

Rancang Bangun Sistem Informasi di Panti Asuhan Artanita Al-Khairiyah Menggunakan Metode *Extreme Programming*

1st Rizal Adiyaksa Abdul Malik

Fakultas Informatika

Universitas Telkom

Purwokerto, Indonesia

rizaladiyaksaabdulma@student.telkomuniversity.ac.id

2nd Ipam Fuaddina Adam

Fakultas Informatika

Universitas Telkom

Purwokerto, Indonesia

ipamya@telkomuniversity.ac.id

Abstrak — Panti Asuhan Artanita Al-Khairiyah merupakan lembaga yang menampung anak-anak dengan masalah sosial untuk mendapatkan perawatan, bimbingan, dan pendidikan yang diberikan dengan berbagai pengetahuan dan keterampilan. Dalam pelaksanaannya sering kali mengalami keterbatasan finansial mengakibatkan keterbatasan dalam operasionalnya, maka dari itu, membutuhkan dukungan dari para donatur yang bersedia membantu mereka, tetapi dalam penyaluran donasi masih metode manual dengan menggunakan buku tulis, menyebabkan efisiensi kurang optimal dalam pengelolaan dana. Selain itu, kurangnya penyebaran informasi donasi untuk mendukung pembiayaan anak-anak asuh kepada calon donatur dan kurangnya informasi mengenai kegiatan anak-anak asuh serta ketidaktransparanan dalam penyaluran hasil donasi juga menjadi masalah yang perlu diselesaikan. Penelitian ini bertujuan merancang sistem informasi donasi panti asuhan Artanita Al-Khairiyah. Dengan adanya website ini, informasi terkait panti asuhan dapat tersebar luas dan dapat mudah diakses oleh para donatur. Meningkatkan tingkat transparansi dalam penyaluran donasi dan program-program yang dilaksanakan sehingga dapat membangun kepercayaan dari para donatur.

Kata kunci— Panti Asuhan, anak-anak asuh, donasi, sistem informasi

I. PENDAHULUAN

Perkembangan teknologi informasi di era globalisasi berkembang sangat pesat mendorong masyarakat semakin mudah untuk memperoleh sebuah informasi. Penggunaan teknologi informasi pada lembaga atau organisasi yang berskala kecil, umumnya digunakan secara terbatas, seperti untuk tugas-tugas penyetoran dan untuk mendukung proses Administratif. Namun, dalam konteks lembaga sosial seperti panti asuhan, kemajuan dalam teknologi informasi yang lebih efektif dapat dimanfaatkan untuk pengembangan sistem informasi panti asuhan berbasis *web* [1]

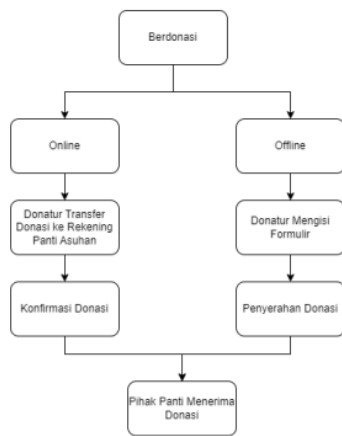
Panti Asuhan adalah lembaga sosial yang berperan memberikan pendidikan dan pembinaan kepada anak-anak

yang menghadapi masalah sosial, seperti kendala ekonomi dan kehilangan salah satu atau kedua orang tua [2]. Maka dari itu, panti asuhan membutuhkan dukungan dari para donatur yang bersedia membantu mereka. Namun, seringkali para donatur mengharapkan adanya transparansi mengenai penggunaan dana yang telah disalurkan kepada panti asuhan [3].

Panti asuhan perlu mengikuti Standar Nasional Pengasuhan Anak untuk Lembaga Kesejahteraan Sosial Anak mencakup Visi, Misi, Perijinan, Pendirian, Peran Dinas Sosial, Akreditasi Lembaga, dan Fasilitas [1]. Mengutip dari Peraturan Menteri Sosial 30 tahun 2011 pasal 6 ayat 1 point g yang tertulis ketersediaan dana, sistem pengelolaan, dan pertanggung jawaban, maka Panti Asuhan memerlukan peningkatan keterampilan manajemen pengelolaan Lembaga Layanan Sosial Anak (LKSA) [2][4]. Saat ini, sebagian besar panti asuhan telah memanfaatkan teknologi informasi berbasis *web* sebagai sarana untuk mengelola dan menyajikan informasi lembaga, khususnya dalam konteks pelayanan sosial dan kemanusiaan. Salah satu implementasinya adalah sistem pengelolaan informasi donasi, yang berperan penting dalam mendukung kebutuhan ekonomi anak-anak asuh. Selain itu, sistem ini juga berkontribusi dalam peningkatan untuk akreditasi panti asuhan, yang menjadi tolak ukur kelayakan dan standar lembaga kesejahteraan sosial. Penilaian akreditasi ini mencakup berbagai aspek, seperti program layanan, sumber daya manusia, manajemen dan struktur organisasi, sarana dan prasarana, serta hasil dari pelayanan kesejahteraan sosial yang diberikan [2].

Panti Asuhan Artanita Al-Khairiyah merupakan Lembaga Kesejahteraan Sosial Anak yang bertempat di Kota Tasikmalaya menampung anak-anak yang akan mendapatkan perawatan, bimbingan, dan pendidikan yang diberikan dengan berbagai pengetahuan dan keterampilan. Hal ini bertujuan untuk mengembalikan

rasa percaya diri mereka dengan memperluas wawasan dan mengasah kreativitas, sehingga mereka merasa yakin memiliki prospek masa depan yang cerah [2].



GAMBAR 1
Alur Penyaluran Donasi

Alur penyaluran donasi di Panti Asuhan Artanita Al-Khairiyah, sebagaimana ditunjukkan pada Gambar 1.1, saat ini masih berjalan melalui dua jalur, yakni secara *offline* dan *online*. Untuk metode *offline*, donatur diwajibkan mengisi formulir yang telah disediakan oleh pihak panti, yang mencakup informasi pribadi, jenis dan jumlah donasi, serta tujuan pemberian bantuan. Sementara itu, untuk donasi secara online, panti asuhan menyediakan nomor rekening resmi yang dapat digunakan oleh donatur untuk mentransfer dana dalam bentuk uang tunai. Mekanisme ini masih digunakan hingga saat ini sebagai sistem utama dalam pengelolaan dan penyaluran donasi di lembaga tersebut.

Berdasarkan hasil wawancara, Panti Asuhan Artanita Al-Khairiyah masih mengalami kendala dalam pengelolaan donasi, terutama karena pencatatan masih dilakukan secara manual melalui buku tulis. Donasi *online* hanya sebatas transfer rekening tanpa pencatatan detail seperti tujuan atau catatan donatur. Penyebaran informasi mengenai kebutuhan donasi dan perkembangan anak-anak juga terbatas, hanya melalui banner dan unggahan sporadis di Instagram. Transparansi donasi belum optimal karena tidak tersedia laporan yang dapat diakses donatur. Selain itu, informasi daftar anak asuh hanya ditampilkan melalui banner, dan meskipun pendataan sudah dilakukan secara digital menggunakan Excel, belum terintegrasi dalam sistem yang mendukung akuntabilitas dan keterbukaan informasi.

Berdasarkan permasalahan diatas peneliti tertarik untuk membuat “Rancang Bangun Sistem Informasi Donasi di Panti Asuhan Artanita Al-Khairiyah Berbasis *Website*”. Dalam penelitian ini, penulis memilih menggunakan metode *Extreme Programming* yaitu salah satu pendekatan pengembangan perangkat lunak yang ringan termasuk dalam kategori metode *Agile*. *Extreme Programming* lebih cenderung menggunakan pendekatan berorientasi objek [5]. Tahapan-tahapan yang harus dilalui dalam *Extreme Programming* mencakup *Planning*, *Design*, *Coding*, dan *Testing*. Sasaran utama dari *Extreme Programming* adalah tim yang dibentuk dalam skala kecil hingga menengah, tanpa kebutuhan untuk menggunakan tim yang besar [6].

Penelitian ini mempermudah para donatur dalam mengakses informasi terkait panti asuhan, melakukan donasi secara online, meningkatkan tingkat transparansi sehingga mendapatkan kepercayaan dari para donatur dan membantu

meningkatkan akreditasi untuk Panti Asuhan Artanita Al-Khairiyah.

II. KAJIAN TEORI

A. *Extreme Programming (XP)*

Extreme Programming (XP) merupakan model pengembangan perangkat lunak yang banyak digunakan oleh para developer, diperkenalkan oleh Kent Block beserta Ron Jeffries sekitaran tahun 1996 pada saat ditunjuk oleh Chrysler untuk menangani sebuah proyek yang terancam akan gagal karena rumitnya sebuah sistem yang dibuat serta kegagalan pada saat tahap uji sistem. Metode XP membuat proyek tersebut menjadi lebih efisien, lebih fleksibel dan berhasil menyelesaikan proyek, XP terkesan merupakan sekumpulan ide lama yang sederhana, dan tidak memiliki efek apapun pada pengembangan perangkat lunak. Penerapan metode XP tidak selalu cocok pada setiap proyek pengembangan lunak hanya saja XP memiliki kelebihan menyesuaikan berbagai requirements yang tidak jelas dari klien sehingga sangat cocok untuk pengembangan perangkat lunak yang memerlukan penyesuaian cepat dalam perubahan – perubahan yang terjadi selama pengembangan perangkat lunak dan cocok untuk anggota tim yang tidak terlalu banyak [7].

B. *Unified Modeling Language (UML)*

Unified Modeling Language (UML) merupakan suatu standar yang digunakan untuk memvisualisasikan, mendokumentasikan, membangun, serta menspesifikasikan sistem perangkat lunak. UML dikembangkan sebagai respons terhadap kebutuhan industri perangkat lunak akan metode pemodelan yang dapat digunakan secara umum dalam pengembangan sistem berbasis objek [8].

C. React JS

React JS merupakan sebuah *library* JavaScript yang digunakan untuk membuat website dinamis dengan respons yang efisien, deklaratif, dan fleksibel. Dikembangkan oleh Facebook pada tahun 2013 berbasis ES6, React JS menggunakan JSX untuk mempermudah penulisan HTML dengan menambahkan struktur XML ke dalam JavaScript. Keuntungan utama menggunakan React terletak pada penggunaan virtual DOM, yang membantu mencapai efisiensi optimal dengan merender ulang node sesuai kebutuhan. Namun, kelemahan React JS adalah peralihannya dari modul berbasis kelas, yang dapat menghambat pendekatan Pemrograman Berorientasi Objek (OOP) [9].

D. *Black Box*

Black Box bisa digunakan untuk menguji fungsional sebuah perangkat lunak. Pengujian dilakukan secara tertulis untuk memeriksa apakah perangkat lunak berjalan sesuai yang diharapkan atau tidak. Pengujian fungsional meliputi seberapa baik sistem melaksanakan fungsinya, seperti perintah-perintah *user*, manipulasi sebuah data, pencarian dan proses bisnis, pengguna layar, dan integrasi. Pengujian *Black Box* untuk fungsional bisa juga meliputi permukaan yang jelas dari jenis fungsi-fungsi, serta operasi *backend* [10].

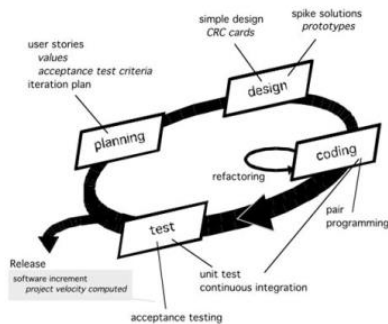
E. *User Acceptance Testing (UAT)*

User Acceptance Testing (UAT) merupakan tahap akhir dalam proses pengujian perangkat lunak yang bertujuan untuk memastikan bahwa sistem yang dikembangkan telah

sesuai dengan kebutuhan dan ekspektasi pengguna akhir. Pengujian ini dilakukan langsung oleh pengguna atau perwakilan pengguna dengan cara mengoperasikan sistem dalam kondisi nyata guna memverifikasi bahwa seluruh fungsionalitas berjalan sesuai dengan spesifikasi yang telah ditentukan [11].

III. METODE

Penelitian ini menggunakan metode pengembangan sistem *Extreme Programming* (XP). Metode *Extreme Programming* (XP) mendukung proses pengembangan perangkat lunak melalui penerapan prinsip transparansi serta pengelolaan waktu yang efisien, dengan batas kerja maksimal 40 jam per minggu. Pendekatan ini memungkinkan pelaksanaan dan pemantauan program dilakukan secara terarah, sehingga hasil pengembangan dapat disesuaikan dengan kebutuhan dan spesifikasi sistem yang telah ditetapkan [12]. Berikut merupakan gambar dari tahapan pengembangan sistem menggunakan metode XP yang disajikan pada Gambar 2 di bawah ini:



GAMBAR 2
Metode *Extreme Programming*

A. Planning (Perencanaan)

Planning merupakan tahapan awal untuk membangun sebuah sistem dengan kegiatan memahami konteks dari sebuah aplikasi, mengidentifikasi permasalahan, menganalisa kebutuhan, fungsi dari perangkat lunak yang akan dibuat dan menetapkan jadwal pelaksanaan pembangunan sistem serta alur pengembangan aplikasi [5][7].

B. Design (Perancangan)

Design merupakan tahapan perancangan sebuah sistem pemodelan dengan kegiatan membuat pemodelan sistem, pemodelan arsitektur dan pemodelan basis data [5]. Metodologi XP menggunakan desain sistem yang sederhana, lebih di fokuskan pada pembuatan program dan uji program.

C. Coding (Pengkodean)

Coding merupakan tahapan implementasi pemodelan yang sudah dibuat kedalam bentuk user interface dengan menggunakan bahasa pemrograman [5]. Pada tahap *coding* kegiatan yang dilakukan yaitu membuat sebuah basis data, dan membuat program.

D. Testing (Pengujian)

Testing merupakan tahapan pengujian sistem bilamana tahapan pengkodean telah selesai agar mengetahui kesalahan yang muncul saat aplikasi berjalan dan memastikan apakah sistem sudah sesuai dengan kebutuhan pengguna [5]. Pada XP *testing* bisa dilaksanakan kapan saja, tidak harus

menunggu program sudah selesai. Dapat dilakukan per modul bilamana terdeteksi kesalahan pada awal agar *programmer* bisa menangani dari awal [7]. Salah satu *output testing* dengan menggunakan *testing Black Box*.

E. Software Increment

Tahapan ini dilakukan bilamana pada tahapan *testing* sudah dinyatakan sesuai dengan kebutuhan bisnis maka aplikasi sudah siap untuk dirilis.

IV. HASIL DAN PEMBAHASAN

Hasil dan pembahasan penelitian ini menyajikan hasil implementasi dari sistem donasi yang dirancang menggunakan metode *Extreme Programming*. Proses pengembangan dilakukan secara bertahap, dimulai dari *planning* hingga proses *software increment* sistem yang dibangun.

A. Planning

User requirement atau Pemetaan kebutuhan pengguna dilakukan dengan observasi untuk menganalisis berbagai aktivitas yang diperlukan pada setiap peran yang ada. Setiap fitur yang dibutuhkan diidentifikasi berdasarkan fungsionalitas yang harus ada dalam *platform*, yang kemudian diprioritaskan untuk memastikan pengembangan sistem berjalan dengan efisien dan efektif. Pemetaan kebutuhan ini menjadi acuan utama dalam merancang dan mengimplementasikan fitur-fitur pada *platform*. Hasil pemetaan kebutuhan tersebut disajikan dalam *User requirement* di bawah ini.

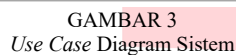
TABEL 1
User Requirement

Fitur	User (Donatur)	Admin (Pengelola)
Melakukan Donasi	✓	
Melihat Daftar Donatur yang Sudah Berdonasi	✓	
Melihat Program Donasi	✓	
Melihat Aktivitas	✓	
Melihat Sejarah	✓	
Melihat Transparansi Donasi	✓	
Melihat Legalitas	✓	
Menghubungi Kontak Admin	✓	
Melihat Lokasi Panti Asuhan	✓	
Melihat Visi, Misi dan Tujuan	✓	
Melihat Daftar Pengurus & Pengasuh	✓	
Melihat Daftar Anak-anak Panti	✓	
Mengelola Program Donasi		✓
Mengelola Aktivitas		✓
Mengelola Pengurus & Anak-anak Asuh (User)		✓
Melihat Dashboard Donasi		✓

B. Desain Sistem

Pada tahap ini berisi perancangan alur kerja dan struktur sistem berdasarkan kebutuhan yang telah dianalisis. Desain mencakup *Use Case Diagram*, *Timeline Pengembangan*, *Activity Diagram*, *Class Diagram*, *ERD*, dan Desain

	Pekan 12		berdasarkan <i>feedback</i> .
--	----------	--	-------------------------------



```

graph TD
    subgraph User
        Start(( )) --> Home1[Home]
        Home1 --> Donasi[Donasi]
        Donasi --> PengisianData[Pengisian Data]
        PengisianData --> Validasi{Validasi}
        Validasi -- "Tidak Valid" --> PengisianData
        Validasi -- "Valid" --> MenampilkanDetail[Menampilkan Informasi Detail]
        MenampilkanDetail --> PengisianData
        MenampilkanDetail --> MenampilkanTransaksi[Menampilkan Transaksi]
        MenampilkanTransaksi --> MelakukanPembayaran[Melakukan Pembayaran]
        MelakukanPembayaran --> Home2[Home]
        Home2 --> End((( )))
    end

    subgraph Sistem
        MenampilkanFormDonasi[Menampilkan Form Donasi]
        MenampilkanDetail
        MenampilkanTransaksi
        MenampilkanStatus[Menampilkan Status]
    end

    subgraph Midtrans
        MengirimkanURL[Mengirimkan URL transaksi]
        MengirimkanNotifikasi[Mengirimkan Notifikasi Transaksi]
    end

    Donasi --> MenampilkanFormDonasi
    MenampilkanFormDonasi --> PengisianData
    MenampilkanTransaksi --> MengirimkanURL
    MengirimkanURL --> MelakukanPembayaran
    MelakukanPembayaran --> MenampilkanStatus
    MenampilkanStatus --> MenampilkanDetail
    MenampilkanStatus --> MengirimkanNotifikasi
    MengirimkanNotifikasi --> MenampilkanDetail
  
```

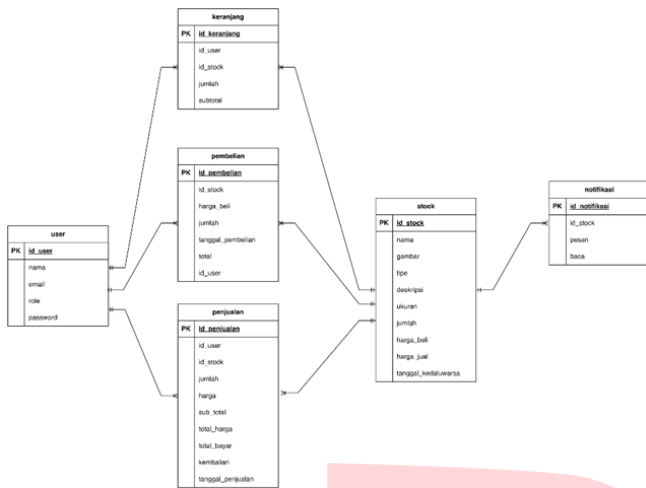
Pada Gambar 4 divisualisasikan alur bisnis proses untuk melakukan sebuah donasi dari sisi *user* atau donatur. Dapat dilihat bahwa proses diawali ketika *user* sudah mengunjungi *website* dengan diarahkan ke *homepage* atau *landing page*. Selanjutnya, *user* memilih menu Donasi untuk memulai proses donasi. Setelah itu, sistem akan menampilkan formulir donasi yang harus diisi oleh *user*. *User* kemudian mengisi data yang diperlukan, seperti nominal donasi, nama, email, dan informasi lainnya yang dibutuhkan. Data yang telah diisi tersebut kemudian melalui proses validasi oleh sistem.

```

classDiagram
    class donation {
        +uid: integer
        +name: varchar
        +address: varchar
        +phone_number: integer
        +email: varchar
        +amount: integer
        +message: text
        +status: integer
        +payment_type: varchar
        +interaction_time: datetime
    }
    class program_donation {
        +uid: integer
        +title: varchar
        +description: text
        +goal_amount: integer
        +current_amount: integer
        +image: varchar
    }
    class admin {
        +uid: integer
        +name: varchar
        +email: varchar
        +password: varchar
    }
    class card_payment {
        +createPayment()
        +updatePayment()
        +cancelPayment()
        +viewDetails()
    }
    class onpage_user {
        +uid: integer
        +name: varchar
        +address: varchar
        +age: integer
        +education: varchar
        +position: varchar
        +image: varchar
    }
    class onpage_activity {
        +uid: integer
        +title: varchar
        +description: text
        +location: varchar
        +time: varchar
        +video: varchar
        +image: varchar
    }
    donation "0..*" -- "1" program_donation
    donation "0..*" -- "1" admin
    donation "0..*" -- "0..*" onpage_user
    donation "0..*" -- "0..*" onpage_activity
    program_donation "1" -- "0..*" onpage_activity
    admin "1" -- "0..*" onpage_activity
    
```

Gambar 5 memperlihatkan *class* diagram dari sistem pengelolaan donasi dan informasi panti asuhan, yang terdiri dari enam kelas utama, yaitu: *Admin*, *Program_Donation*, *Donation*, *Orphanage_User*, *Orphanage_Activity*, dan hubungan antar kelas yang membentuk struktur data inti dalam sistem.

<i>Sprint/Iterasi</i>	Durasi	Aktivitas Utama	Hasil Akhir
1	Pekan 1 s/d Pekan 2	Analisis Kebutuhan & Perancangan Sistem	Rancangan teknologi yang digunakan, <i>Database</i> fitur yang skema
2	Pekan 3 s/d Pekan 4	Implementasi <i>Backend</i> untuk <i>Admin</i>	API untuk fitur-fitur <i>Admin</i> + pengujian awal (unit test)
3	Pekan 5 s/d Pekan 6	Implementasi <i>Backend</i> untuk <i>User</i>	API untuk fitur <i>user</i> + pengujian awal
4	Pekan 7 s/d Pekan 8	Integrasi <i>Frontend</i> untuk <i>Admin</i>	Fitur <i>Admin</i> yang terintegrasi dan berjalan di UI
5	Pekan 9 s/d Pekan 10	Integrasi <i>Frontend</i> untuk <i>User</i>	Fitur-fitur <i>user</i> yang dapat diakses via UI (<i>frontend-backend</i> terhubung)
6	Pekan 11 s/d	<i>Adjustment</i> Fitur-fitur dan Pengujian	<i>Website</i> siap digunakan, fitur telah diuji dan disesuaikan



GAMBAR 6
ERD Sistem

Pada Gambar 6 menggambarkan struktur *Database* yang dirancang untuk mendukung operasional *website* sistem informasi donasi. Setiap entitas memiliki atribut-atribut yang relevan untuk menyimpan data terkait. Relasi antar entitas menggambarkan bagaimana data pada tabel-tabel ini saling terhubung.

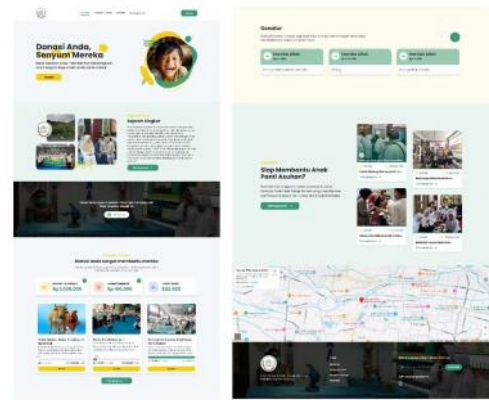


GAMBAR 7
Desain *Wireframe* Halaman Kelola Stok Barang

Pada Gambar 7 merupakan *wireframe* halaman program donasi yang menampilkan daftar program secara terstruktur, lengkap dengan informasi jumlah donatur, total donasi terkumpul, dan tombol aksi untuk berdonasi, sebagai acuan desain antarmuka pengguna

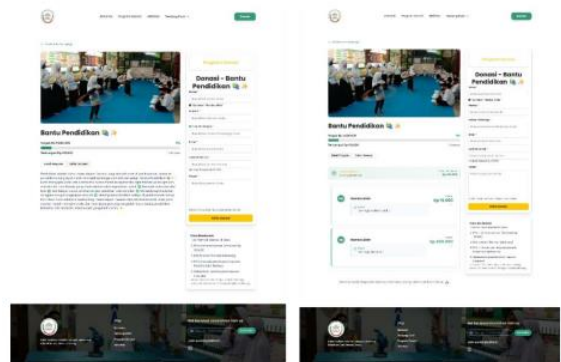
C. Implementasi Sistem

Pada tahapan ini, implementasi dilakukan berdasarkan rancangan sistem yang telah dirancang pada tahap sebelumnya. Implementasi mencakup pengembangan berbagai komponen sistem yang terdiri dari *frontend*, *backend*, *Database*, dan integrasi antar komponen. Semua komponen diimplementasikan sesuai dengan spesifikasi teknis yang telah ditentukan agar menghasilkan sistem yang dapat berfungsi dengan baik.



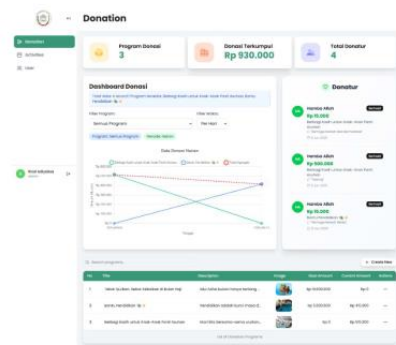
GAMBAR 8
Desain Antarmuka Halaman Beranda

Pada Gambar 8 ditampilkan beranda (*homepage*) dari sistem informasi donasi yang dirancang sebagai halaman utama ketika pengguna pertama kali mengakses *website*. Halaman ini menampilkan *button* berdonasi, informasi singkat mengenai latar belakang panti asuhan, serta ringkasan statistik donasi yang telah terkumpul. Selain itu, pengguna juga dapat melihat program donasi, aktivitas panti asuhan dan peta lokasi.



GAMBAR 2
Desain Antarmuka Detail Program Donasi

Pada Gambar 9 menampilkan halaman detail program donasi yang berfungsi untuk memberikan informasi lengkap kepada pengguna mengenai salah satu program yang dipilih. Pada bagian kiri halaman ditampilkan nama program, target donasi, jumlah yang sudah terkumpul, serta deskripsi program secara rinci. Sementara itu, pada sisi kanan terdapat formulir donasi yang memungkinkan pengguna langsung memberikan donasi tanpa kembali ke halaman utama.



GAMBAR 10
Desain Antarmuka Halaman *Dashboard* Donasi

Pada Gambar 10 menampilkan halaman *Dashboard Admin* Donasi, yang digunakan oleh *Admin* untuk memantau seluruh aktivitas donasi dalam sistem. Di bagian atas terdapat ringkasan jumlah program donasi, total donasi yang terkumpul, dan jumlah donatur. Tengah halaman menampilkan grafik donasi yang dapat difilter berdasarkan program dan periode waktu. Di sisi kanan, terdapat daftar donatur yang sudah melakukan donasi. Bagian bawah menampilkan tabel daftar program donasi dilengkapi fitur pencarian dan tombol aksi untuk pengelolaan data.

Tahap selanjutnya adalah menyajikan hasil dari masing-masing pengujian. Hasil ini mencerminkan bagaimana sistem merespons terhadap skenario yang telah dirancang, baik dari sisi fungsionalitas melalui *Black Box Testing*, maupun dari sisi pengalaman pengguna melalui *User Acceptance Testing* (UAT). Pengujian *Black Box* dilakukan oleh dua perwakilan, masing-masing satu orang untuk bagian *user* dan satu orang untuk bagian *admin*. Hasil pengujian untuk bagian *user* disajikan pada Tabel 3 berikut ini.

TABEL 3
Hasil Pengujian *Black Box User*

Penguji: Andika Purnama Putra Profesi: FrontEnd Developer				
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil yang Didapatkan	Keterangan
1.	Mengisi semua kolom pada <i>form</i> donasi dengan data valid	Sistem membuka halaman Kasir	Sistem menyimpan data donasi, menampilkan konfirmasi, dan mengarahkan ke halaman pembayaran	Berhasil
2.	Mengisi semua kolom pada <i>form</i> donasi dengan data tidak valid	Sistem menampilkan pesan kesalahan dan meminta perbaikan data	Sistem menampilkan pesan kesalahan dan meminta perbaikan data	Berhasil
3.	Mengklik tombol "Lanjutkan ke Pembayaran" setelah data donasi valid	Sistem mengarahkan pengguna ke halaman pembayaran Midtrans	Sistem mengarahkan pengguna ke halaman pembayaran Midtrans	Berhasil
4.	Mengklik tombol "Donasi Baru" setelah data donasi valid	Sistem mereset semua isian dan menampilkan <i>form</i> kosong	Sistem mereset semua isian dan menampilkan <i>form</i> kosong	Berhasil
5.	Melakukan proses pembayaran donasi melalui Midtrans	Sistem menampilkan halaman pembayaran	Sistem menampilkan halaman pembayaran hanya masih menggunakan sandbox	Gagal

Tabel 3 menunjukkan hasil pengujian *Black Box Testing* yang berhasil. Dari total 24 skenario pengujian terhadap fitur-fitur sisi *user* (donatur), mayoritas fungsi berjalan dengan baik dan sesuai dengan ekspektasi tanpa adanya gangguan

terhadap alur fungsionalitas. Hasil pengujian untuk bagian *Admin* disajikan pada Tabel 4 berikut ini.

TABEL 4
Hasil Pengujian *Black Box Admin*

Penguji: Deo Septry Wengi Profesi: FrontEnd Developer				
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil yang Didapatkan	Keterangan
1.	<i>Login</i> sebagai <i>Admin</i> dengan kredensial valid	Sistem mengarahkan ke dashboard	Sistem mengarahkan ke dashboard	Berhasil
2.	<i>Login</i> dengan kredensial tidak valid	Sistem menolak akses dan menampilkan pesan kesalahan	Sistem menolak akses dan menampilkan pesan	Berhasil
3.	Mengakses dashboard <i>Admin</i>	Menampilkan data dan grafik donasi	Menampilkan data dan grafik donasi	Berhasil
4.	Menambahkan program donasi baru	Program baru tersimpan dan muncul di daftar	Program baru tersimpan dan muncul di daftar	Berhasil
5.	Menampilkan tabel program donasi	Menampilkan daftar lengkap program donasi	Menampilkan daftar lengkap program donasi	Berhasil

Tabel 4 menunjukkan hasil pengujian *Black Box Testing* terhadap fitur-fitur pada sisi *Admin* yang seluruhnya berjalan dengan baik. Dari total 27 skenario pengujian, tidak terdapat satu pun fitur yang gagal atau mengalami kendala saat dilakukan pengujian. Semua proses yang melibatkan autentikasi, pengelolaan data program donasi *user*, *user* (pengurus & anak asuh), hingga fitur pencarian, *filter*, dan paginasi berhasil dijalankan sesuai alur dan fungsionalitas yang dirancang sebelumnya.

Berdasarkan skenario pengujian *Black Box Testing* yang telah dilakukan terhadap berbagai fitur dalam sistem, tercatat sebanyak 51 skenario uji yang mencakup pengujian fungsionalitas untuk peran *Admin* dan *user*. Dari keseluruhan skenario tersebut, 50 skenario berhasil dijalankan dengan baik, sedangkan 1 skenario mengalami kegagalan. Secara keseluruhan, sistem menunjukkan tingkat keberhasilan sebesar 98% berdasarkan hasil pengukuran dibawah ini:

$$\text{Persentase skor} = \left(\frac{50}{51} \right) \times 100\% = 98\%$$

Setelah pengujian *Black Box* tahap selanjutnya yaitu pengujian *User Acceptance Testing* (UAT). Secara umum, hasil UAT menunjukkan bahwa mayoritas fitur utama sistem berjalan dengan baik dan diterima oleh pengguna. Pada sisi *Admin*, kedua responden menyatakan bahwa proses login, pengelolaan program donasi, aktivitas, dan data *user* dapat dilakukan tanpa kendala berarti. Berikut hasil dari *pengujian User Acceptance Testing* (UAT) bagian *user* yang dapat dilihat pada Tabel 5 di bawah ini.

TABEL 5
Nilai Rata-rata dan Hasil Persentase UAT *User*

No	Kode	Nilai Rata-rata	Persentase
1	U-F1	4.85	97%

2	U-F2	4.95	99%
3	U-F3	4.75	95%
4	U-F4	4.70	94%
5	U-F5	4.75	95%
6	U-N1	4.95	99%
7	U-N2	4.975	99.5%
8	U-N3	4.70	94%
9	U-N4	4.755	95.5%
10	U-N5	4.875	97.5%

Tabel 5 menampilkan hasil rata-rata pengujian *User Acceptance Testing* (UAT) yang dilakukan oleh 41 responden. Dalam perhitungan, jumlah respon untuk setiap skala dikalikan dengan bobotnya, menghasilkan nilai pembobotan per kategori. Setelah seluruh data diproses, dilakukan perhitungan rata-rata dari keseluruhan persentase untuk mendapatkan gambaran umum tingkat penerimaan sistem secara keseluruhan melalui pengujian UAT. Berdasarkan hasil pengujian, rata-rata keseluruhan pada sistem di bagian *user* menunjukkan nilai sebesar 96,55%. Berikut hasil dari pengujian *User Acceptance Testing* (UAT) bagian admin yang dapat dilihat pada Tabel 6 di bawah ini.

TABEL 6
Nilai Rata-rata dan Hasil Persentase UAT *Admin*

No	Kode	Nilai Rata-rata	Persentase
1	A-F1	4.6	92%
2	A-F2	4.6	92%
3	A-F3	3.8	76%
4	A-F4	5.0	100%
5	A-F5	4.8	96%
6	A-F6	5.0	100%
7	A-F7	5.0	100%
8	A-F8	3.4	68%
9	A-F9	4.8	96%
10	A-F10	4.8	96%
11	A-F11	5.0	100%
12	A-F12	5.0	100%
13	A-F13	3.6	72%
14	A-F14	5.0	100%
15	A-F15	4.4	88%
16	A-F16	4.8	96%
17	A-F17	4.8	96%
18	A-N1	5.0	100%
19	A-N2	5.0	100%
20	A-N3	5.0	100%
21	A-N4	4.8	96%
22	A-N5	4.4	88%

Tabel 6 menunjukkan bahwa sebagian besar fitur pada sisi *Admin* telah berjalan dengan baik dan mendapatkan respons positif dari responden. Proses perhitungannya dilakukan dengan metode yang sama seperti pada bagian *User*. Berdasarkan hasil pengujian, rata-rata keseluruhan pada sistem di bagian *admin* menunjukkan nilai sebesar 91,6%.

Berdasarkan hasil *User Acceptance Testing* (UAT), sistem informasi donasi ini telah memenuhi sebagian besar kebutuhan pengguna, baik dari sisi fungsionalitas maupun kemudahan penggunaan. Hasil rata-rata skor UAT menunjukkan tingkat kepuasan sebesar 96,55% untuk bagian *user* dan 91,6% untuk bagian *Admin*, yang keduanya termasuk dalam kategori "Sangat Baik" berdasarkan kriteria interpretasi skor.

Setelah sistem dinyatakan lolos pengujian, dilakukan *software increment* sebagai tanda bahwa versi tersebut siap digunakan. Sistem kemudian di-*deploy* ke server produksi dan telah diserahkan kepada pihak panti asuhan. Meskipun

sudah dapat digunakan dan proses pengembangan masih berlanjut, terutama untuk melengkapi data yang dibutuhkan oleh pihak panti guna memenuhi persyaratan pendaftaran ke layanan pembayaran Midtrans. Tahap ini juga membuka peluang iterasi berikutnya agar sistem semakin sempurna sesuai kebutuhan pengguna.

V. KESIMPULAN

Penelitian ini berhasil merancang dan mengembangkan sistem informasi donasi berbasis *website* untuk Panti Asuhan Artanita Al-Khairiyah, yang mendukung pengelolaan program donasi, aktivitas panti, serta transparansi dana donasi. Sistem ini dibangun dengan metode *Extreme Programming* (XP), yang terdiri dari tahapan *planning*, *design*, *coding*, dan *testing*. Proses pengembangan berlangsung dalam iterasi-iterasi terstruktur guna menghasilkan perangkat lunak yang adaptif terhadap kebutuhan pengguna. Teknologi utama yang digunakan yaitu Golang pada sisi backend, React dengan TypeScript untuk *frontend*, Tailwind CSS dan shadcn/ui untuk antarmuka, serta PostgreSQL sebagai basis data utama.

Platform ini berhasil menghadirkan fungsionalitas utama yang memudahkan pengurus dalam manajemen donasi serta menyampaikan informasi secara transparan kepada para donatur. Fitur-fitur seperti pengelolaan program donasi, pencatatan aktivitas, informasi anak asuh dan pengurus, sistem transaksi donasi uang secara online, hingga visualisasi data di *dashboard* telah berjalan dengan baik. Setiap pengguna memiliki hak akses yang sesuai, yaitu sebagai *Admin* (pengelola) dan *user* (donatur). Melalui pengujian Black Box Testing yang dilakukan oleh dua perwakilan pengguna (*Admin* dan *user*) dan UAT (*User Acceptance Testing*) yang dilakukan oleh 41 responden *user* serta 5 responden untuk *admin*.

REFERENSI

- [1] A. Ampuh Yunanto, H. Abdullah Taufik, A. Juhda Azmi, and A. Labib, "RANCANG BANGUN SISTEM INFORMASI MANAJEMEN PANTI ASUHAN AL-KAHFI SURABAYA".
- [2] J. Pengabdian, K. Masyarakat, O. Sukmana, H. Agustino, and W. Hidayat, "PENDAMPINGAN PENGELOLAAN PANTI ASUHAN PUTRI AISYIYAH KOTA MALANG DALAM UPAYA PERSIAPAN AKREDITASI LEMBAGA KESEJAHTERAAN SOSIAL ANAK (LKSA)," 2021.
- [3] M. Fikri, A. Shiddiqy, and N. Ekawati, "Aplikasi Donasi Anak Yatim Berbasis Web Menggunakan Payment Gateway."
- [4] "PERATURAN MENTERI SOSIAL REPUBLIK INDONESIA."
- [5] S. Kasus, : Yayasan, A. Khair, Y. Yoga, D. Martin, and I. Handayani, "OKTAL : Jurnal Ilmu Komputer dan Science Implementasi Scan QR Code Pada Sistem Donasi Anak Yatim Berbasis *Mobile* Menggunakan *Metode Extreme Programming*," vol. 2, no. 2, 2023.
- [6] R. Haykal Chorul Fikri and M. J. Bagoes Satria, "Analisa Dan Perancangan Sistem Informasi Penerimaan Donasi *Metode Extreme Programming*

Di LSM Desa Klunggen.” [Online]. Available: <https://jurnal.publikasitecno.id/index.php/jim>

- [7] I. G. N. Suryantara, *Merancang Aplikasi dengan Metodologi Extreme Programmings*. Jakarta: PT Elex Media Komputindo, 2017.
- [8] A. Voutama, U. Singaperbangsa Karawang, J. H. Ronggo Waluyo, T. Timur, and J. Barat Kode, “INFORMATIKA PEMANFAATAN UML (UNIFIED MODELING LANGUAGE) DALAM PERENCANAAN SISTEM PENYEWAAN BAJU ADAT BERBASIS WEBSITE,” *Jurnal Informatika, Manajemen dan Komputer*, vol. 14, no. 2, 2022.
- [9] I. Madurapperuma, M. Shafana, and M. Sabani, *State-of-Art Frameworks for Front-end and Back-end Web Development*. 2020.
- [10] J. Sinarmata, *Rekayasa Perangkat Lunak*. Yogyakarta: C. V ANDI OFFSET(penerbit ANDI), 2009.
- [11] D. Nurvalentini *et al.*, “Pengujian Aplikasi Homestay Kosasih: Analisis Fungsionalitas Dan Penerimaan Pengguna Menggunakan Metode Black Box Dan UAT Testing Homestay Kosasih Application: *Analysis Of Functionality And User Acceptance Using Black Box And UAT Methods*,” ONLINE.
- [12] M. Fery Fajar *et al.*, “PENGEMBANGAN SISTEM INFORMASI PEGAWAI DENGAN METODE EXTREME PROGRAMMING (STUDI KASUS: ITSK RS DR. SOEPRAOEN MALANG),” 2024.