>
        </Container>
      </div>
    </div>
  )
}

```

GAMBAR 8 (CODINGAN FRONTEND)

b) Pengembangan *Backend*

Pengembangan *backend* difokuskan pada logika bisnis utama dan penyediaan layanan API untuk mendukung komunikasi antara *frontend* dan *server*. *Backend* dibangun menggunakan *Node.js* dan *Express.js* sebagai *framework* utama, yang memungkinkan pembuatan RESTful API secara modular, efisien, dan mudah dikelola. Berikut adalah struktur sistem *backend* untuk panti asuhan Masjid Gelora Indah.

Backend sistem terintegrasi dengan database *MySQL* menggunakan ORM *Sequelize*, yang memudahkan pengelolaan tabel seperti *users* dan *donations*. Relasi antar entitas dirancang secara optimal untuk mendukung efisiensi

query dan skalabilitas sistem. Struktur database yang digunakan ditampilkan pada gambar berikut:

db_pantiasuhan users	db_pantiasuhan donations
id : int(11)	id : int(11)
name : varchar(100)	user_id : int(11)
email : varchar(100)	name : varchar(100)
phone : varchar(20)	email : varchar(100)
password : varchar(255)	phone : varchar(20)
created_at : timestamp	amount : decimal(12,2)
role : enum('user','admin')	message : text
is_verified : tinyint(1)	payment_type : varchar(50)
verification_token : varchar(255)	transaction_id : varchar(100)
reset_otp : varchar(6)	order_id : varchar(100)
reset_otp_expiry : datetime	status : varchar(50)
	created_at : timestamp
	updated_at : timestamp

GAMBAR 9 (DATABASE ITERASI 1)

Fitur utama *backend* mencakup *endpoint* untuk proses registrasi, *login*, lupa sandi, dan pengelolaan akun pengguna. Proses autentikasi menggunakan *JSON Web Token (JWT)* untuk menjaga keamanan sesi pengguna, dengan validasi input dan hashing password yang dilakukan menggunakan *bcrypt.js*. *Backend* juga menyediakan *endpoint* khusus untuk meneruskan permintaan pengguna ke *Midtrans Snap* sebagai sistem pembayaran. Untuk terhubung dengan *Midtrans*, diperlukan konfigurasi *client key* dan *server key* yang disimpan secara aman dalam file environment. Setelah proses donasi selesai—baik berhasil, gagal, maupun kedaluwarsa—*Midtrans* akan mengirim notifikasi ke *backend* melalui *webhook* untuk memperbarui status transaksi.

c) Implementasi Fitur

Berbagai tampilan antarmuka pengguna serta hasil implementasi dari fitur-fitur utama dalam sistem penggalangan dana berbasis *web*. Setiap tampilan mencerminkan fungsionalitas sistem yang telah dirancang untuk memenuhi kebutuhan *user* maupun *admin*. Berikut tampilan antarmuka dari iterasi pertama:

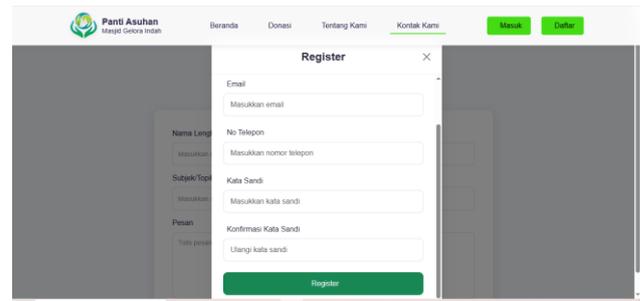
1. Halaman Beranda



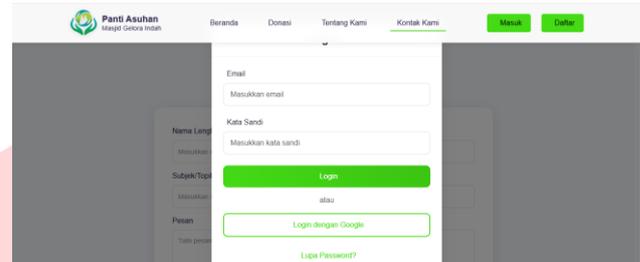
GAMBAR 10 (GAMBAR HALAMAN BERANDA)

Tampilan awal ketika *user* mengakses *website*. Berisi *informasi* umum tentang panti asuhan, ajakan untuk berdonasi, dan tombol navigasi ke fitur lainnya. Halaman ini dirancang menarik dan *informatif* untuk menarik perhatian calon donatur.

2. Modal Register, Modal Login, dan Modal Lupa Sandi



GAMBAR 11 (MODAL LOGIN)



GAMBAR 12

Menampilkan *form* untuk mendaftar dan masuk ke dalam sistem serta mengganti kata sandi. *User* dapat membuat akun menggunakan email atau melalui *login* Google. Validasi dilakukan untuk memastikan data yang dimasukkan valid dan aman.

3. Halaman Donasi

User dapat mengisi *form* donasi, yang dimana memasukkan nominal donasi. selanjutnya *user* akan menginputkan nama, email, nomor telephone. ketika *user* sudah *login* *user* tidak perlu menginputkan nama, email, nomor telephone. Dan *user* bisa klik *checkbox* untuk mengganti nama mereka menjadi hamba Allah. Pengguna juga harus mengisi pesan. Setelah *user* klik button lanjutkan pembayaran akan keredirect ke *payment gateway* yaitu *midtrans snap*.

4. Payment Gateway

Setelah *user* berhasil ke *redirect* ke *midtrans snap* *user* akan memilih metode pembayaran yang dimana *user* akan melanjutkan pembayaran dengan metode pembayaran yang dipilih.

5. Halaman Data Donasi

Halaman Data Donasi menampilkan seluruh *informasi* terkait aktivitas donasi yang dilakukan oleh pengguna. Pada bagian atas halaman, terdapat ringkasan berupa total donasi berhasil, jumlah donasi yang masuk minggu ini, dan donasi yang diterima pada hari ini, yang ditampilkan secara *real-time* untuk memudahkan pemantauan. Selain itu, halaman ini juga dilengkapi dengan fitur pencarian donasi berdasarkan nama atau *informasi* terkait lainnya, serta *filter* berdasarkan status donasi seperti sukses, pending, atau gagal. Dengan tampilan yang rapi dan interaktif, halaman ini membantu *admin* dalam mengelola serta mengevaluasi data donasi secara lebih efisien dan terstruktur.

6. Halaman User

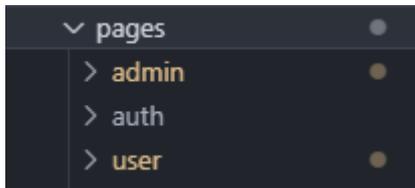
Halaman Kelola *User* berfungsi sebagai tempat bagi admin untuk mengelola data pengguna dalam sistem. Fitur-fitur ini dirancang agar proses manajemen akun dalam sistem penggalangan dana dapat dilakukan dengan cepat, dan terorganisir.

d) *Continuous Integration*

Dalam proyek ini, *Continuous Integration* (CI) diterapkan menggunakan GitHub sebagai sistem *version control*. Setiap perubahan kode yang di-commit atau di-push akan diuji dan digabungkan secara bertahap ke repositori utama. Pengembangan dilakukan di cabang *master*, lalu dipindahkan ke cabang *main* setelah fungsionalitas dianggap sesuai.

e) *Refactor Code*

Refactoring dilakukan setelah semua fitur sistem berhasil diimplementasikan dan berjalan sesuai kebutuhan. *Refactor* ini bertujuan untuk meningkatkan kualitas kode tanpa mengubah fungsionalitasnya. Yang dimana pada iterasi pertama *refactor* pada struktur folder pages pada *frontend* yang dimana sebelumnya halaman admin dan pengguna biasa digabung.



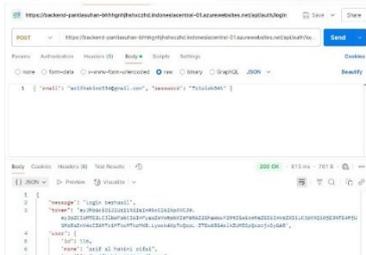
GAMBAR 13 (PEMBAGIAN PAGE)

4. *Testing*

Pengujian dilakukan untuk memastikan bahwa seluruh fungsi dalam sistem berjalan sesuai dengan yang diharapkan. Metode yang digunakan dalam tahap ini adalah *Blackbox Testing* dan postman.

a. Postman

Pengujian *backend* dilakukan secara manual menggunakan Postman untuk mengirim permintaan HTTP dan memverifikasi *respons* API, guna memastikan sistem berjalan sesuai logika yang telah dirancang.



GAMBAR 14 (GAMBAR PENGUJIAN API POSTMAN)

b. *Blackbox Testing*

Pengujian dilakukan untuk memastikan semua fungsi sistem berjalan sesuai harapan. Tahap ini menggunakan metode *Blackbox Testing*, yaitu pengujian yang berfokus pada fungsi sistem tanpa melihat struktur internal program. Prosesnya dilakukan dengan memberikan input dan mengamati output untuk memastikan sistem berfungsi dengan benar.

TABEL 2 (TABEL PENGUJIAN KEBUTUHAN FUNGSIONAL)

No	Nama Fitur	Input	Output	Hasil
1	<i>Login</i>	<i>Email dan password</i>	Pengguna berhasil <i>login</i>	Berhasil
	<i>Register</i>	Input <i>register</i>	Pengguna berhasil terdaftar ke sistem	Berhasil
	Lupa Sandi	Input kata sandi baru	Berhasil mengganti kata sandi	Berhasil
	Donasi	Input <i>form donasi</i>	Ke redirect ke <i>payment gateway</i>	Berhasil
	Laporan Donasi	Masuk ke halaman laporan donasi	Menampilkan data donasi	Berhasil
	<i>Payment Gateway</i>	Melakukan transaksi donasi	Donasi berhasil	Berhasil
	Laporan <i>User</i>	Masuk ke halaman laporan <i>user</i>	Menampilkan data laporan <i>user</i>	Berhasil
	<i>Logout</i>	Menekan <i>button logout</i>	Keluar dari akun yang <i>login</i>	Berhasil

B. Iterasi 2

Setelah penyelesaian fitur utama di iterasi pertama, iterasi kedua difokuskan pada pengembangan fitur pendukung yang bertujuan memperkuat fungsi administratif dan pelaporan dalam sistem." Silakan beri tahu jika ingin versi yang lebih *formal* atau teknis.

1. Planning

Perencanaan pada iterasi kedua dilakukan perubahan yang dimana menambahkan beberap fitur pendukung.

TABEL 3 (KEBUTUHAN FUNGSIONAL ITERASI 2)

No	Nama Kebutuhan	Deskripsi
1	Riwayat	Pengguna dapat melihat data donasi yang mereka lakukan.
	Hubungi Kami	<i>user</i> dapat mengirim pesan atau pertanyaan.
	Laporan Pesan	Admin dapat melihat laporan atau daftar pesan.
	Pencarian Pesan	Admin dapat mencari pesan masuk berdasarkan nama pengirim.
	Pencarian <i>User</i>	Admin dapat mencari data pengguna berdasarkan nama atau email.
	Pencarian Donasi	Admin dapat mencari donasi berdasarkan nama.

3. Coding

Pengkodean pada iterasi kedua difokuskan pada pengembangan fitur pendukung berdasarkan desain yang sudah dibuat di tahap sebelumnya.

a) Pengembangan *Frontend*

Pada iterasi kedua dilakukan perubahan dalam bentuk penambahan yang dimana menambahkan fitur riwayat donasi, hubungi kami, laporan pesan, pencarian pesan, pencarian *user*, pencarian donasi, *filter* donasi berdasarkan status, download donasi.

```
const KelolaPesan = () => {
  return (
    <div className="kelola-pesan-container">
      <Container fluid className="px-4 py-3">
        <Card className="shadow-sm border-0">
          <Card.Body>
            <div className="d-flex justify-content-between align-items-center mb-4">
              <div className="d-flex align-items-center">
                <Fasvelope className="text-primary me-2" size={24} />
                <h2 className="mb-0">Kelola Pesan</h2>
              </div>
            </div>
            <div className="search-container" style={{ width: "400px" }}>
              <InputGroup>
                <InputGroup.Text className="bg-light border-end-0">
                  <FasSearch className="text-muted" />
                </InputGroup.Text>
                <Form.Control
                  placeholder="Cari pesan berdasarkan nama, email, atau isi pesan..."
                  value={searchTerm}
                  onChange={(e) => setSearchTerm(e.target.value)}
                  className="border-start-0"
                />
                <Button
                  searchTerm && {
                    </Button

```

GAMBAR 20

(PROSES PENGEMBANGAN FRONTEND ITERASI 2)

b) Pengembangan *Backend*

Pada iterasi kedua dilakukan perubahan dalam bentuk penambahan yang dimana menambahkan tabel di database yaitu tabel pertanyaan yang dimana akan menampung pesan dari pengguna.

Pada iterasi kedua, ditambahkan tabel pertanyaan ke dalam struktur database. Relasi antar entitas diatur secara optimal untuk mendukung efisiensi kueri dan skalabilitas sistem. Berikut adalah tabel-tabel yang telah dibuat:

db_pantiasuhan.users	db_pantiasuhan.donations	db_pantiasuhan.pertanyaan
<ul style="list-style-type: none"> id: int(11) name: varchar(100) email: varchar(100) phone: varchar(20) password: varchar(255) created_at: timestamp role: enum('user', 'admin') is_verified: tinyint(1) verification_token: varchar(255) reset_otp: varchar(6) reset_otp_expiry: datetime 	<ul style="list-style-type: none"> id: int(11) user_id: int(11) name: varchar(100) email: varchar(100) phone: varchar(20) amount: decimal(12,2) message: text payment_type: varchar(50) transaction_id: varchar(100) order_id: varchar(100) status: varchar(50) created_at: timestamp updated_at: timestamp 	<ul style="list-style-type: none"> id: int(11) name: varchar(100) email: varchar(100) subjek: varchar(150) message: text created_at: timestamp

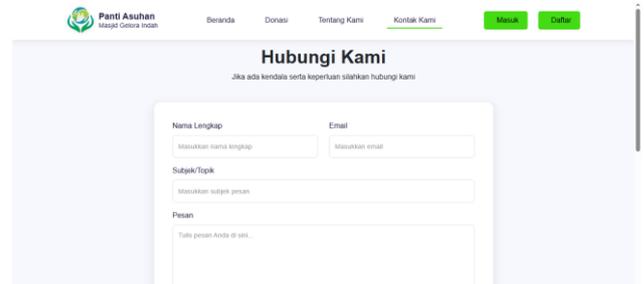
GAMBAR 21

(BACKEND ITERASI 2)

Backend menyediakan endpoint khusus untuk menangani pengiriman pesan dari pengguna melalui fitur Hubungi Kami. Pengiriman dilakukan dengan metode POST, di mana data seperti nama, email, dan isi pesan dikirim dari *frontend* ke *server* dan disimpan dalam database. Selain itu, tersedia endpoint GET yang memungkinkan admin mengambil dan menampilkan seluruh pesan masuk. Fitur ini mendukung manajemen pesan oleh admin, termasuk pencarian berdasarkan nama pengirim atau isi pesan, serta menjadi dasar pembuatan laporan pesan.

c) Implementasi Fitur

Pada iterasi kedua, ditampilkan berbagai antarmuka pengguna beserta hasil implementasi fitur tambahan dalam sistem donasi berbasis *web*. Setiap tampilan mencerminkan fungsionalitas yang dirancang untuk memenuhi kebutuhan pengguna dan admin.



GAMBAR 22

(HALAMAN HUBUNGI KAMI)

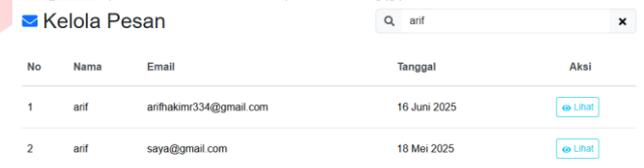
Berisi *form* yang dapat diisi oleh *user* untuk mengirimkan pertanyaan, kritik, atau saran. Pesan yang dikirim akan disimpan dan ditampilkan di dashboard admin.



GAMBAR 23

(RIWAYAT ITERASI 2)

Menampilkan daftar donasi yang telah dilakukan oleh *user*. Setiap riwayat mencantumkan jumlah donasi, metode pembayaran, status transaksi, dan tanggal.



GAMBAR 24

(PENCARIAN PESAN ITERASI 2)

Pencarian pesan digunakan oleh admin untuk mencari pesan yang dikirim oleh pengguna. dengan menekan *form* inputan admin dapat mencari berdasarkan nama dan email pengguna.



GAMBAR 25

(PENCARIAN DONASI ITERASI 2)

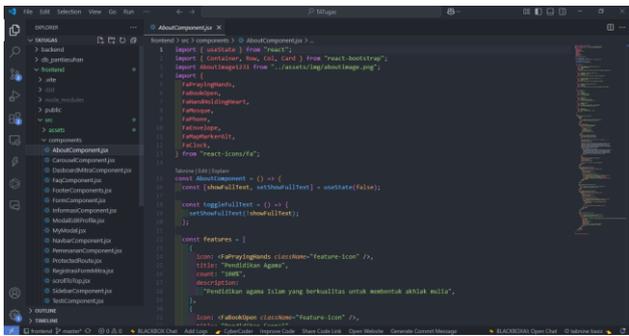
Pencarian donasi digunakan oleh admin untuk mencari donasi yang dikirim oleh pengguna. dengan menekan *form* inputan admin dapat mencari berdasarkan nama dan email pengguna

d) Continuous Integration

Pada iterasi kedua, praktik Continuous Integration (CI) tetap diterapkan untuk memastikan bahwa setiap fitur baru yang dikembangkan tidak mengganggu kestabilan sistem yang telah dibangun pada iterasi sebelumnya. Setiap perubahan atau penambahan fitur, seperti pencarian data, *filter* donasi, dan unduhan laporan Excel, langsung di-commit dan push ke repositori GitHub secara bertahap.

e) Refactor Code

Pada iterasi kedua, proses refactoring kembali dilakukan untuk meningkatkan struktur dan keterbacaan kode seiring bertambahnya kompleksitas fitur yang dikembangkan. Khususnya, bagian komponen *frontend* dengan menerapkan prinsip reusability, yaitu mengelompokkan bagian-bagian UI yang sering digunakan menjadi komponen:



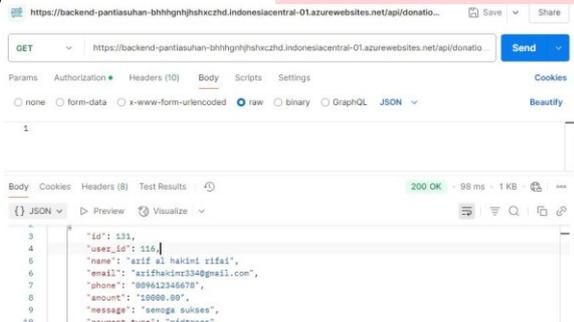
GAMBAR 26
(REUSABLITY ABOUT COMPONENT ITERASI 2)

4. Testing

Pada iterasi ke dua, terdapat perubahan yang dimana pengujian dilakukan dengan menambahkan fitur tambahan untuk memastikan bahwa seluruh fungsi dalam sistem berjalan sesuai dengan yang diharapkan.

a) Postman

Pada iterasi kedua pengujian API masih menggunakan postman sebagai alat uji untuk memastikan sistem merespon dengan benar.



GAMBAR 27
(PENGUJIAN POSTMAN ITERASI 2)

Pengujian API riwayat donasi bertujuan memastikan bahwa sistem dapat menampilkan data donasi milik masing-masing pengguna secara individual. Pengujian dilakukan menggunakan Postman dengan metode HTTP GET melalui endpoint: `http://localhost:5000/api/donations/history`.

Endpoint ini hanya bisa diakses oleh pengguna yang telah login dan memiliki token otorisasi. Oleh karena itu, pengujian login dilakukan terlebih dahulu untuk mendapatkan *access token*, yang kemudian dikirim melalui tab *Authorization* (tipe Bearer Token) atau melalui tab *Headers* dengan *key Authorization* dan value *Bearer <token>*. Jika token valid, sistem merespons dengan status 200 OK dan menampilkan data donasi pengguna, termasuk jumlah donasi, tanggal transaksi, status pembayaran, dan nomor order.

b) Blackbox Testing

Pengujian pada iterasi ke dua dilakukan pada fitur-fitur tambahan baik dari sisi *user* maupun *admin*. Berikut adalah tabel hasil pengujian:

TABEL 4
(PENGUJIAN BLACKBOX TESTING ITERASI KE DUA)

No	Nama Fitur	Input	Output	Hasil
1	Riwayat	Masuk ke halaman riwayat	Menampilkan data riwayat	Berhasil
2	Hubungi Kami	Mengirim pesan	Data berhasil terkirim	Berhasil
3	Laporan Pesan	Masuk ke halaman laporan pesan	Menampilkan data	Berhasil

No	Nama Fitur	Input	Output	Hasil
			laporan pesan	
4	Pencarian Pesan	Memasukkan keyword nama dan email	Menampilkan data yang dicari	Berhasil
5	Pencarian Donasi	Memasukkan keyword nama, email, id_donations	Menampilkan data yang dicari	Berhasil
6	Filter Donasi	Pilih dropdown status yang ingin di cari	Berhasil menampilkan status berdasarkan pencarian status	Berhasil
7	Pencarian User	Memasukkan keyword nama, email dan no telephone	Menampilkan data yang dicari	Berhasil
8	Download Donasi	Klik button download donasi	Data donasi berhasil ke unduh dengan format excel	Berhasil

V. KESIMPULAN

Penelitian ini bertujuan untuk mempermudah partisipasi donatur terhadap Panti Asuhan Masjid Gelora Indah sekaligus membantu panti dalam menyebarkan informasi serta memberikan akses yang lebih mudah bagi donatur. Sistem yang dibangun berhasil menyediakan platform digital donasi online dengan berbagai metode pembayaran, seperti Virtual Account, QRIS, dan VISA.

Melalui pengujian menggunakan metode *Blackbox Testing*, seluruh fitur utama seperti registrasi, login, donasi, pengelolaan dan riwayat donasi, validasi data, pencarian donasi berdasarkan ID, nama, dan status, pencarian pengguna, pencarian pesan, serta pengiriman pesan terbukti berjalan sesuai fungsinya. Hal ini menunjukkan bahwa sistem mampu memenuhi kebutuhan pengguna dan admin. Dengan tambahan fitur riwayat transaksi dan grafik laporan donasi, sistem semakin mendukung transparansi dan manajemen data.

Dengan demikian, tujuan tugas akhir ini berhasil dicapai, yaitu membangun sistem penggalangan dana yang mempermudah proses donasi bagi donatur dan membantu operasional Panti Asuhan Masjid Gelora Indah.

REFERENSI

- [1] P. Mauliana, R. Firmansyah, and A. Sutardi, "Perancangan Sistem Informasi Panti Asuhan 'e-PANTI' Berbasis *Web*," *Jurnal Sistem Informasi*, vol. 10, 2019.
- [2] Y. Sulthoni, "Strategi Pembentukan Karakter Anak Di Panti Asuhan Muhammadiyah Wiyung Surabaya," Vol. 1, No. 1, 2013.
- [3] I. G. N. Suryantara, *Merancang Aplikasi Dengan Metodologi Extreme Programming*. Jakarta: Elex Media Komputindo, 2017.
- [4] R. Sitanggang, T. U. Dachi, And I. H. G. Manurung, "Rancang Bangun Sistem Penjualan Tanaman Hias Berbasis *Web* Menggunakan Php Dan Mysql," Vol. 4, No. 1, 2022.
- [5] I. Rochmawati, "Iwearup.Com *User* Interface Analysis," *VSLT*, Vol. 7, No. 2, Pp. 31–44, Feb. 2019, Doi: 10.33375/Vslt.V7i2.1459.
- [6] C. Christian And A. Voutama, "Implementasi Aplikasi Antrian Pencucian Mobil Berbasis *Web* Menggunakan Php, *Javascript*, Html, Css Dan Uml: Studi Kasus Di Asy Cuci Steam Mobil," *Jati*, Vol. 8, No. 2, Pp. 2243–2248, Apr. 2024, Doi: 10.36040/Jati.V8i2.9460.
- [7] R. Fitri, *Pemrograman Basis Data Menggunakan Mysql*. Banjarmasin: Poliban Press, 2020.
- [8] R. F. Ramadhan And R. Mukhaiyar, "Penggunaan Database Mysql Dengan Interface Phpmyadmin Sebagai Pengontrolan Smarthome Berbasis Raspberry Pi," *JTEIN*, Vol. 1, No. 2, Pp. 129–134, Nov. 2020, Doi: 10.24036/Jtein.V1i2.55.
- [9] H. Thamrin, O. Fajarianto, And A. Ahmad, "Pelatihan Pemrograman Css Dan Html Di Smk Avicena," *Abdiawl*, Vol. 4, No. 1, Pp. 51–60, Jan. 2021, Doi: 10.56301/Awal.V4i1.125.
- [10] G. A. Supriatmaja, K. D. D. Widyaputra, J. Deva, And G. S. Mahendra, "Sistem Informasi Perpustakaan Menggunakan *Framework* Bootstrap Dengan PHP Native Dan Database Mysql Berbasis *Web* Pada SMP Negeri 2 Dawan," Vol. 1, No. 1, 2022.
- [11] P. Sulistyorini, "Pemodelan Visual Dengan Menggunakan UML Dan Rational Rose," 2009.
- [12] D. Kustiawan, W. N. Cholifah, R. Destriana, And N. Heriyani, "Rancang Bangun Sistem Informasi Akuntansi Pengelolaan Koperasi Menggunakan Metode *Extreme Programming*," *JATI*, Vol. 12, No. 1, Pp. 78–92, Apr. 2022, Doi: 10.34010/Jati.V12i1.6756.
- [13] M. D. Firmansyah And E. E. Wei, "Perancangan Dan Implementasi Sistem Manajemen Sekolah Di SMA Kartini Batam Menggunakan *Framework* Scrum," Vol. 4, 2022.
- [14] T. Alfa Rizzi, R. Aziz Abdillah, Y. Efani Yancandra, M. Wijaya, And A. Voutama, "Implementasi *React* Vite Dalam Pengembangan Antarmuka Sistem Pemesanan Tiket Pesawat Dengan Metode Scrum," *Processor*, Vol. 20, No. 1, May 2025, Doi: 10.33998/Processor.2025.20.1.2231.
- [15] M. R. Nahjan, A. A. Ridha, N. Heryana, And A. Voutama, "Rancang Bangun *Website* Pencarian Informasi Berita Dan Cuaca Daerah Di Indonesia Menggunakan Api Dan Express.js," Vol. 7, No. 5, 2023.
- [16] Hasanuddin, H. Asgar, And B. Hartono, "Rancang Bangun Rest Api Aplikasi Weshare Sebagai Upaya Mempermudah Pelayanan Donasi Kemanusiaan," *Jinteks*, vol. 4, no. 1, pp. 8–14, Feb. 2022, doi: 10.51401/jinteks.v4i1.1474.
- [17] W. Firdaus and A. Prihanto, "Integrasi *Website* Danzstore Dengan Midtrans Menggunakan Metode *Websnap* Untuk Menangani Transaksi Pembayaran," vol. 06, 2024.