

SISTEM PEMINJAMAN ALAT BANTU MEDIS BERBASIS WEB MENGGUNAKAN REACT JS DAN GO

Diajukan untuk Memenuhi Kelulusan Matakuliah Proyek 3
Program Studi DIV Teknik Informatika

DISUSUN OLEH :

Moch Restu Agis Burhanudin (714230059)

Aghni Hasna Mufida (714230069)



Universitas Logistik & Bisnis Internasional

PROGRAM STUDI DIV TEKNIK INFORMATIKA
UNIVERSITAS LOGISTIK DAN BISNIS INTERNASIONAL
BANDUNG
2026

LEMBAR PENGESAHAN

Laporan Proyek III ini telah diperiksa, disetujui dan disidangkan
di Bandung,/...../2026

Oleh:

Penguji Pendamping,

Penguji Utama,

Rolly Maulana Awangga, S.T., M.T. Nisa Hanum Harani, S.Kom., M.T.

NIK : 117.86.219

NIK : 117.89.223

Pembimbing,

Koordinator Proyek II

Nisa Hanum Harani, S.Kom., M.T. Roni Habibi, S.Kom., M.T., SFPC

NIK : 117.89.223

NIK : 103.78069

Menyetujui,

Ketua Program Studi D-IV

Teknik Informatika

Roni Andarsyah, S.T., M.Kom.

NIK : 115.88.193

SURAT PERNYATAAN TIDAK MELAKUKAN PLAGIARISME

Nama : Moch Restu Agis Burhanudin
NPM : 714230059
Program Studi : D-IV Teknik Informatika
Judul : Sistem Pengembangan dan Rental Media Berbasis Web
Menggunakan React.js dan GO

Menyatakan bahwa:

1. Proyek Pengembangan Aplikasi (Proyek 3) yang saya susun merupakan karya asli yang dibuat secara mandiri dan belum pernah dipublikasi sebagai syarat kelulusan pada Program Studi D-IV Teknik Informatika, Politeknik Logistik dan Bisnis Internasional seyogyanya di persyaratan tinggi lainnya. Proyek ini berkenaan dengan sistem, perangkat, dan teknologi yang terkini dan lainnya sesuai dengan aturan dan dosen pembimbing sesuai ketentuan yang berlaku.
2. Seluruh isi dalam proyek ini tidak sepenuhnya karya saya sendiri melainkan hasil kerja pengembangan seminar secara jujur. Setiap rujukan terhadap karya ilmiah atau pendapat orang lain belum didokumentasikan secara tertulis dan dalam rangka membatilkan sama penggunaan yang didekumsikan ke dalam daftar pustaka sesuai kaidah penulisan ilmiah.
3. Pernyataan ini saya buat dengan sebetar-benarnya. Apabila di kemudian hari terbukti pernyataan saya tidak benar, maka saya bersedia menerima sanksi terbulat yang berlaku di lingkungan persyaratan tinggi.

BANDUNG, 2026
Yang membuat pernyataan,

Moch Restu Agis Burhanudin
NPM : 714230059

SURAT PERNYATAAN TIDAK MELAKUKAN PLAGIARISME

Nama : Aghni Hasna Mufida
NPM : 714230069
Program Studi : D-IV Teknik Informatika
Judul : Sistem Pengembangan dan Rental Media Berbasis Web
Menggunakan React.js dan GO

Menyatakan bahwa:

1. Proyek Pengembangan Aplikasi (Proyek 3) yang saya susun merupakan karya asli yang dibuat secara mandiri dan belum pernah dipublikasi sebagai syarat kelulusan pada Program Studi D-IV Teknik Informatika, Politeknik Logistik dan Bisnis Internasional seyogyanya di persyaratan tinggi lainnya. Proyek ini berkenaan dengan sistem, perangkat, dan teknologi yang terkini dan lainnya sesuai dengan aturan dan dosen pembimbing sesuai ketentuan yang berlaku.
2. Seluruh isi dalam proyek ini tidak sepenuhnya karya saya sendiri melainkan hasil kerja pengembangan seminar secara jujur. Setiap rujukan terhadap karya ilmiah atau pendapat orang lain belum didokumentasikan secara tertulis dan dalam rangka membatilkan sama penggunaan yang didekumsikan ke dalam daftar pustaka sesuai kaidah penulisan ilmiah.
3. Pernyataan ini saya buat dengan sebetar-benarnya. Apabila di kemudian hari terbukti pernyataan saya tidak benar, maka saya bersedia menerima sanksi terbulat yang berlaku di lingkungan persyaratan tinggi.

BANDUNG, 2026
Yang membuat pernyataan,

Aghni Hasna Mufida
NPM : 714230069

ABSTRAK

Sebagian besar masyarakat menghadapi masalah aksesibilitas alat bantu medis, terutama terkait biaya dan keterbatasan informasi stok di fasilitas kesehatan. Tujuan dari penelitian ini adalah untuk membangun MedisLink, sebuah sistem informasi berbasis *web* yang menghubungkan peminjaman dan donasi alat medis. Sistem ini akan memudahkan hubungan antara donatur dan pengguna yang membutuhkan secara jelas dan efektif. Pengembangan teknologi sistem ini menggunakan arsitektur yang terpisah dengan *React.js* di sisi depan, *Golang (Fiber)* di sisi *backend*, dan *MongoDB* sebagai basis data non-relasional. Struktur *API* yang responsif dan terbagi berdasarkan layanan fungsional mendukung efisiensi sistem dan menjaga keamanan akses pengguna melalui autentikasi *JSON Web Token (JWT)*. Sistem notifikasi pengembalian alat otomatis adalah salah satu fitur unggulan yang memastikan waktu distribusi inventaris yang tepat dengan menggunakan mekanisme *Cron Job* dengan interval waktu harian.

Kata Kunci: MedisLink, *React.js*, *Go (Fiber)*, *MongoDB*, *JWT*, *Cron Job*.

ABSTRACT

Most people face accessibility issues for medical devices, particularly regarding costs and limited stock information at healthcare facilities. The purpose of this research is to build *MedisLink*, a *web-based information system* that connects medical equipment lending and donation. This system will facilitate clear and effective relationships between donors and users in need. The system technology development uses a *separated architecture* with *React.js* on the *frontend*, *Golang (Fiber)* on the *backend*, and *MongoDB* as a *non-relational database*. The responsive *API* structure divided by *functional services* supports *system efficiency* and maintains *user access security* through *JSON Web Token (JWT) authentication*. The automatic equipment return *notification system* is one of the featured capabilities that ensures proper *inventory distribution* timing using *Cron Job mechanisms* with daily intervals.

Keywords: *MedisLink, React.js, Go (Fiber), MongoDB, JWT, Cron Job.*

KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga kami dapat menyelesaikan Proyek III ini. Penulisan proyek ini merupakan bagian dari upaya saya menuju gelar Sarjana Terapan Program Studi Teknik Informatika Universitas Logistik & Bisnis Internasional Bandung. Kami menyadari bahwa hasil ini merupakan hasil dukungan dan bimbingan yang kami terima dari semua pihak mulai dari masa perkuliahan hingga penyusunan proyek III ini. Oleh karena itu, kami mengucapkan terima kasih kepada semua pihak yang telah mendukung dan membimbing kami.

DAFTAR ISI

Daftar Isi

ABSTRAK	1
ABSTRACT	2
KATA PENGANTAR	3
BAB I PENDAHULUAN	11
0.1 Latar Belakang	11
0.2 Identifikasi Masalah	12
0.3 Tujuan	12
0.4 Ruang Lingkup	13
0.4.1 Cakupan Fungsional	13
0.4.2 Cakupan Teknis	13
0.4.3 Cakupan Pengujian	14
0.4.4 Batasan Penelitian	14
0.4.5 Luaran Penelitian	15
0.5 Sistematika Penulisan	15
BAB II LANDASAN TEORI	16
0.6 Platform Donasi Digital	16
0.7 Arsitektur Aplikasi Web Modern	17
0.7.1 Arsitektur Three-Tier/N-Tier	17
0.7.2 Konsep Client-Server	18
0.8 REST API dan Protokol HTTP	18
0.9 Autentikasi dan Otorisasi	18

0.9.1	OAuth 2.0	18
0.9.2	OpenID Connect	19
0.9.3	JWT	19
0.9.4	Skema Bearer Token	19
0.9.5	Keamanan API	20
0.10	Database NoSQL (MongoDB)	20
0.11	Unified Modeling Language (UML)	21
0.11.1	Use Case Diagram	21
0.11.2	Activity Diagram	22
0.11.3	Sequence Diagram	22
0.11.4	Class Diagram	22
BAB III METODE PENELITIAN		22
0.12	Metode Pengembangan Sistem	23
0.12.1	Konsep Agile Development	23
0.12.2	Alur Iterasi Pengembangan	23
0.12.3	Penerapan Agile pada Proyek Nyumbangin	24
0.13	Analisis Kebutuhan	24
0.13.1	Sumber Kebutuhan	24
0.13.2	Kebutuhan Fungsional	25
0.13.3	Kebutuhan Non-Fungsional	25
0.14	Perancangan Sistem	26
0.14.1	Arsitektur Logis	26
0.14.2	Arsitektur Fisik	26
0.14.3	Arsitektur Teknologi	27
0.14.4	Modul Utama	28
0.14.5	Strategi Desain	28
0.15	Pemodelan Sistem	28
0.15.1	Use Case	29
0.15.2	Activity Diagram	29
0.15.3	Activity Diagram Proses Payout	31
0.15.4	Activity Diagram Feedback	32
0.15.5	Activity Diagram Login	33
0.15.6	Activity Diagram Kelola Overlay	34
0.15.7	Activity Diagram Kelola Profil	35
0.15.8	Activity Diagram Request Payout	36
0.15.9	Activity Diagram Riwayat Donasi	37
0.15.10	Sequence Diagram	38

0.15.11 Model Koleksi	45
0.15.12 Gopay Merchant	46
0.15.13 Perancangan Mekanisme Macrodroid Trigger Webhook . .	46
0.15.14 Integrasi Payment Gateway Midtrans	48
0.16 Metode Perancangan Teknis	48
0.17 Metode Pengujian	49
0.17.1 Jenis Pengujian	49
0.17.2 Skenario Pengujian	50
0.17.3 Alat Pengujian	51
0.18 Evaluasi Keberhasilan	51
0.18.1 Cakupan Pengujian	51
0.18.2 Hasil Code Coverage	52
0.18.3 Interpretasi dan Evaluasi	52
0.18.4 Kesimpulan Evaluasi	53
BAB IV HASIL DAN PEMBAHASAN	53
0.19 Halaman Autentikasi	54
0.20 Halaman Donasi	55
0.20.1 Form Donasi	55
0.20.2 Pembayaran Donasi Menggunakan QRIS Gopay Merchant	56
0.20.3 Halaman Pembayaran Midtrans	57
0.20.4 Fitur Sharelink Donasi	58
0.21 Dashboard Kreator	59
0.22 Halaman Request Payout	59
0.23 Overlay	60
0.23.1 Overlay Notifikasi Donasi	60
0.23.2 Overlay Media Share	61
0.23.3 Overlay QR Link Donasi	62
0.23.4 Overlay Leaderboard	63
0.24 Dashboard Admin	63
0.24.1 Halaman Dashboard	64
0.24.2 Halaman Creator	64
0.24.3 Halaman Payout	65
0.25 Struktur Basis Data	66
BAB V KESIMPULAN DAN SARAN	67
0.26 Kesimpulan	67

DAFTAR TABEL

Daftar Tabel

1	Code Coverage	52
---	-------------------------	----

DAFTAR GAMBAR

Daftar Gambar

1	Arsitektur Teknologi	27
2	Use Case Diagram	29
3	Activity Diagram Donasi	30
4	Activity Diagram Payout	31
5	Activity Diagram Feedback	32
6	Activity Diagram Login	33
7	Activity Diagram Kelola Overlay	34
8	Activity Diagram Kelola Profil	35
9	Activity Diagram Request Payout	36
10	Activity Diagram Riwayat Donasi	37
11	Sequence Diagram Donasi	38
12	Sequence Diagram Leaderboard	39
13	Sequence Diagram Login	40
14	Sequence Diagram Feedback	41
15	Sequence Diagram Kelola Profil	42
16	Sequence Diagram Riwayat Donasi	43
17	Sequence Diagram Request Payout	44
18	MacroDroid	47
19	Halaman Login	54
20	Form Donasi	55
21	QRIS Gopay Merchant	56
22	Halaman Pembayaran Midtrans	57
23	Fitur Sharelink Donasi	58
24	Dashboard Kreator	59

25	Halaman Payout	59
26	Notifikasi Donasi	60
27	Media Share	61
28	QR Link Donasi	62
29	Leaderboard Donatur	63
30	Dashboard Admin	64
31	Tabel Daftar Creator dan Detail Creator	64
32	Tampilan Payout	65
33	Basis Data	66

BAB I

PENDAHULUAN

0.1 Latar Belakang

Keterbatasan akses terhadap alat bantu medis di Indonesia masih menjadi masalah kesehatan masyarakat yang signifikan. Banyak penyandang disabilitas dan pasien dengan kebutuhan medis khusus mengalami kesulitan mendapatkan alat bantu seperti kursi roda, tongkat oksigen, dan alat rehabilitasi lainnya karena faktor ekonomi dan ketersediaan alat di fasilitas kesehatan yang terbatas. Akses yang terbatas ini tidak hanya memperburuk kondisi kesehatan pengguna tetapi juga berdampak pada tingkat kemandirian dan risiko kemalangan akibat tidak tersedianya kebutuhan dasar medis.

Banyak fasilitas kesehatan, terutama rumah sakit umum, sering kali kekurangan stok alat bantu medis yang dibutuhkan oleh pasien, sementara kemampuan masyarakat untuk membeli alat tersebut secara mandiri masih rendah. Hal ini memperlihatkan kebutuhan untuk solusi alternatif agar alat bantu medis dapat diakses oleh masyarakat yang membutuhkan tanpa harus membeli dengan harga tinggi.

Dalam konteks tersebut, pengembangan sistem peminjaman alat bantu medis menjadi solusi inovatif yang dapat membantu mengatasi masalah keterbatasan akses ini. Sistem ini akan memungkinkan masyarakat memperoleh alat bantu medis secara temporal melalui peminjaman, terutama bagi mereka yang tidak mampu membeli alat tersebut dan ketika stok alat di rumah sakit habis. Dengan adanya sistem ini, diharapkan dapat meningkatkan ketersediaan alat bantu medis bagi masyarakat yang membutuhkan dan mengurangi risiko komplikasi kesehatan yang dapat timbul akibat tidak tersedianya alat bantu yang penting.

Berdasarkan kondisi tersebut, aplikasi *MedisLink* dikembangkan sebagai sistem informasi berbasis *web* yang menghubungkan peminjaman dan donasi

alat medis. Sistem ini mengintegrasikan teknologi modern seperti *React.js* di sisi depan, *Golang (Fiber)* di sisi *backend*, dan *MongoDB* sebagai basis data non-relasional. *MedisLink* dirancang untuk memberikan pengalaman pengguna yang transparan, efisien, dan terpercaya dalam mengakses alat bantu medis.

0.2 Identifikasi Masalah

Meskipun kebutuhan alat bantu medis sangat tinggi di masyarakat, sistem peminjaman alat medis yang terorganisir dan mudah diakses masih terbatas. Sebagian besar fasilitas kesehatan belum memiliki sistem yang dapat memfasilitasi peminjaman alat medis kepada masyarakat umum secara efisien.

Berdasarkan kondisi tersebut, kebutuhan yang muncul adalah:

1. Kebutuhan untuk memahami sekaligus mempraktikkan implementasi sistem informasi berbasis *web* untuk sektor kesehatan melalui proyek nyata.
2. Kebutuhan akan platform peminjaman alat medis yang mudah diakses dan sederhana, tanpa kompleksitas berlebih.
3. Kebutuhan akan sebuah media inovasi teknologi di bidang kesehatan yang tetap memiliki potensi digunakan oleh publik.

0.3 Tujuan

Tujuan utama dari pengembangan sistem *MedisLink* adalah membangun platform peminjaman alat bantu medis yang sederhana, fungsional, dan dapat menjadi dasar pengembangan lebih lanjut. Secara khusus, tujuan proyek ini adalah:

1. Merancang dan mengimplementasikan sistem peminjaman alat medis berbasis *web* sebagai penerapan teknologi informasi di bidang kesehatan.
2. Menyediakan alternatif sistem peminjaman alat medis yang mudah diakses, fleksibel, dan mudah dikembangkan sesuai kebutuhan.
3. Membangun pondasi produk digital yang dapat diekspansi menjadi sistem yang lebih kompleks di masa depan sekaligus menjadi sarana pembelajaran dan inovasi di bidang teknologi kesehatan.

0.4 Ruang Lingkup

Ruang lingkup pengembangan ini difokuskan pada perancangan, pembangunan, dan evaluasi prototipe *MedisLink*, yaitu sistem informasi *web* untuk peminjaman alat bantu medis. Lingkup ini mencakup proses analisis kebutuhan, perancangan arsitektur dan model data, implementasi modul inti, serta pengujian fungsional dalam lingkungan pengembangan.

0.4.1 Cakupan Fungsional

Penelitian ini mencakup pengembangan fitur inti sebagai berikut:

1. Manajemen Pengguna

Meliputi registrasi, autentikasi, dan otorisasi pengguna menggunakan dua metode:

- *JSON Web Token (JWT)* untuk autentikasi berbasis *username/password*.
- Sistem verifikasi identitas untuk memastikan keamanan peminjaman alat medis.

2. Pengelolaan Inventaris Alat Medis

Pencatatan alat medis yang tersedia (nama, kondisi, ketersediaan), penyimpanan data inventaris, serta penyajian informasi ketersediaan alat kepada pengguna.

3. Sistem Peminjaman

Proses peminjaman alat medis dengan pencatatan identitas peminjam, durasi peminjaman, dan status pengembalian alat.

4. Notifikasi Otomatis

Sistem notifikasi pengembalian alat otomatis menggunakan mekanisme *Cron Job* dengan interval waktu harian untuk memastikan waktu distribusi inventaris yang tepat.

5. Antarmuka Pengguna Web

Halaman publik untuk menampilkan katalog alat medis dan melakukan proses peminjaman, serta halaman *admin* untuk pengelolaan inventaris.

0.4.2 Cakupan Teknis

Secara teknis, penelitian ini mencakup:

1. Arsitektur Aplikasi

Pembangunan sistem dengan arsitektur terpisah menggunakan *React.js* di sisi *frontend*, *Golang (Fiber)* di sisi *backend*, dan *MongoDB* sebagai basis data non-relasional.

2. Keamanan Dasar Sistem

Meliputi:

- Implementasi *JWT* untuk autentikasi pengguna
- Validasi input dan sanitasi untuk keamanan data
- Penanganan akses *endpoint* privat berdasarkan *token*
- Verifikasi identitas untuk peminjaman alat medis

3. Perancangan dan Pemodelan

Meliputi *use case diagram*, *flowchart* proses (alur peminjaman, autentikasi, inventaris), dan rancangan model data (User, Equipment, Transaction).

4. Integrasi Modul Internal

Modul koneksi basis data, *middleware* verifikasi *token*, utilitas *hashing*, serta pengelolaan data melalui *schema* User dan Equipment.

0.4.3 Cakupan Pengujian

Pengujian dilakukan meliputi:

1. Uji fungsional terhadap *endpoint* inti seperti autentikasi pengguna, pencatatan/pengambilan data peminjaman, dan pemuatan data inventaris.
2. Uji integritas sederhana untuk memastikan alur peminjaman hingga pengembalian berjalan *end-to-end* menggunakan data simulasi.

0.4.4 Batasan Penelitian

Penelitian ini dibatasi pada aspek-aspek berikut:

1. Fokus Pada Alur Peminjaman Dasar

Sistem hanya mencakup proses peminjaman sederhana tanpa fitur pendukung seperti manajemen *maintenance* alat, penjadwalan peminjaman lanjutan, atau sistem rating.

2. Lingkup Pengembangan Berskala Proyek

Optimasi performa produksi, skalabilitas tinggi, dan *security hardening* tingkat lanjut tidak menjadi fokus utama.

3. Integrasi Sistem Kesehatan

Sistem tidak terintegrasi dengan sistem informasi rumah sakit atau fasilitas kesehatan lainnya; beroperasi sebagai sistem mandiri untuk tujuan pembelajaran dan pengujian konsep.

0.4.5 Luaran Penelitian

Luaran yang dihasilkan meliputi:

1. Prototipe aplikasi *web MedisLink* yang dapat dijalankan pada lingkungan pengembangan.
2. Dokumen desain arsitektur, *use case*, *flowchart*, dan *schema* model data.
3. Spesifikasi *endpoint API* inti, termasuk manajemen inventaris dan sistem notifikasi.
4. Hasil pengujian fungsional serta evaluasi ketercapaian kebutuhan.

0.5 Sistematika Penulisan

Sistematika penulisan analisis ini disusun dalam 5 bab dan bagian akhir terdapat daftar pustaka dan lampiran. Di mana pada setiap bab tersebut akan dibagi lagi menjadi sub-bab yang akan dibahas secara terperinci. Berikut merupakan sistematika penulisan dan keterangan singkatnya:

1. Bab I Pendahuluan

Berisi latar belakang, identifikasi masalah, tujuan, ruang lingkup, batasan, dan sistematika penulisan. Bab ini juga menjelaskan alasan pemilihan teknologi seperti *React.js*, *Golang (Fiber)*, *MongoDB*, dan *JWT*.

2. Bab II Landasan Teori

Menguraikan teori dan konsep pendukung seperti arsitektur *web* modern, sistem informasi kesehatan, autentikasi *JWT*, *API REST*, basis data *NoSQL*, serta *UML*.

3. Bab III Metode Penelitian

Menjelaskan metode analisis kebutuhan, perancangan, pemodelan, dan pengujian untuk sistem peminjaman alat medis.

4. Bab IV Hasil dan Pembahasan

Menyajikan hasil implementasi prototipe, struktur proyek, model data, *endpoint*, antarmuka pengguna, hasil pengujian, serta evaluasi sesuai ruang lingkup.

5. Bab V Kesimpulan dan Saran

Berisi kesimpulan akhir dan rekomendasi pengembangan lebih lanjut sistem *MedisLink*.

BAB II

LANDASAN TEORI

0.6 Platform Donasi Digital

Platform donasi digital merupakan pengembangan dari teknologi platform berbasis internet yang memfasilitasi interaksi antara berbagai pihak untuk tujuan sosial dan filantropi. Platform digital didefinisikan sebagai seperangkat komponen teknologi yang menyediakan fungsi inti bagi suatu sistem dan menjadi fondasi bagi pengembangan layanan pelengkap di atasnya (?, ?, ?). Secara konseptual, platform ini beroperasi sebagai *two-sided market* yang mempertemukan kelompok pengguna berbeda namun saling bergantung, seperti donatur dan penerima manfaat, di mana nilai platform tercipta dari interaksi antar pengguna tersebut (?, ?).

Dalam konteks filantropi di Indonesia, platform digital digunakan sebagai alternatif lembaga amil konvensional dengan menawarkan kemudahan akses, transparansi, dan kecepatan distribusi dana untuk zakat, infaq, sedekah, dan wakaf (?, ?, ?). Perkembangan ini sejalan dengan meningkatnya kepercayaan masyarakat terhadap sistem donasi daring serta pergeseran perilaku filantropi ke media digital (?, ?).

Dalam lingkup penggalangan dana massal, platform donation-based crowdfunding didefinisikan secara spesifik sebagai perantara teknologi yang digunakan oleh penggalang dana untuk mencocokkan donasi dengan tujuan para donatur. Tujuan yang dipertemukan dalam platform ini tidak bersifat moneter, melainkan pemenuhan kebutuhan psikologis donatur (? , ?). Dalam ekosistem ini, penyedia situs web berperan menyediakan layanan web dan membangun sistem yang mendukung pemilik proyek untuk mempresentasikan kampanye mereka kepada calon pendukung guna menggalang donasi (? , ?, ?).

0.7 Arsitektur Aplikasi Web Modern

Bagian ini membahas konsep dasar arsitektur aplikasi web modern yang menjadi landasan dalam memahami cara kerja sistem berbasis web (? , ?). Pemahaman mengenai pola komunikasi serta pembagian lapisan dalam aplikasi diperlukan untuk menjelaskan bagaimana komponen penyusun sistem saling berinteraksi dan menjalankan fungsinya. Oleh karena itu, pembahasan berikut difokuskan pada model client-server dan arsitektur three-tier/N-tier sebagai struktur arsitektural yang umum digunakan dalam pengembangan aplikasi web masa kini.

0.7.1 Arsitektur Three-Tier/N-Tier

Arsitektur three-tier (tiga lapisan) atau N-tier merupakan sebuah model arsitektur perangkat lunak yang membagi fungsionalitas aplikasi menjadi tiga lapisan logis dan fisik yang berbeda untuk meningkatkan skalabilitas dan keandalan sistem (? , ?, ?). Tiga lapisan utama tersebut terdiri dari: lapisan presentasi (*presentation tier*), yang berinteraksi langsung dengan pengguna; lapisan aplikasi (*application tier* atau *business logic tier*), yang menangani pemrosesan data dan logika bisnis inti; dan lapisan data (*data tier*), yang bertanggung jawab atas penyimpanan dan manajemen basis data (? , ?). Pemisahan fungsionalitas ini memungkinkan setiap lapisan dikelola dan dikembangkan secara *independent*, menjadi arsitektur ini pilihan yang efektif untuk sistem yang memerlukan ketersediaan tinggi (*high availability*), seperti pada kasus penerapan LMS Moodle (? , ?).

0.7.2 Konsep Client-Server

Client-server merupakan model perangkat lunak yang memungkinkan sumber daya dan permintaan layanan dipenuhi melalui jaringan, di mana komputer klien akan meminta layanan dan server akan menerima, memproses, serta memberikan respons yang sesuai (?, ?, ?). Komunikasi antara klien dan server difasilitasi melalui protokol standar seperti HTTP, FTP, dan SMTP (?, ?).

0.8 REST API dan Protokol HTTP

REST API dipahami sebagai pendekatan arsitektur web service yang memanfaatkan prinsip *Representational State Transfer* (?, ?, ?). REST menekankan penggunaan URI standar untuk mengidentifikasi resource, komunikasi stateless, serta penerapan uniform interface. HTTP berperan sebagai protokol utama yang digunakan sebagai standar komunikasi dengan metode GET, POST, PUT, dan DELETE (?, ?, ?).

0.9 Autentikasi dan Otorisasi

Bagian ini membahas konsep dasar autentikasi dan otorisasi yang menjadi fondasi penting dalam pengamanan aplikasi berbasis web. Mekanisme pengenalan identitas pengguna dan pemberian hak akses harus dirancang secara tepat agar interaksi antar sistem tetap aman, terukur, dan sesuai dengan tingkat kewenangan yang dibutuhkan. Oleh karena itu, pembahasan berikut mencakup OAuth 2.0 sebagai protokol delegasi akses, OpenID Connect sebagai lapisan identitas, JSON Web Token (JWT) sebagai format token yang umum digunakan, skema Bearer Token yang banyak diadopsi dalam komunikasi API, serta prinsip-prinsip keamanan API yang memastikan perlindungan terhadap ancaman dan penyalahgunaan akses.

0.9.1 OAuth 2.0

OAuth 2.0 didefinisikan sebagai framework otorisasi yang memungkinkan suatu aplikasi memperoleh akses terbatas ke resource yang dilindungi tanpa harus menyimpan kredensial pengguna secara langsung (?, ?). OAuth 2.0 menyediakan seperangkat authorization server (?, ?). Framework ini dirancang untuk mendukung berbagai konteks-mulai dari aplikasi web, single-page apps, hingga

aplikasi mobile-dengan cara memberikan fleksibilitas pada mekanisme autentikasi dan otorisasi yang aman di antara berbagai jenis klien(?, ?).

0.9.2 OpenID Connect

OpenID Connect (OIDC) merupakan sebuah protokol yang mapan yang digunakan secara luas dalam manajemen identitas terfederasi (federated identity management). Protokol ini berfungsi sebagai dasar bagi otentikasi dan sistem Masuk Tunggal (Single Sign-On atau SSO), yang memungkinkan klien untuk memverifikasi identitas pengguna akhir berdasarkan otentikasi yang dilakukan oleh Penyedia Identitas (Identity Provider)(?, ?, ?). Dibangun di atas kerangka kerja otorisasi OAuth 2.0, kegunaan OIDC meluas hingga ke infrastruktur kompleks, seperti memfasilitasi akses Secure Shell (SSH) pada pengaturan terfederasi dengan menggunakan token akses OIDC untuk otentikasi pengguna pada server jarak jauh(?, ?).

0.9.3 JWT

JSON Web Token (JWT) merupakan sebuah standar terbuka yang didasarkan pada RFC 7519, yang digunakan secara luas sebagai mekanisme standar untuk otentikasi dan otorisasi pengguna pada layanan web. Standar ini tidak hanya populer untuk mengamankan transmisi data dan otentikasi pada RESTful API, tetapi juga dapat diperluas untuk meningkatkan keamanan dengan menyimpan informasi historis perilaku pengguna, seperti konsistensi alamat IP dan jenis user agent (?, ?, ?). Sementara itu, JWT secara fundamental adalah format token yang memfasilitasi transmisi data yang ringkas dan aman antara pihak-pihak yang berkepentingan sebagai objek JSON, yang menjadikannya mekanisme otentikasi yang penting dalam implementasi berbagai aplikasi modern (?, ?).

0.9.4 Skema Bearer Token

Skema Bearer Token merupakan mekanisme autentikasi pada OAuth 2.0 di mana klien cukup menyertakan token pada header (`Authorization: Bearer {token}`) untuk memperoleh akses ke resource yang dilindungi (?, ?). Karena token ini bersifat bearer, siapa pun yang memiliki dapat menggunakannya tanpa verifikasi tambahan, sehingga membuat keamanan transport menjadi aspek kritis. Penelitian terbaru menyoroti bahwa risiko pencurian token dapat diminimalkan melalui penggunaan kalal terenskripsi, pembatasan masa hidup

token, serta validasi ketat pada sisi server (? , ?). Selain itu, praktik modern juga menekankan pentingnya menghindari pengiriman token melalui URL dan memastikan proses otorisasi mengikuti pedoman keamanan OAuth 2.0 (? , ?).

0.9.5 Keamanan API

Keamanan API merupakan aspek kritis karena API sering menjadi target serangan (? , ?). Banyak celah keamanan muncul akibat pengelolaan asset API yang lemah, API lama yang tidak terinventarisasi, serta kerentanan pada alur data dan logika bisnis (? , ?). Selain itu, meningkatnya kompleksitas arsitektur RESTful dan GraphQL memperluas permukaan serangan, termasuk risiko seperti information leakage, unauthorized access, dan eksloitasi validasi input yang tidak memadai (? , ?)

Untuk mengatasi ancaman tersebut, mekanisme keamanan API membutuhkan pendekatan berlapis yang mencakup autentikasi kuas berbasis OAuth/JWT, penggunaan HTTPS/TLS untuk mengamankan transmisi data, serta manajemen hak akses yang detail guna mencegah penyalahgunaan kredensial (? , ?). Pentingnya teknik seperti asset discovery, traffic auditing, dan analisis alur data untuk mengidentifikasi API tersembunyi dan aktivitas mencurigakan (? , ?). Di samping itu, penggunaan API Gateway dapat membantu menerapkan pembatasan trafik, filtrasi permintaan, dan perlindungan terhadap serangan seperti DDoS, sehingga API tetap terawasi dan terlindungi secara konsisten.

0.10 Database NoSQL (MongoDB)

Basis data dokumen NoSQL (*Not Only SQL*) muncul sebagai alternatif yang signifikan terhadap basis data relasional tradisional yang sering memiliki batasan ketat pada struktur data dan relasi, sehingga kurang efisien untuk menangani volume data yang sangat besar (*huge database*) (? , ?). NoSQL document database mengatasi masalah ini dengan menyediakan kemampuan untuk menyimpan dan mengelola data dalam format dokumen, sehingga dapat menampung data yang tidak terstruktur, semi-struktur, maupun terstruktur (? , ?). Keunggulan utama NoSQL, khususnya jenis berorientasi dokumen seperti MongoDB, terletak pada fleksibilitas dan skalabilitas horizontal yang tinggi, menjadikannya pilihan esensial ketika skema data yang dinamis tidak sesuai dengan kebutuhan basis data realisonal (? , ?).

Secara opsional, basis data dokumen NoSQL menyimpan dalam bentuk dokumen. Meskipun memiliki perbedaan dalam beberapa aspek, MongoDB,

Couchbase, dan CouchDB adalah contoh utama dari basis data dokumen yang terkenal (?, ?). Sebagai contoh MongoDB merupakan basis data berorientasi dokumen, crossplatform, yang menawarkan kinjera tinggi, ketersediaan tinggi, dan skalabilitas yang sederhana. Basis data ini menggunakan MongoDB Query Language (MQL) yang dirancang untuk kemudahan penggunaan oleh pengembang (?, ?). Oleh karena itu, basis data dokumen NoSQL menjadi solusi penting untuk aplikasi padat data, memastikan penyimpanan big data dan kinerja kueri yang baik.

0.11 Unified Modeling Language (UML)

Unified Modeling Language (UML) didefinisikan sebagai sebuah bahasa pemodelan standar yang digunakan untuk merancang dan mendokumentasikan sistem berorientasi objek. Sebagai bahasa standar, UML menyediakan seperangkat notasi grafis yang komprehensif untuk memvisualisasikan, memspesifikasi, membangun, dan mendokumentasikan artefak dalam sistem perangkat. Tujuan utama penggunaan UML adalah untuk memperjelas model yang tidak konsisten dan mengurangi ambiguitas selama proses pengembangan perangkat lunak (?, ?). UML membantu memvisualkan, menspesifikasi, dan mendokumentasikan desain sistem secara grafis (?, ?).

Dengan menggunakan diagram-diagram yang berbeda, seperti Use Case Diagram dan Activity Diagram, UML membantu pengembang dalam memodelkan interaksi, struktur, dan perilaku sistem (?, ?). Penerapan UML sangat krusial dalam siklus hidup pengembangan sistem (*System Development Life Cycle* atau SDLC) karena membantu memastikan konsistensi model dan mempermudah komunikasi antara pihak-pihak yang terlibat dalam proyek (?, ?).

0.11.1 Use Case Diagram

Use Case adalah suatu diagram fundamental yang umum diajarkan dalam ilmu komputer dan rekayasa perangkat lunak. Diagram ini berfungsi sebagai representasi visual dari fungsionalitas sistem dari sudut pandang pengguna. Meskipun definisinya tampak sederhana, penilaian terhadap diagram use case sering kali menjadi hambatan dalam proses pembelajaran, terutama karena dua masalah utama: masalah interpersonal (tidak adanya standar penilaian di antara para pengajar) (?, ?) dan masalah intrapersonal (inkonsistensi seorang pengajar saat menilai banyak diagram) (?, ?, ?, ?).

0.11.2 Activity Diagram

Activity Diagram adalah salah satu diagram perilaku yang tersedia dalam Unified Model Language (UML) yang digunakan untuk memodelkan alur kontrol dan alur data dalam suatu sistem (?, ?). Diagram ini secara visual merepresentasikan Langkah-langkah, keputusan, dan urutan tindakan yang diperlukan untuk menyelesaikan suatu proses atau kegiatan bisnis tertentu (?, ?). Dalam konteks pemodelan sistem, Activity Diagram sangat berguna untuk memvisualisasikan bagaimana berbagai kegiatan saling terkait dan bergantung satu sama lain (?, ?, ?).

0.11.3 Sequence Diagram

Sequence Diagram adalah diagram UML yang paling umum kedua, digunakan untuk merepresentasikan interaksi objek dan pertukaran pesan antar objek tersebut seiring berjalannya waktu (?, ?). Diagram ini secara visual menunjukkan bagaimana peristiwa atau aktivitas yang ada dalam sebuah use case dipetakan menjadi operasi-operasi dari kelas objek yang ada pada Class Diagram (?, ?).

0.11.4 Class Diagram

Class Diagram merupakan salah satu diagram Unified Modeling Language (UML) yang paling umum digunakan dalam Pendidikan dan pengembangan perangkat lunak berorientasi objek (?, ?). Fungsi utama dari Class Diagram adalah untuk merepresentasikan kelas-kelas dalam sistem perangkat lunak dan hubungan yang terjalin antar kelas-kelas tersebut (?, ?).

BAB III

METODE PENELITIAN

0.12 Metode Pengembangan Sistem

Pengembangan platform Nyumbangin menggunakan pendekatan Agile, yang menekankan proses pengembangan sistem secara bertahap, adaptif, dan berulang. Pendekatan Agile dipilih karena sesuai dengan karakteristik proyek berskala kecil hingga menengah, serta memungkinkan penyesuaian fitur berdasarkan hasil evaluasi pada setiap tahap pengembangan.

Pendekatan ini memungkinkan sistem dikembangkan secara inkremental, di mana setiap iterasi menghasilkan fungsionalitas yang dapat diuji dan dievaluasi sebelum melanjutkan ke iterasi berikutnya. Dengan demikian, risiko kesalahan desain dan implementasi dapat diminimalkan sejak tahap awal.

0.12.1 Konsep Agile Development

Agile Development merupakan pendekatan pengembangan perangkat lunak yang berfokus pada fleksibilitas, kolaborasi, dan kemampuan beradaptasi terhadap perubahan kebutuhan (? , ?). Berbeda dengan metode linear seperti waterfall, Agile memungkinkan perubahan kebutuhan terjadi selama proses pengembangan tanpa harus mengulang seluruh tahapan dari awal.

Dalam konteks pengembangan platform Nyumbangin, Agile digunakan sebagai kerangka kerja konseptual untuk mengelola proses pengembangan fitur secara bertahap, mulai dari analisis kebutuhan dasar, implementasi modul inti, hingga pengujian dan evaluasi sistem. Pendekatan ini mendukung pengembangan sistem yang responsif terhadap kebutuhan fungsional dan teknis yang berkembang selama proyek berlangsung.

0.12.2 Alur Iterasi Pengembangan

Proses pengembangan sistem dilakukan melalui beberapa siklus iterasi yang masing-masing mencakup tahapan berikut:

1. Perencanaan Iterasi

Penentuan fitur yang akan dikembangkan berdasarkan prioritas kebutuhan sistem.

2. Implementasi Fitur

Pengembangan modul atau fungsi tertentu sesuai dengan hasil perencanaan iterasi.

3. Pengujian Fungsional

Pengujian terhadap fitur yang telah dikembangkan untuk memastikan kesesuaian dengan kebutuhan.

4. Evaluasi dan Penyempurnaan

Evaluasi hasil iterasi dan perbaikan terhadap keterbatasan sebelum melanjutkan ke iterasi berikutnya.

Setiap iterasi menghasilkan peningkatan fungsional sistem yang dapat langsung diuji, sehingga kemajuan proyek dapat dipantau secara berkelanjutan.

0.12.3 Penerapan Agile pada Proyek Nyumbangin

Penerapan pendekatan Agile pada proyek Nyumbangin dilakukan dengan membagi pengembangan sistem ke dalam beberapa iterasi utama. Iterasi awal difokuskan pada pembangunan fitur inti, seperti autentikasi kreator, pencatatan donasi, dan integrasi sistem pembayaran. Iterasi berikutnya mencakup pengembangan fitur pendukung, seperti notifikasi real-time melalui *overlay*, *leaderboard* donatur, serta mekanisme *payout* bagi kreator.

Pada setiap iterasi, fitur yang telah diimplementasikan langsung diuji menggunakan scenario pengujian fungsional untuk memastikan alur donasi, pembayaran, dan pencairan dana berjalan dengan benar. Hasil pengujian digunakan sebagai dasar evaluasi untuk menentukan perbaikan atau pengembangan fitur pada iterasi selanjutnya.

Pendekatan ini memungkinkan pengembangan aplikasi dilakukan secara terstruktur namun tetap fleksibel, sehingga sistem dapat berkembang secara bertahap hingga memenuhi kebutuhan fungsional yang telah ditetapkan.

0.13 Analisis Kebutuhan

0.13.1 Sumber Kebutuhan

Analisis kebutuhan sistem dilakukan melalui tiga pendekatan utama. Pertama, observasi terhadap platform donasi digital untuk mengidentifikasi pola umum, seperti kebutuhan transparansi transaksi, tampilan notifikasi real-time, dan mekanisme payout yang akuntabel. Kedua, studi pustaka terkait autentikasi modern (OAuth 2.0, JWT), keamanan API, serta pola desain sistem web yang relevan dengan karakteristik aplikasi donasi. Ketiga, analisis artefak kode dan struktur API yang telah dikembangkan, termasuk model basis data, alur

donasi, integrasi Midtrans, serta skrip pemeliharaan yang digunakan untuk verifikasi dan pengarsipan data. Pendekatan ini memastikan kebutuhan sistem dirumuskan berdasarkan konteks teknis dan operasional yang aktual.

0.13.2 Kebutuhan Fungsional

Kebutuhan fungsional mendeskripsikan fitur yang wajib disediakan agar aplikasi donasi dapat berfungsi secara utuh. Fitur tersebut meliputi:

1. Autentikasi Pengguna menggunakan Google OAuth melalui NextAuth, serta validasi akses melalui JWT untuk endpoint sensitif.
2. Validasi Username Kreator, memastikan username unik dan dapat diverifikasi sebelum transaksi dilakukan.
3. Pengelolaan Donasi, mencakup pembuatan transaksi, integrasi Midtrans, pembaruan status melalui webhook, serta penyimpanan konten media share.
4. Notifikasi Real-Time, yang menampilkan donasi terbaru pada overlay kreator untuk keperluan siaran langsung.
5. Leaderboard, baik secara global maupun berdasarkan kreator, untuk menampilkan agregasi donasi.
6. Statistik Kreator, berupa ringkasan donasi berdasarkan periode.
7. Payout, mencakup permintaan penarikan dana, perhitungan biaya platform, dan persetujuan admin.
8. Pengelolaan Data Media Share dan Notifikasi, termasuk pengaturan masa berlaku (TTL) dan keterkaitan dengan transaksi donasi.

0.13.3 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional mencakup karakteristik kualitas sistem, yaitu:

- **Keamanan:** verifikasi token JWT, sanitasi input, pembatasan metode HTTP, dan pemisahan akses berdasarkan peran pengguna.
- **Performa:** optimasi query leaderboard melalui limit dan sorting.
- **Skalabilitas:** rencana pagination serta caching pada proses agregasi data.

- **Reliabilitas:** konsistensi penanganan kesalahan dengan kode status standar (401, 404, 500).
- **Integritas Data:** akurasi perhitungan saldo payout dan pemrosesan donasi berdasarkan statur valid (PAID).

0.14 Perancangan Sistem

0.14.1 Arsitektur Logis

Arsitektur logis sistem terdiri atas empat lapisan utama sebagai berikut:

1. Lapisan Antarmuka Pengguna (UI Layer)

Berisi halaman donasi, dashboard kreator, halaman overlay notifikasi, serta antarmuka admin.

2. Lapisan API (Application Layer)

Mengelola endpoint untuk donasi, *leaderboard*, *overlay*, autentikasi, *payout*, dan operasi admin melalui mekanisme *API Routes* di Next.js.

3. Lapisan Layanan Utilitas (Service Layer)

Melibuti modul koneksi database, pengelolaan token JWT, serta utilitas untuk validasi dan perhitungan internal.

4. Lapisan Data (Data Layer)

Terdiri atas model MongoDB seperti Donation, Creator, MediaShare, Payout, dan Notification.

0.14.2 Arsitektur Fisik

Arsitektur fisik sistem mengikuti pola aplikasi web modern sebagai berikut:

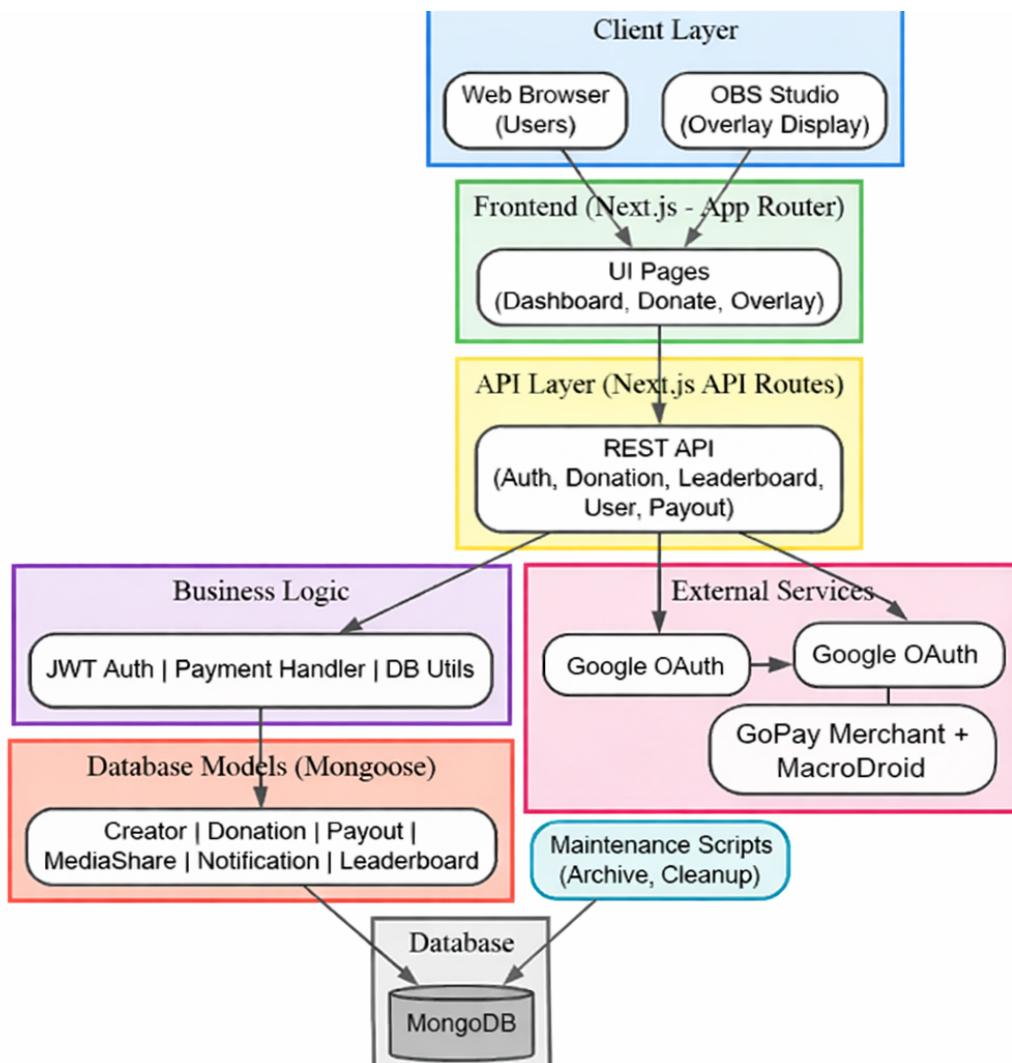
Browser / OBS Overlay → Next.js Runtime (Node.js) → MongoDB →
Layanan Eksternal (Google OAuth, Midtrans)

- Next.js menangani logika antarmuka pengguna dan API dalam satu platform.
- MongoDB digunakan sebagai basis data dokumen.
- Midtrans mengelola proses pembayaran melalui mekanisme *Snap Token* dan *webhook*.

- Overlay digunakan secara mandiri melalui OBS atau *iframe* untuk menampilkan notifikasi donasi.

0.14.3 Arsitektur Teknologi

Arsitektur teknologi sistem mencakup penggunaan Next.js sebagai framework utama yang menjalankan frontend dan backend melalui API Routes, Node.js sebagai runtime server-side, serta MongoDB sebagai basis data dokumen. Sistem autentikasi menggunakan Google OAuth 2.0 melalui NextAuth dan JWT untuk otorisasi endpoint privat. Mekanisme pembayaran dilakukan dengan Midtrans melalui Snap Token dan Webhook. Selain itu, aplikasi menyediakan overlay web real-time untuk integrasi dengan OBS sebagai tampilan notifikasi donasi. Kombinasi teknologi ini menghasilkan sistem yang modern, responsive, serta siap diintegrasikan dengan berbagai layanan eksternal.



Gambar 1: Arsitektur Teknologi

0.14.4 Modul Utama

Sistem dibagi ke dalam beberapa modul utama sebagai berikut:

1. **Authentication Module** (OAuth + JWT)
2. **Donation Module** (pembuatan transaksi, webhook, media share)
3. **Leaderboard Module** (global dan per kreator)
4. **Payout Module** (request, approval, perhitungan fee)
5. **Notification Module** (TTL, event donasi)
6. **Maintenance Module** (arsip donasi, integritas data)

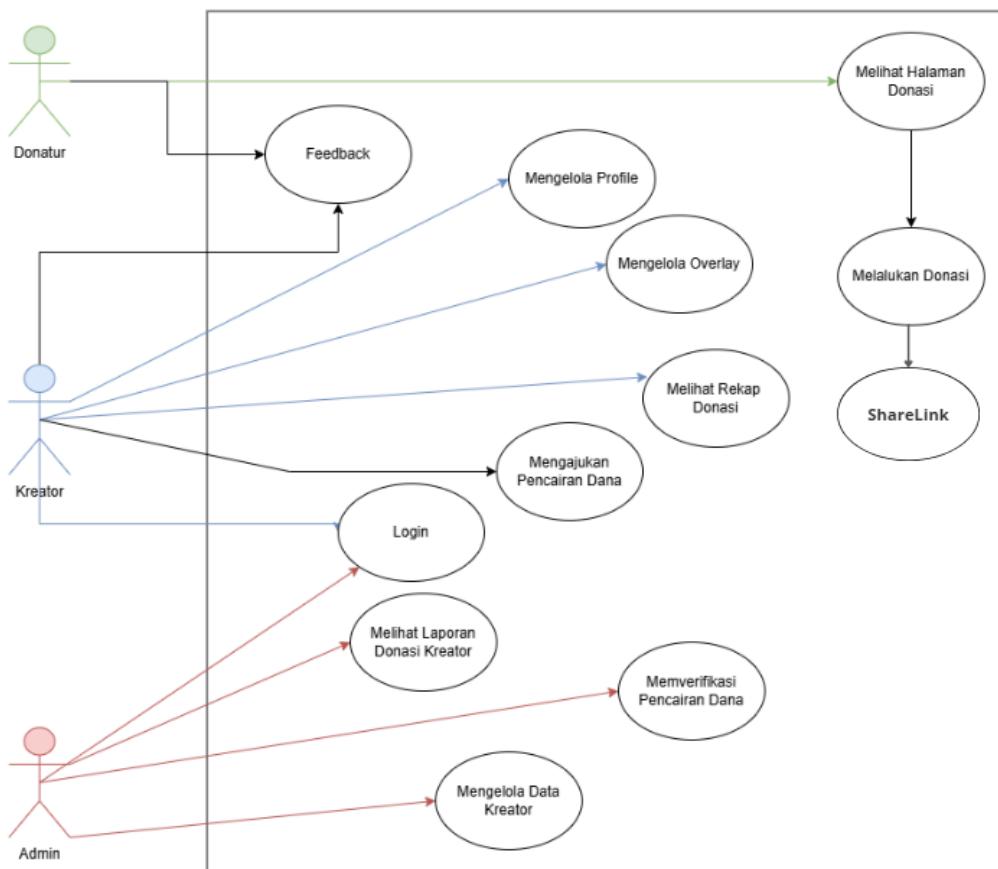
0.14.5 Strategi Desain

Strategi desain menerapkan pola penanganan *API* yang konsisten meliputi validasi metode HTTP, autentikasi, validasi input, eksekusi query database, dan pengembalian response dalam format JSON. Selain itu, prinsip pemisahan tanggung jawab diterapkan melalui pembagian endpoint berdasarkan peran dan fungsi. Pembatasan data seperti penggunaan limit dan sorting digunakan untuk menghindari over-fetching, terutama pada proses pengambilan data leaderboard dan statistik.

0.15 Pemodelan Sistem

Pemodelan sistem dilakukan untuk memberikan representasi visual dan konseptual dari fungsi, alur kerja, serta struktur data yang digunakan dalam aplikasi Nyumbangin ini. Pemodelan ini bertujuan memastikan bahwa kebutuhan fungsional dan non-fungsional yang telah diidentifikasi dapat diterjemahkan ke dalam desain sistem yang jelas, terstruktur, dan mudah diimplementasikan. Diagram-diagram pada bagian ini mencakup model proses, interaksi, dan entitas yang saling berhubungan, sehingga dapat memberikan gambaran menyeluruh mengenai cara sistem beroperasi secara end-to-end.

0.15.1 Use Case



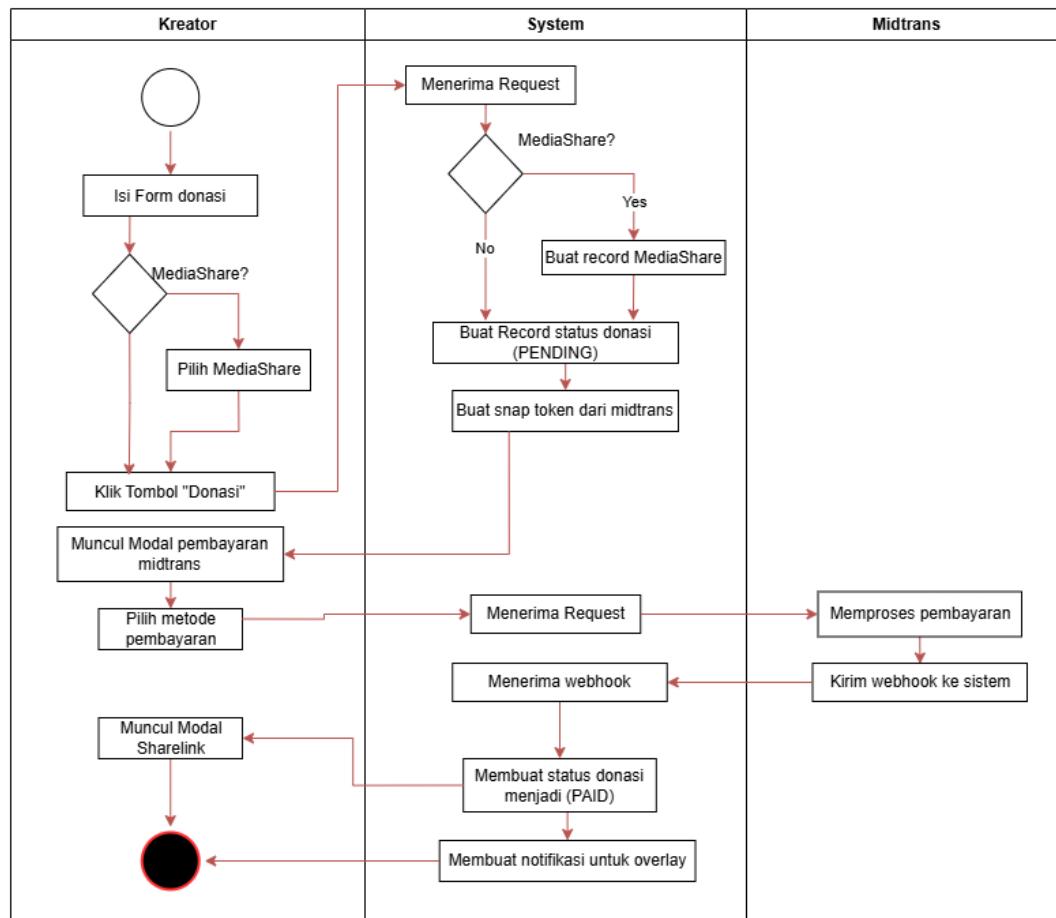
Gambar 2: Use Case Diagram

Use case menggambarkan interaksi antara aktor dengan sistem aplikasi Nyumbangin. Diagram ini bertujuan untuk menunjukkan fungsi-fungsi utama yang dapat dilakukan oleh masing-masing aktor sesuai dengan perannya.

Aktor yang terlibat dalam sistem ini yaitu Donatur, Kreator, dan Admin. Selain itu, sistem terintegrasi dengan layanan eksternal seperti Google OAuth untuk proses autentikasi dan Midtrans sebagai gateway pembayaran. Use case ini menggambarkan secara umum cakupan fungsi sistem tanpa menampilkan detail alur proses.

0.15.2 Activity Diagram

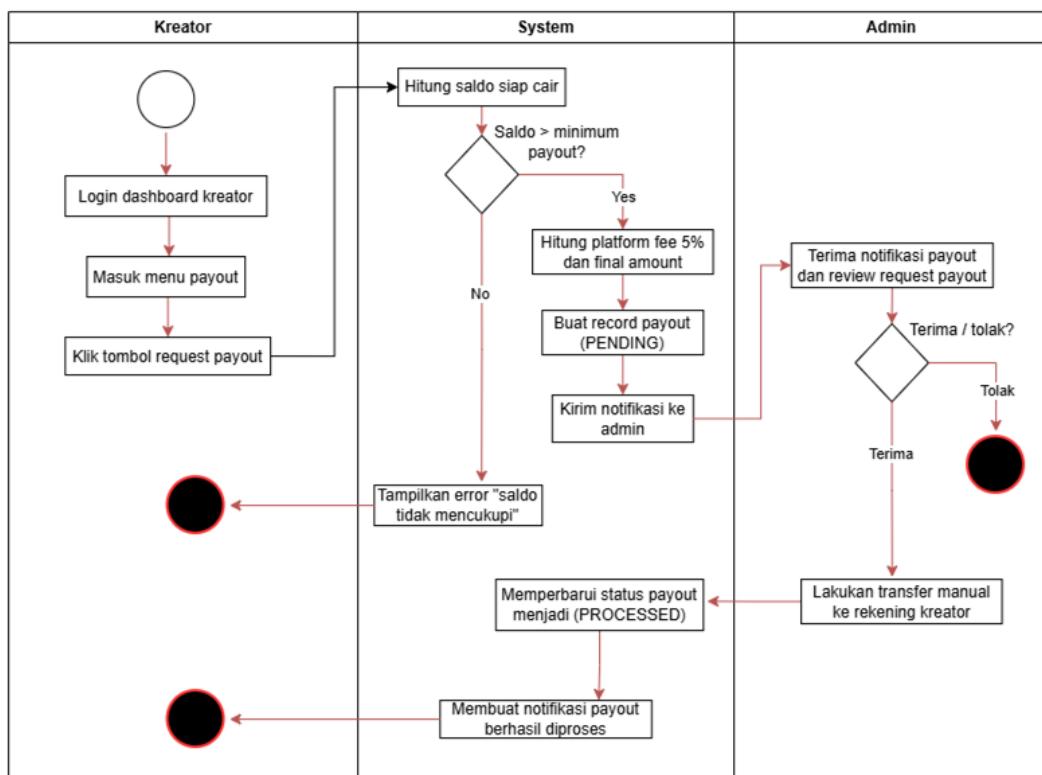
Activity Diagram proses Donasi



Gambar 3: Activity Diagram Donasi

Activity diagram proses donasi menggambarkan alur di mana Donor mengisi formulir donasi (dengan atau tanpa media share), kemudian sistem membuat record donasi dengan status PENDING, menghasilkan *snap token*, dan mengarahkan Donor ke Midtrans untuk proses pembayaran. Setelah Midtrans mengirimkan webhook, sistem memperbarui status donasi menjadi PAID dan membuat notifikasi donasi.

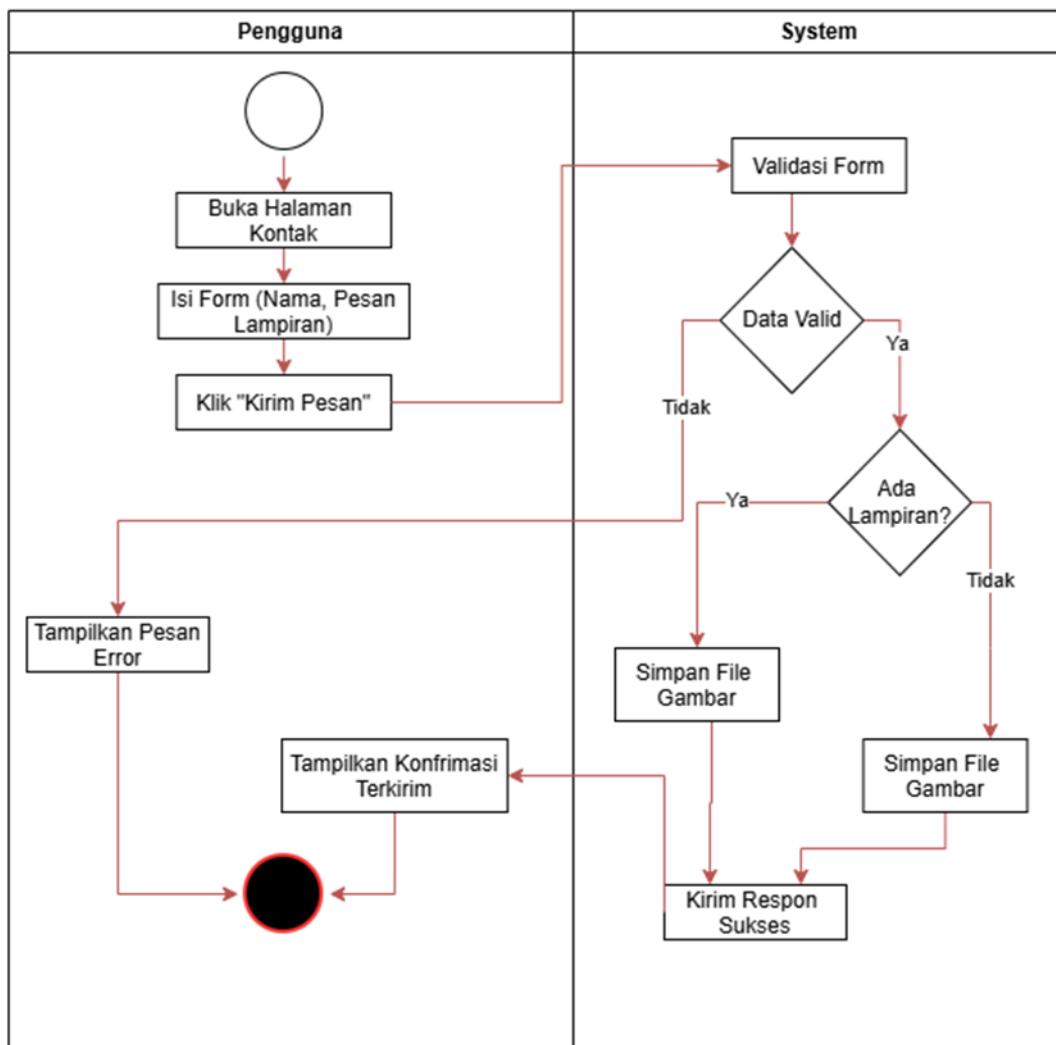
0.15.3 Activity Diagram Proses Payout



Gambar 4: Activity Diagram Payout

Activity diagram proses payout dimulai ketika Kreator mengajukan permintaan payout, di mana sistem melakukan pemeriksaan terhadap saldo minimal. Apabila saldo memenuhi syarat, sistem akan menghitung biaya layanan sebesar 5% dan mengubah status payout menjadi PENDING. Selanjutnya, Admin melakukan peninjauan dan transfer dana secara manual, setelah itu sistem memperbarui status payout menjadi PROCESSED.

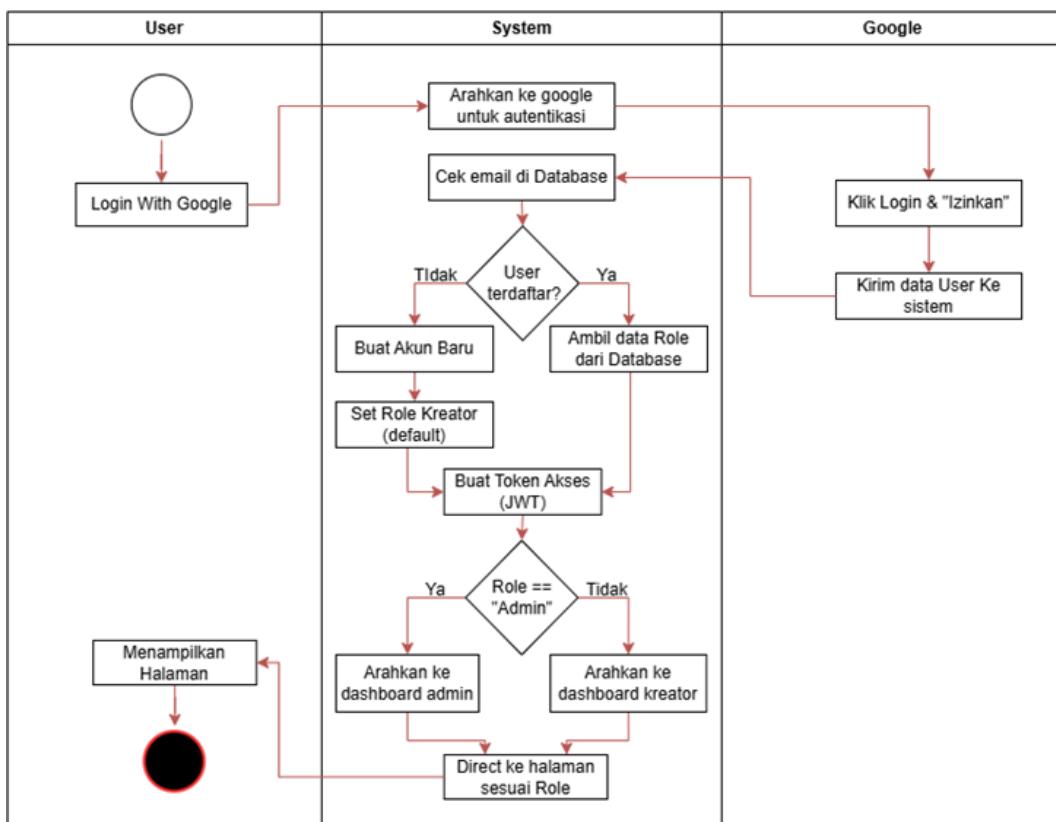
0.15.4 Activity Diagram Feedback



Gambar 5: Activity Diagram Feedback

Activity diagram feedback menggambarkan alur proses pengiriman feedback oleh pengguna. Proses dimulai dari pengisian formulir feedback, kemudian sistem melakukan validasi data. Apabila data dinyatakan valid, sistem akan menyimpan feedback dan menampilkan konfirmasi bahwa feedback berhasil dikirim. Sebaliknya, jika data tidak valid, sistem akan menampilkan pesan kesalahan.

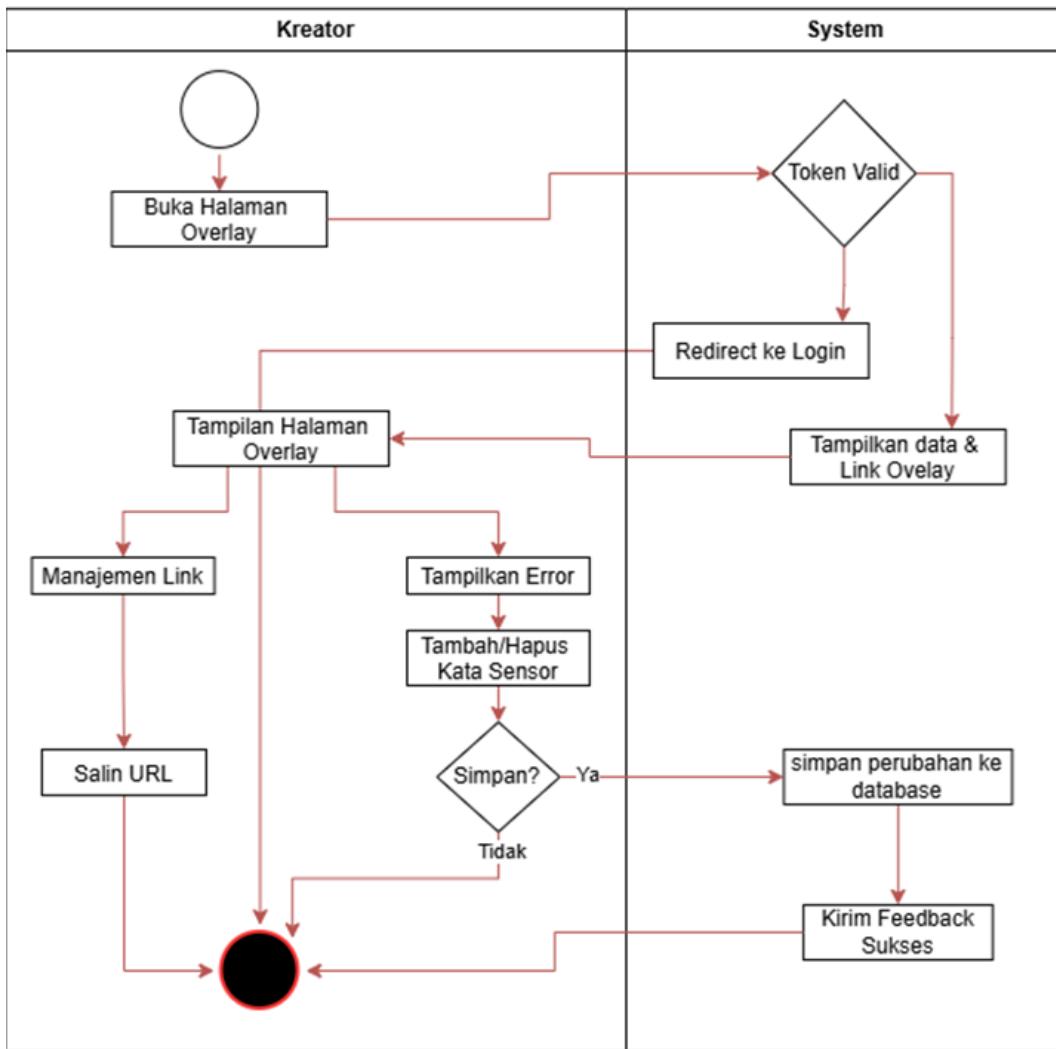
0.15.5 Activity Diagram Login



Gambar 6: Activity Diagram Login

Activity diagram login menggambarkan alur autentikasi pengguna menggunakan Google OAuth. Proses dimulai dari login, autentikasi ke Google, pengecekan data pengguna, pembuatan token akses, dan pengalihan ke halaman dashboard sesuai dengan peran pengguna.

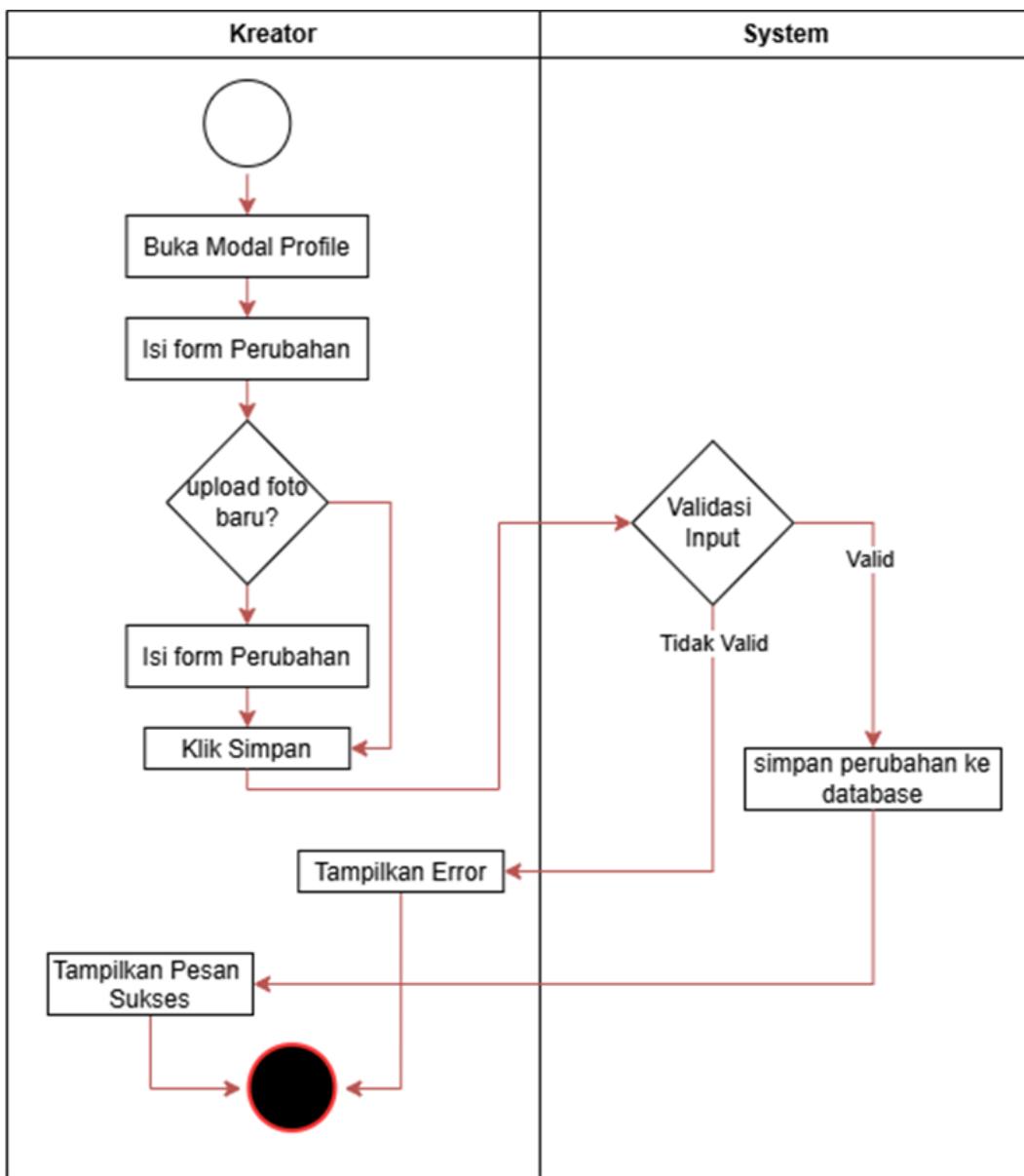
0.15.6 Activity Diagram Kelola Overlay



Gambar 7: Activity Diagram Kelola Overlay

Activity diagram kelola overlay menggambarkan alur pengelolaan overlay oleh kreator. Sistem melakukan pengecekan autentikasi sebelum menampilkan halaman overlay. Kreator dapat menambah, mengubah, atau menghapus data overlay. Sistem memvalidasi dan menyimpan perubahan, kemudian memberikan feedback keberhasilan kepada kreator.

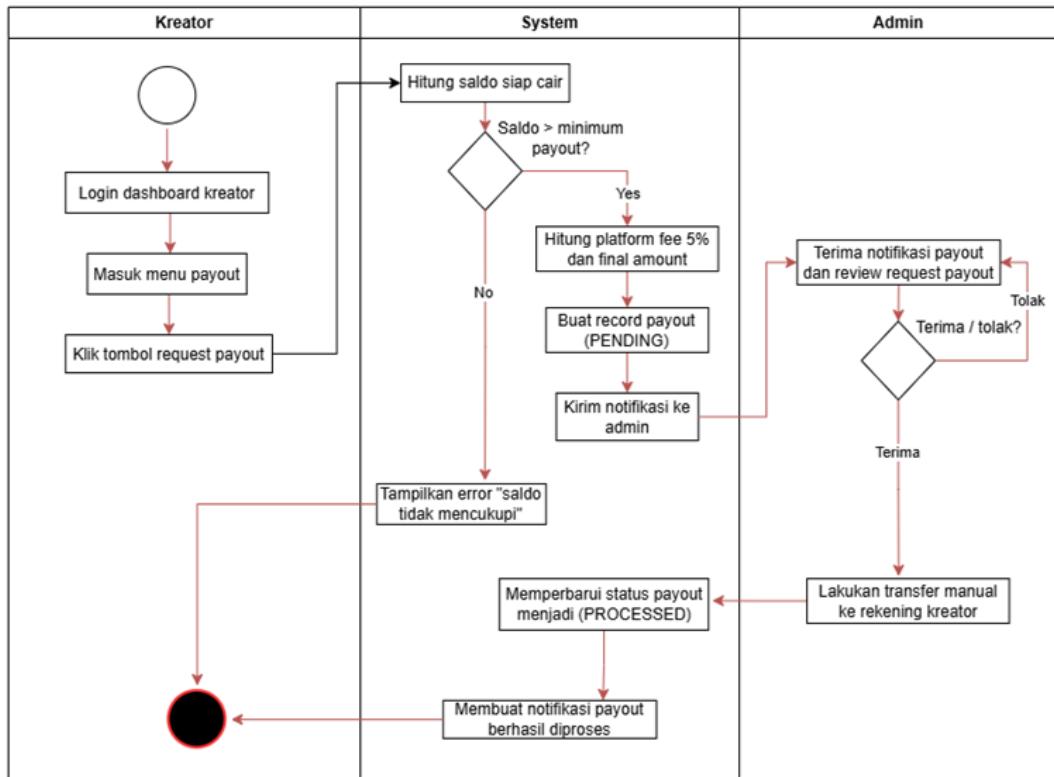
0.15.7 Activity Diagram Kelola Profil



Gambar 8: Activity Diagram Kelola Profil

Activity diagram kelola profil menggambarkan proses perubahan data profil oleh kreator. Kreator mengisi form perubahan data dan mengunggah foto profil jika diperlukan. Sistem melakukan validasi input dan menyimpan perubahan ke database. Jika proses berhasil, sistem menampilkan pesan sukses.

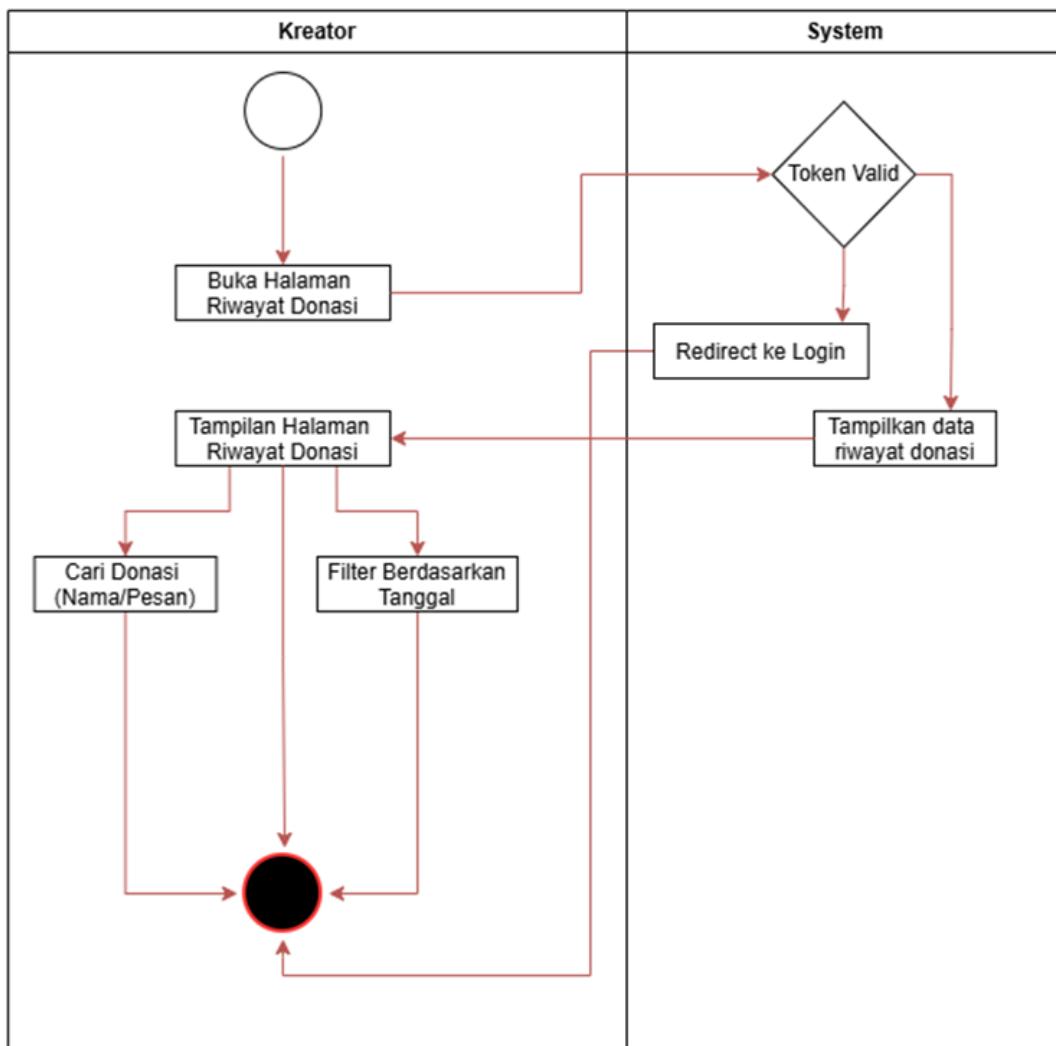
0.15.8 Activity Diagram Request Payout



Gambar 9: Activity Diagram Request Payout

Activity diagram request payout menggambarkan alur pencairan dana yang diajukan oleh kreator. Sistem memeriksa saldo dan minimum payout, menyimpan data payout, serta mengirimkan notifikasi ke admin. Admin melakukan proses review, dan sistem memperbarui status payout sesuai dengan keputusan admin.

0.15.9 Activity Diagram Riwayat Donasi

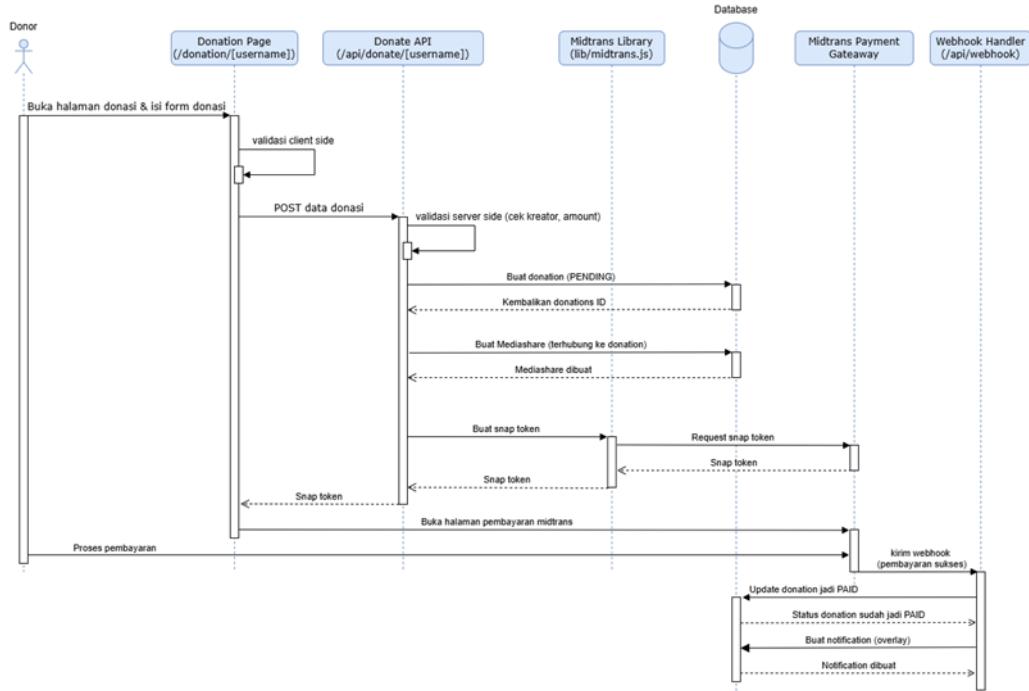


Gambar 10: Activity Diagram Riwayat Donasi

Activity diagram riwayat donasi menggambarkan proses penampilan data riwayat donasi. Kreator membuka halaman riwayat donasi, sistem menampilkan data donasi yang tersimpan, dan kreator dapat melakukan pencarian serta melakukan filter data berdasarkan kriteria tertentu.

0.15.10 Sequence Diagram

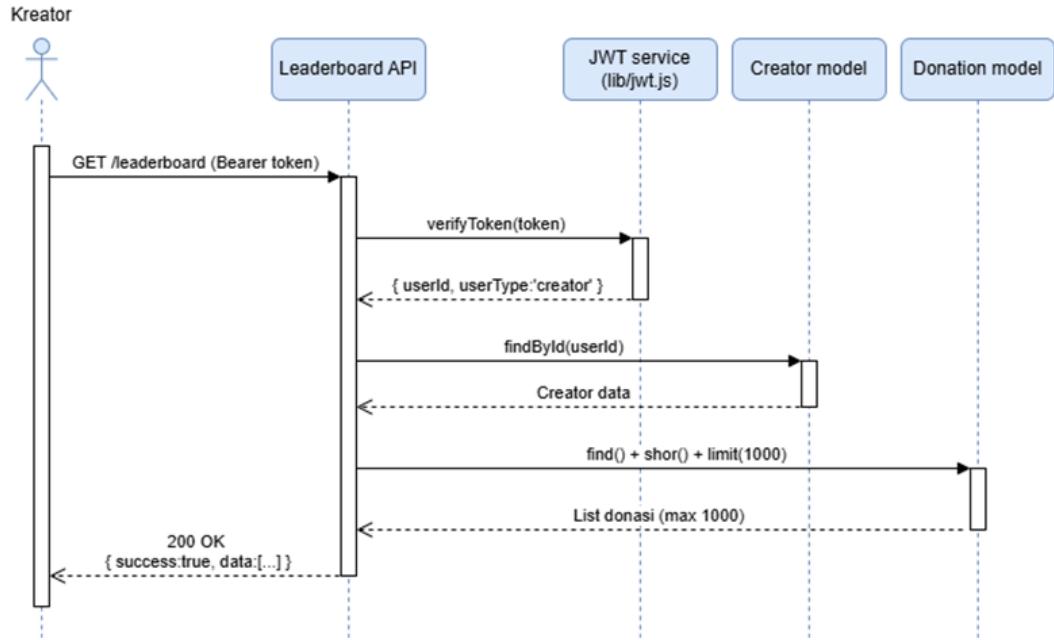
Sequence Diagram Donasi



Gambar 11: Sequence Diagram Donasi

Sequence diagram donasi menggambarkan alur proses donasi yang dimulai ketika donor mengisi formulir pada halaman donasi (*Donation Page*) yang kemudian divalidasi dan dikirim ke *Donate API*. Setelah dilakukan validasi di sisi server, data donasi disimpan ke dalam basis data, dan apabila terdapat *youtubeUrl*, sistem juga akan membuat data *MediaShare*. Selanjutnya, *API* meminta *Snap Token* ke Midtrans yang digunakan untuk membuka halaman pembayaran. Setelah donor menyelesaikan pembayaran dan transaksi dinyatakan berhasil, Midtrans mengirimkan webhook ke server sehingga status donasi diperbarui menjadi *PAID*, dan pada tahap akhir sistem membuat notifikasi overlay.

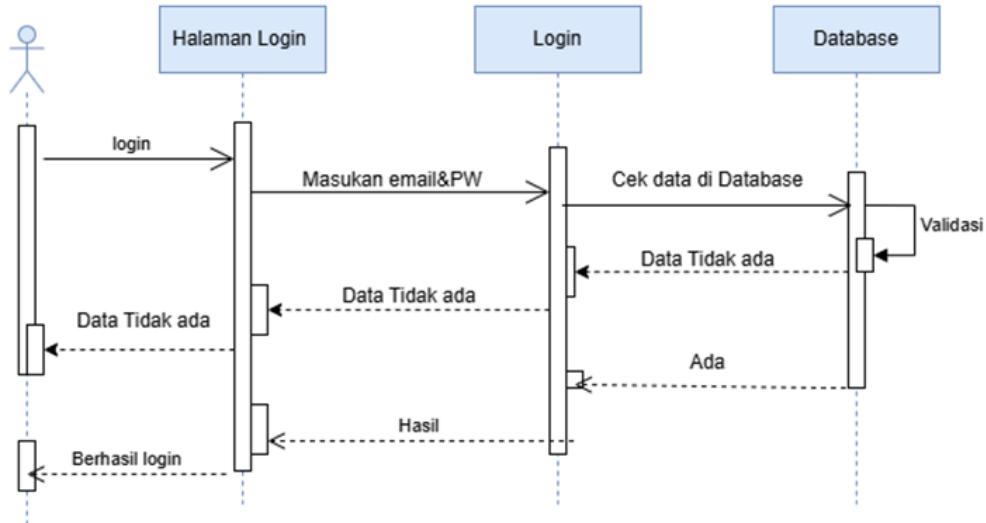
Sequence Diagram Leaderboard



Gambar 12: Sequence Diagram Leaderboard

Sequence diagram leaderboard menggambarkan alur pengambilan data *leaderboard*, di mana client (*dashboard kreator*) mengirimkan request ke *Leaderboard API* dengan menyertakan token. API melakukan pemeriksaan metode HTTP dan validasi token melalui layanan JWT, kemudian memastikan bahwa pengguna merupakan kreator dan data yang diminta tersedia. Selanjutnya, API mengambil data donasi terbaru dari basis data, memformat respons, dan mengirimkan hasilnya kembali ke client. Apabila terjadi kesalahan seperti token tidak valid, kreator tidak ditemukan, atau metode HTTP tidak sesuai, API akan mengembalikan kode error yang relevan.

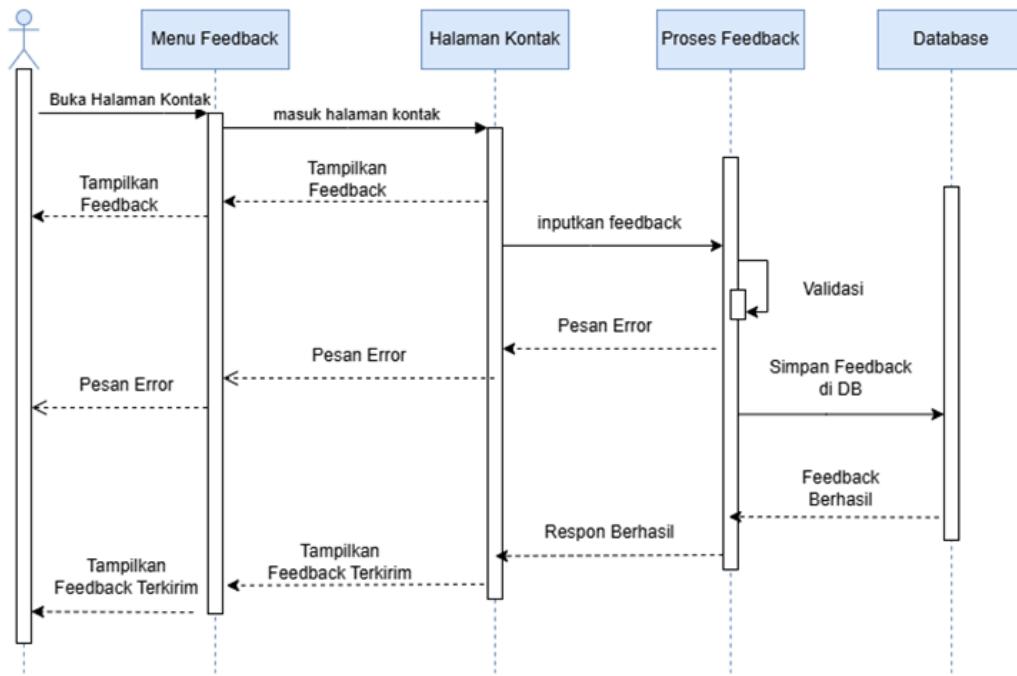
Sequence Diagram Login



Gambar 13: Sequence Diagram Login

Sequence diagram login menggambarkan urutan interaksi antara pengguna dan sistem dalam proses autentikasi menggunakan email dan kata sandi. Proses dimulai ketika pengguna membuka halaman login dan memasukkan data email serta kata sandi. Sistem kemudian melakukan pengecekan data ke basis data untuk memverifikasi kecocokan informasi yang dimasukkan. Apabila data tidak sesuai, sistem akan menampilkan pesan kesalahan kepada pengguna. Jika data sesuai, sistem memproses autentikasi dan mengarahkan pengguna ke halaman utama atau dashboard sesuai dengan hak akses yang dimiliki.

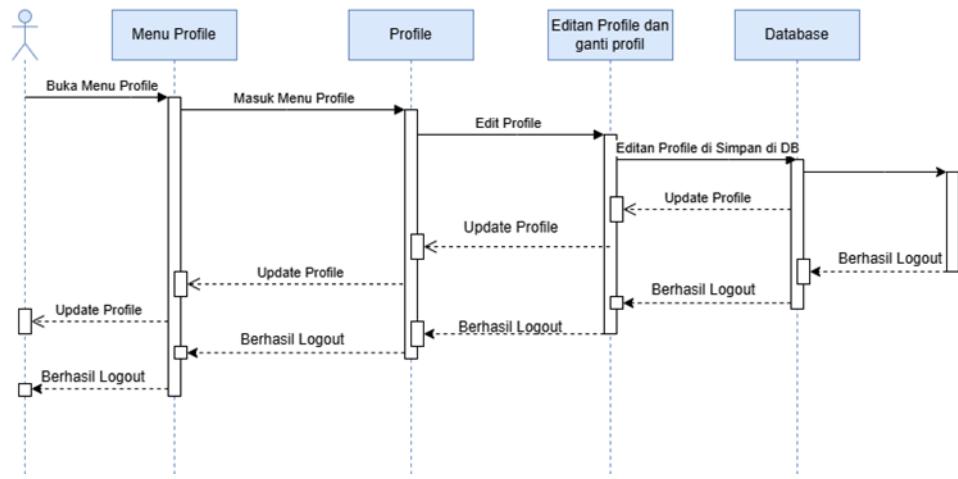
Sequence Diagram Feedback



Gambar 14: Sequence Diagram Feedback

Sequence diagram feedback menggambarkan alur pengiriman feedback dari pengguna ke sistem. Proses dimulai ketika pengguna membuka halaman kontak dan mengisi form feedback. Sistem melakukan validasi terhadap data yang dikirimkan. Jika data tidak valid, sistem akan mengirimkan pesan error. Jika data valid, sistem menyimpan feedback ke dalam basis data dan mengirimkan respons bahwa feedback berhasil dikirim.

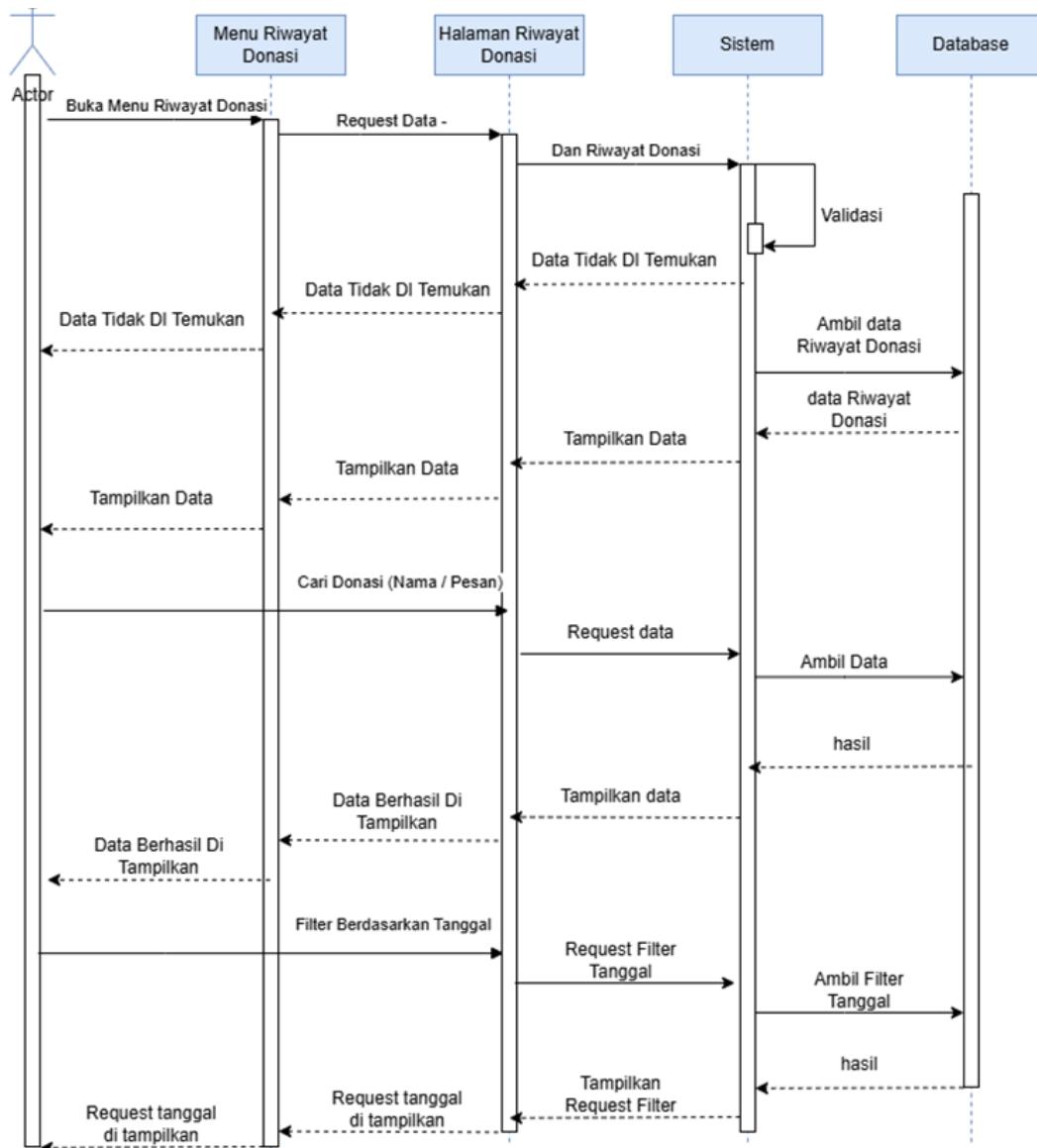
Sequence Diagram Kelola Profil



Gambar 15: Sequence Diagram Kelola Profil

Sequence diagram kelola profil menggambarkan interaksi antara kreator dan sistem dalam memproses pengelolaan data profil. Kreator membuka menu profil dan melakukan perubahan data. Sistem menerima perubahan, melakukan validasi, dan menyimpan perubahan ke basis data. Setelah proses berhasil, sistem mengirimkan respons bahwa profil berhasil diperbarui.

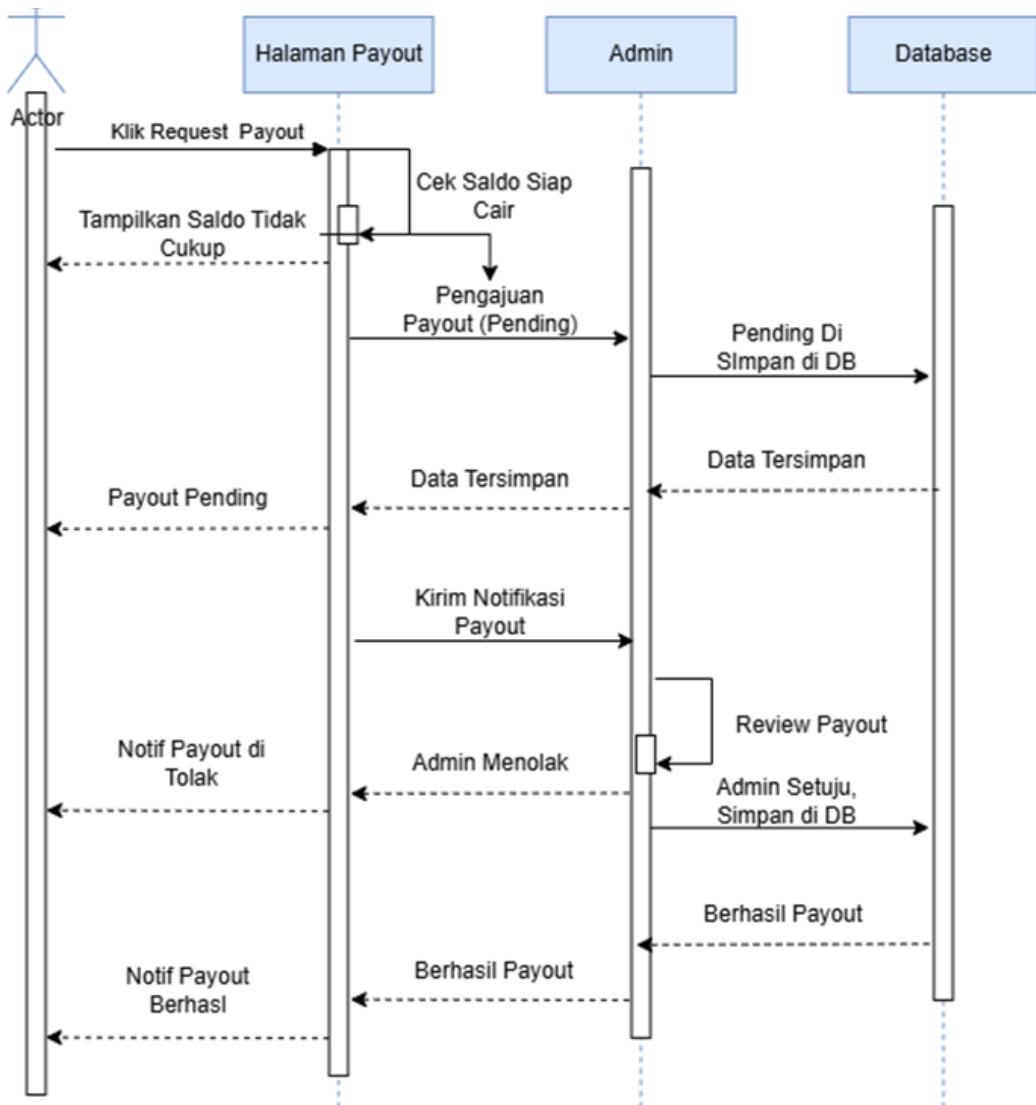
Sequence Diagram Riwayat Donasi



Gambar 16: Sequence Diagram Riwayat Donasi

Sequence diagram riwayat donasi menggambarkan proses pengambilan dan penampilan data riwayat donasi. Kreator mengakses halaman riwayat donasi, sistem melakukan validasi autentikasi, kemudian mengambil data donasi dari basis data. Kreator dapat melakukan pencarian dan filter data, dan sistem akan menampilkan hasil sesuai dengan permintaan.

Sequence Diagram Request Payout



Gambar 17: Sequence Diagram Request Payout

Sequence diagram request payout menggambarkan proses pengajuan pencairan dana oleh kreator. Kreator mengajukan permintaan payout melalui sistem. Sistem melakukan pengecekan saldo dan menyimpan data payout dengan status PENDING. Admin kemudian melakukan proses review terhadap permintaan tersebut dan memutuskan untuk menerima atau menolak. Apabila permintaan diterima, sistem memperbarui status payout dan mengirimkan notifikasi keberhasilan kepada kreator.

0.15.11 Model Koleksi

Model koleksi digunakan untuk merepresentasikan struktur data utama yang bekerja di dalam platform donasi. Setiap collection dirancang untuk mendukung proses transaksi, pengelolaan kreator, penayangan media share di overlay, hingga alur pancairan dana. Secara umum, entitas yang digunakan dapat dikelompokkan menjadi koleksi utama dan koleksi pendukung.

Koleksi utama platform meliputi:

- **Creator**

Menyimpan data kreator seperti nama, email, profil, serta informasi akun yang diperlukan untuk menerima donasi dan melakukan permintaan payout.

- **Donation**

Mencatat seluruh transaksi donasi, termasuk nominal, pesan, metode pembayaran, status (PENDING/PAID), serta relasi terhadap kreator yang menerima donasi.

- **MediaShare**

Entitas untuk menangani request media share (YouTube video) yang dikaitkan dengan donasi tertentu, termasuk durasi dan validasi media.

- **Payout**

Menyimpan informasi permintaan pencairan dana kreator, mencakup jumlah pencairan, fee platform, status (PENDING/APPROVED/PROCESSED), serta log aktivitas admin.

- **Notification**

Berfungsi untuk menampilkan data overlay secara real-time, seperti donasi terbaru atau media share yang harus ditayangkan oleh streamer atau kreator.

Selain itu, terdapat entitas pendukung yang digunakan untuk historisasi dan agregasi data, yaitu:

- **DonationHistory** – mencatat perubahan status donasi.

- **MonthlyLeaderboard** – menyimpan data peringkat donatur setiap bulan.

- **Contact** – mencatat umpan balik dari pengguna.

- **ProfileImage** – menyimpan data gambar untuk kebutuhan profil kreator.
- **Filteredwords** – menyimpan daftar kata yang difilter pada pesan donasi.
- **Donationshares** – menyimpan *record* data agregasi dari hasil *sharalink* donasi.
- **Admin** – menyimpan kredensial admin yang bertugas memverifikasi payout dan melakukan manajemen sistem.

Seluruh koleksi tersebut berperan dalam memastikan integritas data serta menghubungkan seluruh proses inti mulai dari transaksi donasi, pengelolaan kreator, hingga operasional sistem admin.

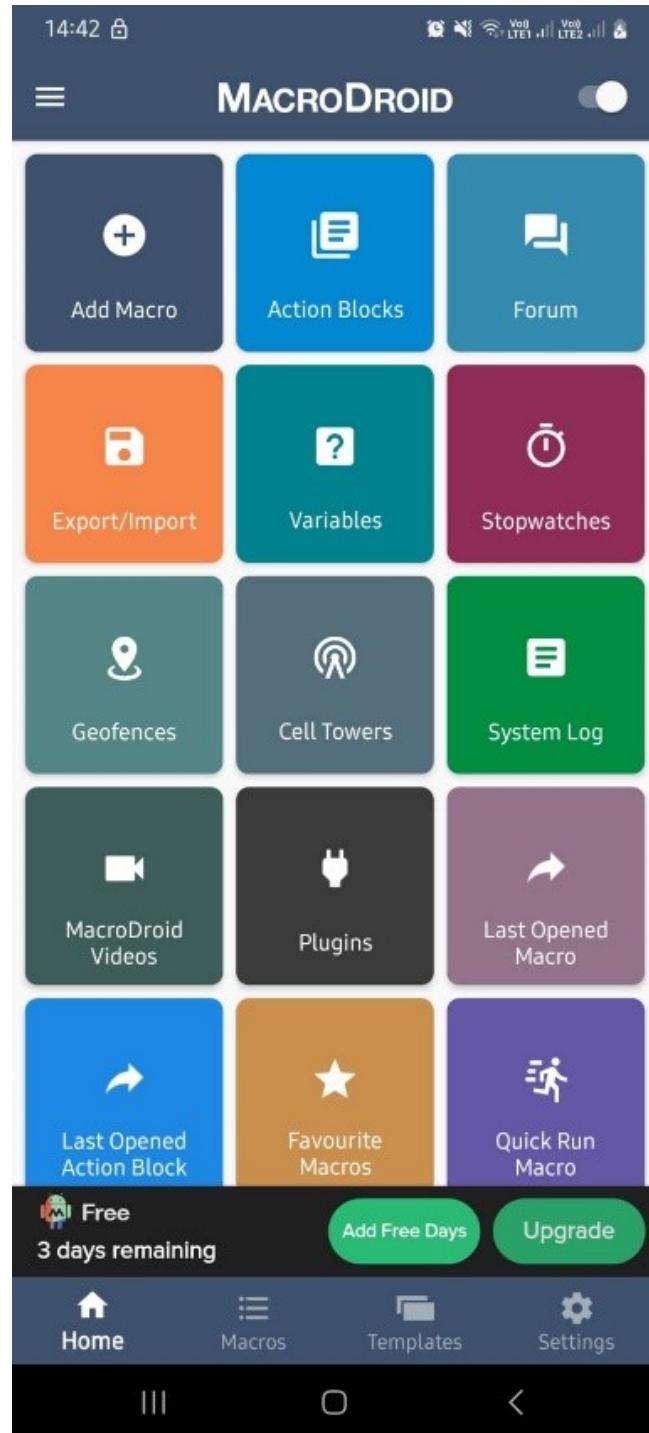
0.15.12 Gopay Merchant

GoPay Merchant digunakan sebagai metode pembayaran alternatif pada aplikasi Nyumbangin selain integrasi utama melalui Midtrans. Fitur ini memungkinkan donatur melakukan pembayaran dengan memindai kode QRIS yang disediakan melalui aplikasi merchant. Setiap transaksi yang berhasil akan menghasilkan notifikasi pada perangkat admin sebagai indikator penerimaan dana. Implementasi ini bertujuan untuk memberikan fleksibilitas metode pembayaran, khususnya untuk transaksi langsung atau kebutuhan tertentu di luar alur payment gateway utama, tanpa menggantikan sistem pembayaran terintegrasi yang telah dibangun sebelumnya.

0.15.13 Perancangan Mekanisme MacroDroid Trigger Webhook

MacroDroid digunakan sebagai mekanisme otomasi untuk mendeteksi notifikasi transaksi dari aplikasi GoPay Merchant dan meneruskannya ke sistem backend Nyumbangin. Aplikasi ini dikonfigurasi menggunakan konsep trigger dan action, di mana trigger berupa notifikasi transaksi masuk, kemudian action yang dijalankan adalah pengiriman HTTP request ke endpoint backend yang telah disediakan. Data yang diterima backend selanjutnya diproses dan disimpan ke dalam database sehingga donasi dapat tercatat secara otomatis

tanpa input manual dari admin. Pendekatan ini memungkinkan integrasi pembayaran berjalan secara semi-otomatis dengan memanfaatkan sistem notifikasi pada perangkat Android.



Gambar 18: MacroDroid

0.15.14 Integrasi Payment Gateway Midtrans

Midtrans dirancang sebagai payment gateway yang menangani proses pembayaran donasi secara eksternal. Sistem Nyumbangin hanya membuat transaksi awal dan menghasilkan Snap Token berdasarkan data donasi yang telah divalidasi. Snap Token tersebut digunakan untuk menampilkan halaman pembayaran Midtrans, sementara penentuan status akhir transaksi tidak dilakukan pada tahap ini dan dibahas pada mekanisme selanjutnya. Implementasi tampilan pembayaran Midtrans ditampilkan pada BAB IV.

0.16 Metode Perancangan Teknis

Perancangan teknis pada platform donasi ini difokuskan pada penyusunan arsitektur layanan yang aman, efisien, dan mudah dipelihara. Pendekatan utama yang digunakan adalah pemisahan tanggung jawab antar modul serta penerapan pola penanganan API yang konsisten. Setiap endpoint dirancang mengikuti alur standar, yaitu validasi metode HTTP, autentikasi menggunakan JWT untuk endpoint privat, validasi input, eksekusi operasi basis data, dan pengembalian respons dalam format JSON. Pola yang seragam ini memudahkan proses debugging serta menjaga konsistensi perilaku lintas layanan.

Dari sisi keamanan, validasi token JWT diterapkan untuk memastikan setiap permintaan memiliki otorisasi yang benar, termasuk pengecekan masa berlaku token dan jenis pengguna (donor, kreator, atau admin). Seluruh input kritis seperti username, nominal donasi, dan URL media share divalidasi untuk mencegah data tidak sah masuk ke dalam sistem. Integrasi pembayaran dirancang agar hanya bergantung pada webhook resmi Midtrans, sehingga status transaksi tidak bergantung pada aktivitas pengguna di sisi klien.

Optimasi basis data dilakukan melalui penempatan indeks pada atribut yang sering digunakan dalam kueri, seperti `createdAt`, `creatorId`, dan `creatorUsername`. Operasi agregasi seperti *leaderboard* dan statistik kreator menggunakan pipeline agregasi *MongoDB* untuk mengurangi beban pada server aplikasi. Pembatasan kueri (*limit*) diterapkan untuk mencegah pengambilan data berlebihan pada endpoint yang memiliki potensi pertumbuhan data tinggi.

Selain itu, proses donasi dan penyajian notifikasi dipisahkan dari alur pembayaran utama. Server hanya membuat record donasi dan *Snap Token* pada permintaan awal, sementara pembaruan status dan pemicu notifikasi dilakukan ketika *webhook* diterima. Pemisahan ini meningkatkan stabilitas sistem dan memastikan overlay hanya menampilkan data yang telah tervalidasi.

0.17 Metode Pengujian

Metode pengujian pada platform donasi ini dirancang untuk memastikan bahwa seluruh fungsi sistem berjalan sesuai kebutuhan, aman digunakan, dan menghasilkan data yang konsisten. Pendekatan pengujian dilakukan melalui kombinasi pengujian unit, pengujian integrasi, serta pengujian fungsional terhadap endpoint API dan alur bisnis utama. Fokus utama pengujian meliputi keakuratan proses donasi, keandalan mekanisme payout, integritas data pada leaderboard serta statistik kreator, dan validitas proses autentikasi berbasis OAuth dan JWT.

Pengujian dilakukan menggunakan data uji terkontrol, simulasi webhook pembayaran, serta verifikasi hasil secara langsung melalui basis data. Seluruh skenario kritis seperti validasi input, autentikasi dan otorisasi, serta penanganan kesalahan diuji untuk memastikan sistem tetap stabil dalam berbagai kondisi operasional.

0.17.1 Jenis Pengujian

Pengujian sistem mencakup beberapa jenis pengujian sebagai berikut:

a. Pengujian Unit (Unit Testing)

Dilakukan pada fungsi atau modul kecil yang berdiri sendiri, seperti:

- Validasi token JWT (sign dan verify),
- Perhitungan *platformFee* dan *finalAmount* pada payout,
- Validasi input donasi (amount, username, media share).

b. Pengujian Integrasi (Integration Testing)

Berfokus pada alur yang melibatkan beberapa komponen, seperti:

- Proses donasi lengkap (create donation → Snap Token → pembayaran → webhook → update status),
- Permintaan payout oleh kreator dan proses approval oleh admin,
- Penampilan notifikasi overlay berdasarkan data terbaru.

c. Pengujian Fungsional (Functional Testing)

Dilakukan untuk mengevaluasi apakah setiap endpoint memenuhi kebutuhan fungsional yang telah ditetapkan, seperti:

- Validasi username saat login atau donasi,

- Pembatasan akses endpoint dashboard menggunakan JWT,
- Respons error 400, 401, 404, dan 500 yang konsisten.

d. **Pengujian Keamanan (Security Testing)**

Meliputi:

- Akses API tanpa token harus ditolak,
- Token kedaluwarsa atau *userType* tidak sesuai harus gagal,
- Percobaan pengiriman payload tidak valid harus tervalidasi.

e. **Pengujian Kinerja (Performance Testing)**

Difokuskan pada:

- Kecepatan query leaderboard (target < 300 ms),
- Stabilitas respons webhook,
- Performa overlay saat proses polling data.

0.17.2 Skenario Pengujian

Beberapa skenario pengujian utama yang digunakan meliputi:

1. **Skenario 1 – Donasi Berhasil**

Input valid → server membuat record PENDING → Snap Token sukses → pembayaran di Midtrans → webhook diterima → status menjadi PAID → overlay menampilkan donasi.

2. **Skenario 2 – Donasi Gagal Validasi**

Nominal di bawah minimum atau format URL salah → server mengembalikan status 400.

3. **Skenario 3 – Akses Endpoint Tanpa Token**

Mengakses leaderboard atau payout tanpa header *Authorization* → sistem mengembalikan 401 *Unauthorized*.

4. **Skenario 4 – Request Payout oleh Kreator**

Saldo cukup → request dicatat → status PENDING → admin review → APPROVED → status PROCESSED → saldo kreator berkurang sesuai *finalAmount*.

5. Skenario 5 – Token Salah atau Kedaluwarsa

Token tidak valid atau kedaluwarsa → akses ditolak dengan pesan error yang konsisten.

6. Skenario 6 – Webhook Simulasi Midtrans

Webhook dikirim manual dari Postman → status donasi berubah menjadi PAID → overlay menampilkan notifikasi.

0.17.3 Alat Pengujian

Alat yang digunakan dalam proses pengujian meliputi:

- **Postman** – untuk pengujian API (header, body, autentikasi).
- **MongoDB Compass** – untuk memverifikasi perubahan data secara langsung.
- **Logging Next.js (console)** – untuk memantau webhook, error, dan alur proses.

0.18 Evaluasi Keberhasilan

Evaluasi keberhasilan dilakukan untuk menilai sejauh mana implementasi sistem memenuhi kebutuhan fungsional, stabilitas operasional, serta ketepatan mekanisme kritis seperti autentikasi, pengelolaan sesi, dan pemrosesan pembayaran. Penilaian dilakukan melalui pengujian terstruktur dan analisis hasil *code coverage* yang dihasilkan dari proses unit testing pada modul-modul inti.

0.18.1 Cakupan Pengujian

Pengujian difokuskan pada komponen yang tergolong kritikal bagi kelangsungan sistem, meliputi:

- Autentikasi dan Otorisasi (JWT, session handling, route protection)
- Pemrosesan Pembayaran (Midtrans Snap dan webhook)
- Fungsi donasi (validasi input, perhitungan status)
- Fungsi payout (perhitungan *finalAmount*, status machine payout)

Komponen lain seperti UI Component dan utilitas pendukung tidak menjadi prioritas utama dalam pengujian ini karena tidak langsung berpengaruh pada keamanan maupun alur transaksi.

0.18.2 Hasil Code Coverage

Pengujian menghasilkan metrik *code coverage* sebagai berikut:

Tabel 1: Code Coverage

Metode	Cakupan
Statements	14.71%
Branch	15.56%
Functions	29.35%
Lines	14.21%

Meskipun angka coverage terlihat rendah secara keseluruhan, hal ini tidak mencerminkan kualitas fungsional sistem secara langsung karena coverage tidak mencakup seluruh file, melainkan hanya modul-modul yang dipilih berdasarkan kategori kritis. File non-kritis seperti UI, utilitas ringan, dan helper statis tidak disertakan dalam pengujian sehingga turut menurunkan total persentase.

Hasil tersebut menegaskan bahwa:

- Seluruh proses inti (autentikasi, sesi, validasi donasi, perhitungan payout, dan pembayaran) telah berhasil diuji.
- Jalur eksekusi utama (*main happy path*) telah tervalidasi.
- Beberapa jalur error belum sepenuhnya dicakup, sehingga membuka peluang peningkatan lebih lanjut.

0.18.3 Interpretasi dan Evaluasi

Berdasarkan hasil pengujian dan *code coverage* tersebut, dapat disimpulkan bahwa:

1. Stabilitas alur bisnis utama telah terverifikasi, terutama mekanisme donasi dan payout yang melibatkan transaksi dan webhook.
2. Keamanan dasar terkait autentikasi, JWT, dan sesi telah diuji dan berfungsi sesuai kebutuhan.
3. Konsistensi data pada proses pembayaran serta pencatatan notifikasi berhasil diuji melalui simulasi webhook.

4. Rendahnya nilai coverage lebih disebabkan oleh fokus pengujian pada modul kritis, bukan karena seluruh sistem tidak diuji.
5. Sistem dinilai layak digunakan, namun peningkatan cakupan pengujian tetap direkomendasikan untuk modul non-kritis seperti UI dan utilitas.

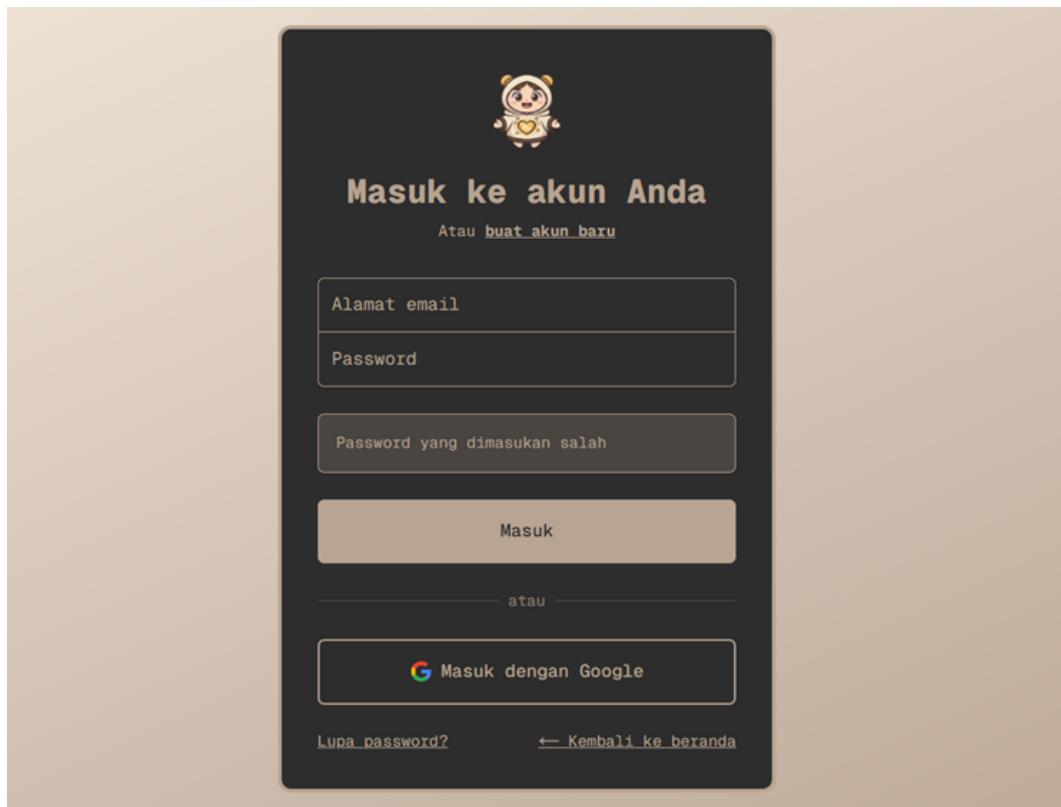
0.18.4 Kesimpulan Evaluasi

Secara keseluruhan, sistem telah memenuhi fungsi utamanya mulai dari pemrosesan donasi, pembayaran, hingga payout dan notifikasi. Hasil pengujian menunjukkan bahwa fitur inti berjalan stabil, meskipun pengujian yang lebih luas masih diperlukan pada tahap pengembangan selanjutnya.

BAB IV

HASIL DAN PEMBAHASAN

0.19 Halaman Autentikasi



Gambar 19: Halaman Login

Halaman autentikasi menampilkan dua opsi untuk login, yaitu login manual dan login menggunakan Google OAuth untuk mempermudah proses autentifikasi pengguna.

0.20 Halaman Donasi

0.20.1 Form Donasi

The screenshot shows a donation form titled "Dukung Spiderman". The form fields include:

- Nama Anda ***: An input field labeled "Masukkan nama Anda".
- Jumlah Donasi ***: A section with five buttons for standard donations: "Rp 5.000", "Rp 10.000", "Rp 25.000", "Rp 50.000", and "Rp 100.000".
- Atau masukkan jumlah custom**: A text input field for custom donation amounts.
- Pesan (Opsional)**: A text input field labeled "Tulis pesan dukungan Anda ...".
- Media Share (YouTube)**: A button with a YouTube icon and the text "Nonaktif".
- Saya menyetujui bahwa donasi ini bersifat sukarela dan tidak dapat dikembalikan. Saya telah membaca dan menyetujui [Syarat dan Ketentuan Nyumbangin](#).**: A checkbox statement with a link to terms and conditions.
- Donasi Rp 0**: A large button at the bottom.

Gambar 20: Form Donasi

Halaman donasi menyediakan input seperti nama donor, nominal donasi, pesan, serta opsi media share. Sistem melakukan validasi dasar terhadap data yang dimasukkan sebelum melanjutkan proses pembayaran ke Midtrans.

0.20.2 Pembayaran Donasi Menggunakan QRIS Gopay Merchant

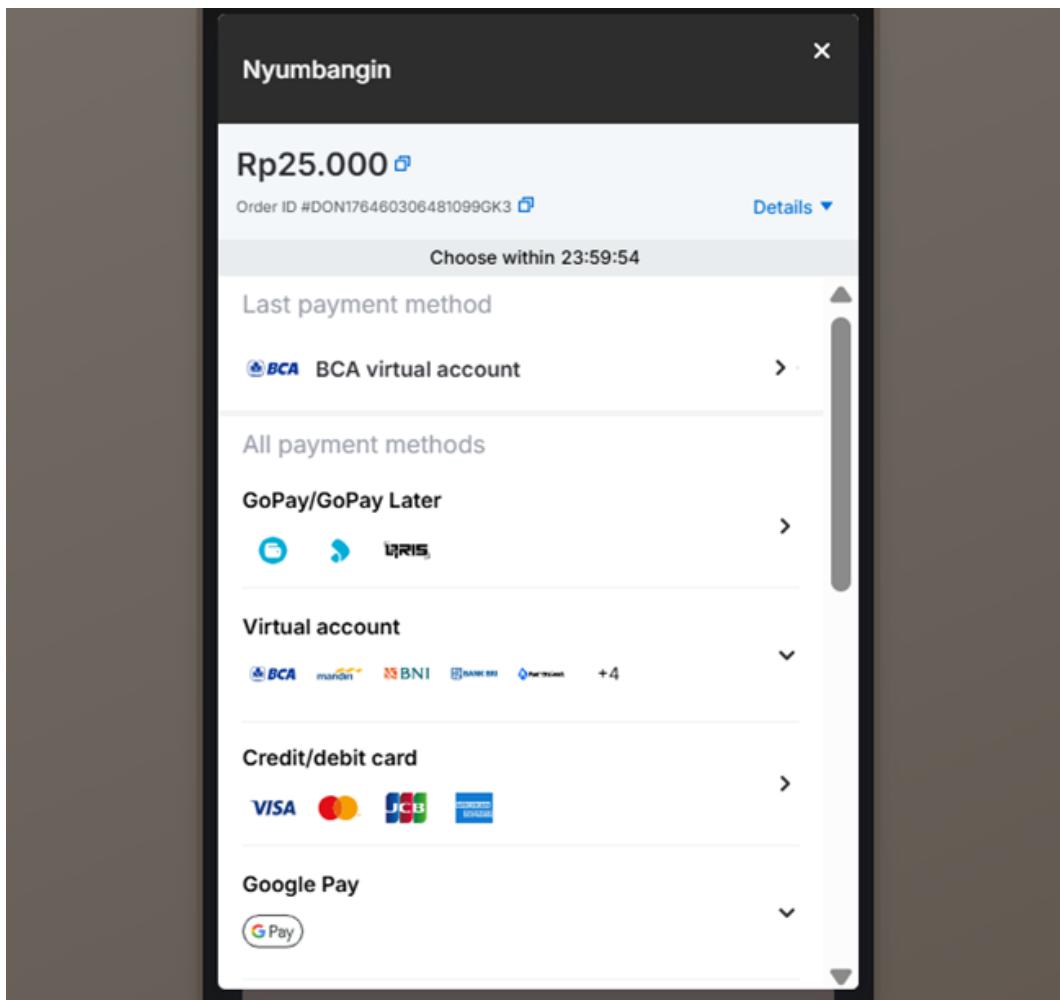


Gambar 21: QRIS Gopay Merchant

Aplikasi Nyumbangin mengimplementasikan pembayaran donasi menggunakan QRIS GoPay Merchant untuk mempermudah donatur dalam melakukan transaksi digital. QRIS ditampilkan sebagai kode pembayaran resmi atas nama merchant Nyumbangin, Digital & Kreatif, seperti ditunjukkan pada Gambar 4.3, dan digunakan oleh donatur dengan cara memindai kode QR melalui aplikasi pembayaran yang mendukung standar QRIS. Proses pembayaran dilakukan di luar sistem aplikasi, sementara konfirmasi keberhasilan transaksi

ditentukan melalui mekanisme webhook yang memperbarui status donasi di sistem sebelum notifikasi dan proses lanjutan dijalankan.

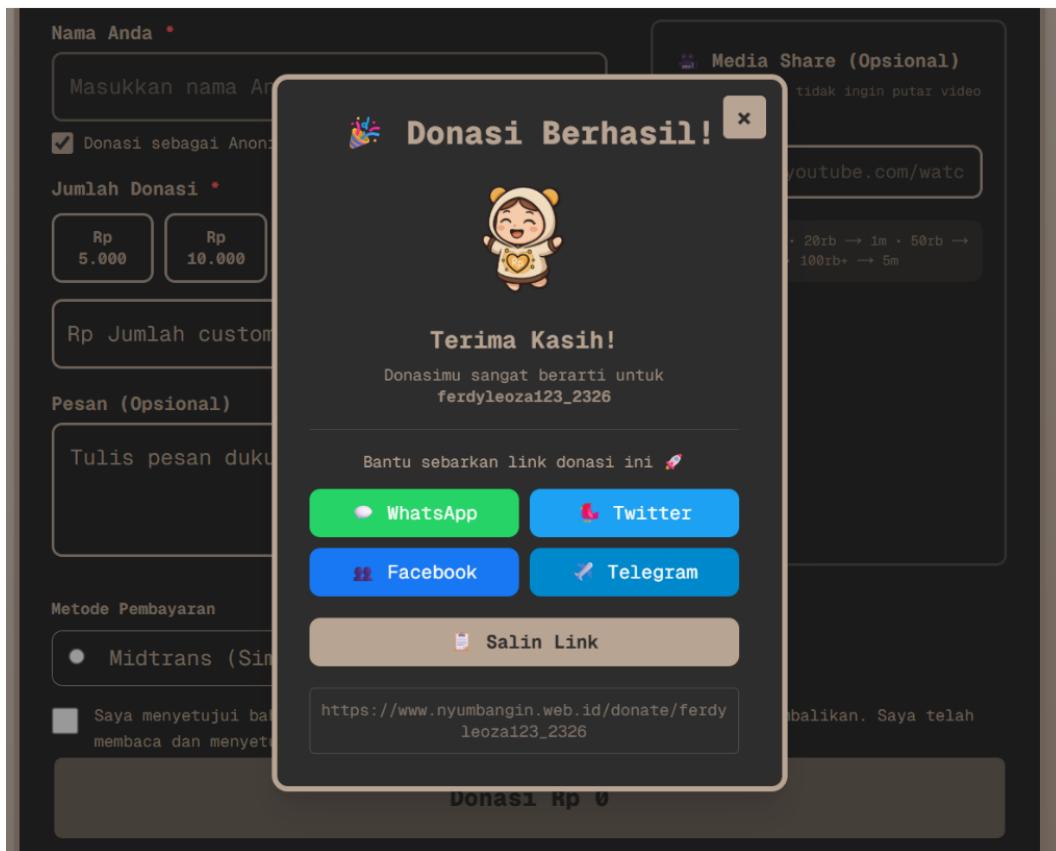
0.20.3 Halaman Pembayaran Midtrans



Gambar 22: Halaman Pembayaran Midtrans

Setelah form donasi berhasil divalidasi, sistem mengirimkan permintaan *Snap Token* ke Midtrans. Donor kemudian diarahkan ke halaman pembayaran yang disediakan oleh Midtrans untuk menyelesaikan proses transaksi.

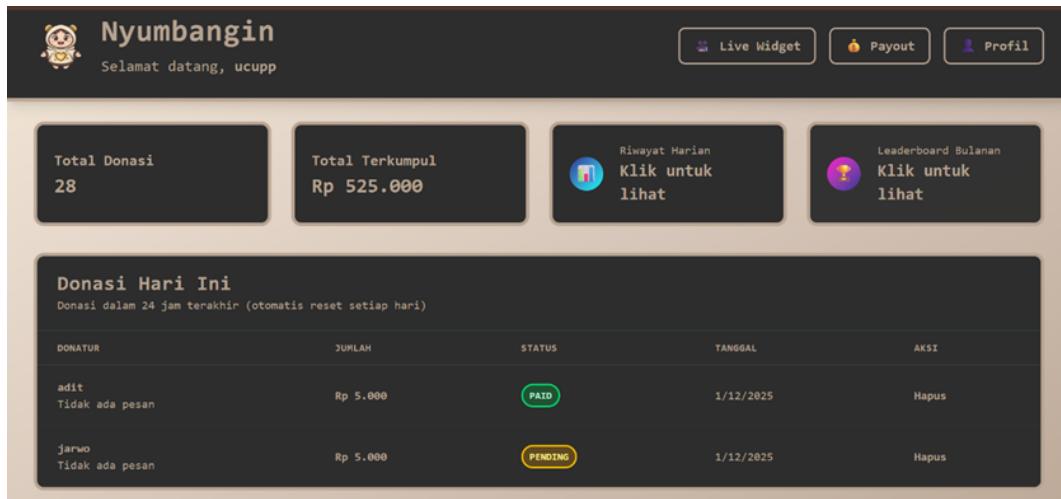
0.20.4 Fitur Sharelink Donasi



Gambar 23: Fitur Sharelink Donasi

Setelah donasi berhasil dan status transaksi tervalidasi, sistem menampilkan dialog konfirmasi yang menyediakan fitur social sharing berupa tombol berbagai tautan donasi ke beberapa platform media sosial serta opsi menyalin tautan. Fitur ini bertujuan mendorong penyebaran link donasi secara organik dan meningkatkan keterlibatan sosial.

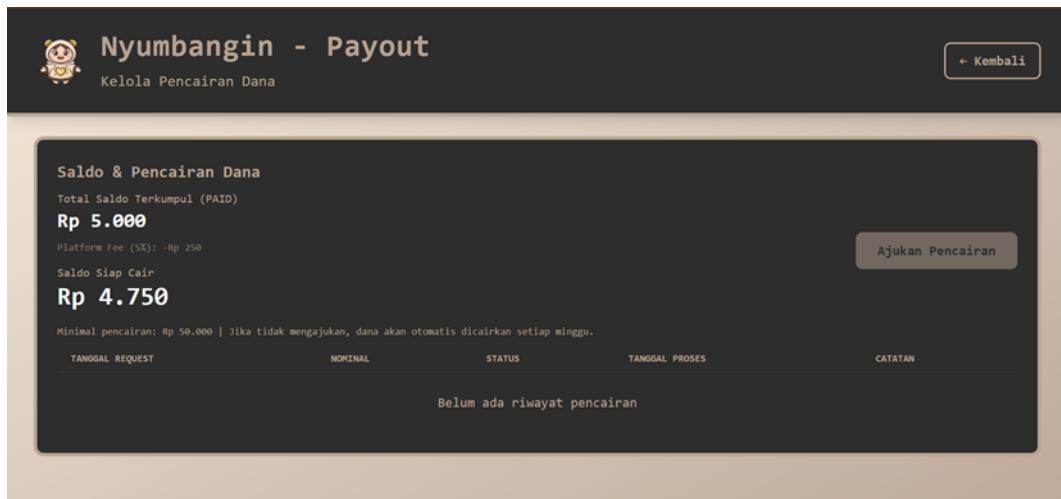
0.21 Dashboard Kreator



Gambar 24: Dashboard Kreator

Dashboard kreator menampilkan statistik utama seperti total donasi, total pendapatan, menu riwayat donasi, serta menu leaderboard donasi terbanyak per bulan.

0.22 Halaman Request Payout



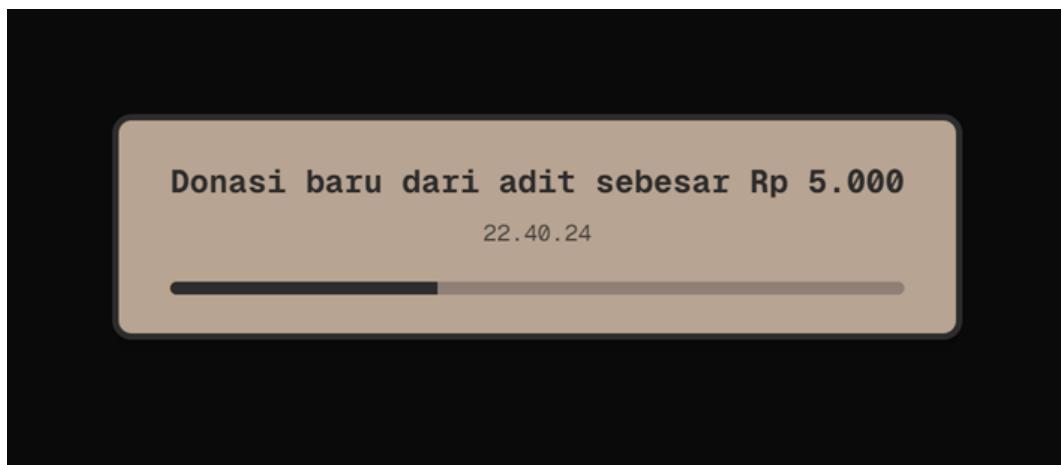
Gambar 25: Halaman Payout

Kreator dapat mengajukan pencairan dana berdasarkan saldo yang tersedia. Sistem menghitung saldo bersih berdasarkan total donasi dengan status PAID yang belum pernah diproses dalam payout sebelumnya.

0.23 Overlay

Overlay berfungsi sebagai antarmuka yang ditampilkan pada platform streaming melalui OBS. Seluruh elemen *overlay* diambil secara real-time dari *API* sehingga kreator dapat menampilkan interaksi donasi secara langsung kepada penonton. Subbab ini menampilkan implementasi setiap komponen overlay.

0.23.1 Overlay Notifikasi Donasi



Gambar 26: Notifikasi Donasi

Overlay menampilkan notifikasi donasi secara real-time yang diambil melalui mekanisme polling API. Tampilan ini dikonfigurasi agar kompatibel dengan OBS untuk keperluan streaming.

0.23.2 Overlay Media Share



Gambar 27: Media Share

Overlay Media Share menampilkan video YouTube yang diputar berdasarkan permintaan donor. Pada tampilan ini, video muncul di area utama layar, sementara informasi donasi seperti nama donor, nominal, dan pesan singkat ditampilkan pada bagian bawah kiri layar.

0.23.3 Overlay QR Link Donasi



Gambar 28: QR Link Donasi

Overlay menyediakan kode QR statis yang mengarahkan penonton langsung ke halaman donasi kreator. QR ini memfasilitasi penonton untuk berdonasi tanpa harus mengetik tautan donasi secara manual.

0.23.4 Overlay Leaderboard



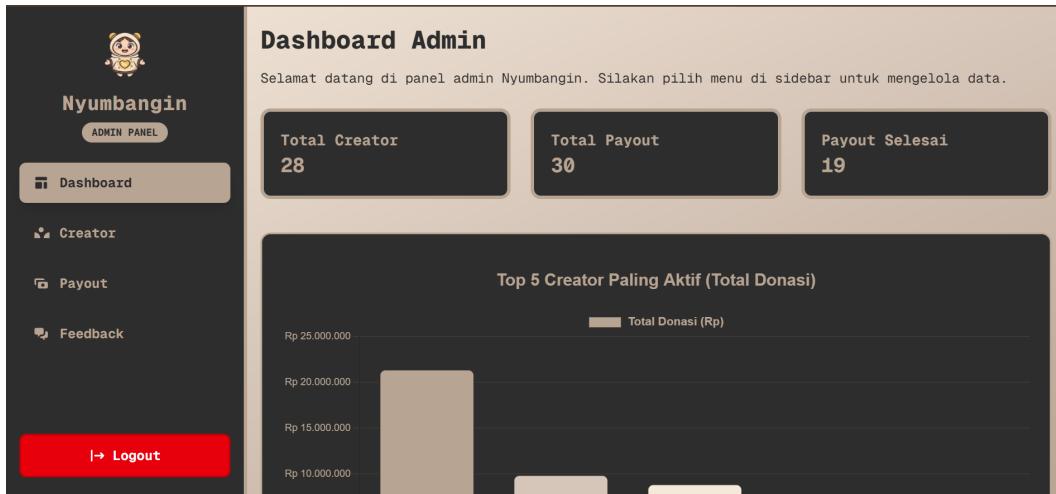
Gambar 29: Leaderboard Donatur

Overlay leaderboard menampilkan daftar pendukung terbesar (*top donors*) atau peringkat total donasi yang telah dioptimasi menggunakan mekanisme limit dan sorting. Tampilan ini digunakan untuk memberikan apresiasi kepada donatur selama sesi streaming berlangsung.

0.24 Dashboard Admin

Bagian ini menampilkan antarmuka yang digunakan admin untuk mengelola aktivitas pada platform, mulai dari pemantauan data donasi hingga pengelolaan kreator dan proses payout. Dashboard admin terdiri dari tiga menu utama, yaitu Dashboard, Creator, dan Payout.

0.24.1 Halaman Dashboard



Gambar 30: Dashboard Admin

Halaman dashboard menampilkan ringkasan statistik utama sistem, seperti jumlah kreator terdaftar, jumlah pengajuan payout, serta jumlah payout yang telah diselesaikan. Selain itu, halaman ini juga memvisualisasikan Top 5 Kreator Paling Aktif berdasarkan total donasi yang diterima.

0.24.2 Halaman Creator

The screenshot shows the Nyumbangin Creator page. The sidebar on the left is identical to the Admin Dashboard. The main area is titled 'Daftar Creator' with the sub-instruction 'Kelola akun creator yang terdaftar di platform'. A search bar says 'Cari creator...'. Below is a table of creators with columns: USERNAME, NAMA, PAYOUT, and AKSI. A modal window titled 'Detail Creator' is open over the table, showing the details for the creator 'sutsudit_'. The modal includes fields for Username, Email, Nama, Payout Account, Payout Holder, Payout Type, and Tanggal Daftar. The table lists several other creators like 'raihanherm', 'ferdy5', 'alissafani', 'ferdyleoza', 'gaveryn1', 'raihanaditya1506_2456', etc., each with their status (Lengkap or Belum) and a 'Hapus' button.

Gambar 31: Tabel Daftar Creator dan Detail Creator

Halaman creator digunakan untuk melihat daftar kreator yang terdaftar pada platform, termasuk informasi status kelengkapan data payout masing-masing.

Admin dapat melakukan proses pencairan dana kreator serta melihat detail lengkap akun kreator melalui dialog *Detail Creator*.

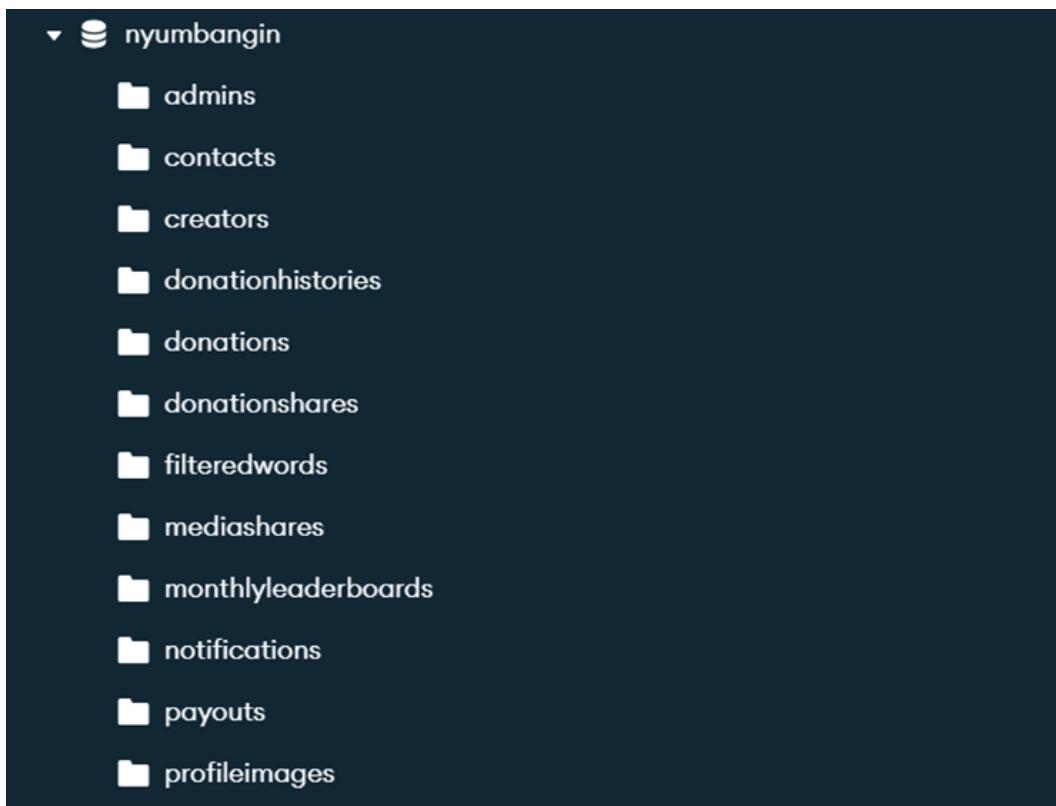
0.24.3 Halaman Payout

USERNAME	TANGGAL REQUEST	NOMINAL	STATUS	TANGGAL PROSES	CATATAN	AKSI
ferdyleoza123_2326	14/1/2026, 16.07.24	Rp 95.000,00	Pending	-	-	<button>Proses</button> <button>Tolak</button>
ferdyleoza123_2326	14/1/2026, 16.03.34	Rp 9.212.121,00	Selesai	14/1/2026, 16.04.25	cair cair	
ferdyleoza123_2326	14/1/2026, 15.40.02	Rp 9.500.000,00	Selesai	14/1/2026, 15.57.27	done	
ferdyleoza123_2326	14/1/2026, 14.04.05	Rp 95.000,00	Selesai	14/1/2026, 14.05.06	done	
raihanhermawan85_1652	12/1/2026, 19.53.30	Rp 95.000,00	Selesai	12/1/2026, 19.58.39	-	
ferdyleoza123_2326	12/1/2026, 13.44.41	Rp 570.000,00	Selesai	12/1/2026, 13.45.26	-	
ferdyleoza123_2326	12/1/2026, 13.13.01	Rp 285.000,00	Selesai	12/1/2026, 13.18.23	done	

Gambar 32: Tampilan Payout

Halaman payout menampilkan daftar pengajuan pencairan dana yang dilakukan oleh para kreator. Admin dapat memantau nominal pencairan, waktu pengajuan, serta catatan yang terkait dengan setiap payout. Halaman ini membantu admin memastikan bahwa seluruh proses pencairan berjalan dengan transparan dan terdokumentasi.

0.25 Struktur Basis Data



Gambar 33: Basis Data

Sistem menggunakan MongoDB sebagai basis data utama. Setiap fitur pada sistem menghasilkan koleksi tersendiri untuk memudahkan proses penyimpanan, pemantauan, dan analisis data.

Koleksi yang terbentuk antara lain:

- **admins** – menyimpan data akun admin yang memiliki akses panel pengelolaan.
- **contacts** – menyimpan data feedback atau saran dan keluhan dari pengguna.
- **creators** – menyimpan informasi kreator, termasuk data payout dan profil.
- **donations** – mencatat data donasi sementara yang masuk dari pengguna yang nantinya dialihkan ke koleksi donationhistories.

- **donationhistories** – menyimpan data riwayat donasi yang berasal dari koleksi donations.
- **donationshares** – mencatat data share link donasi.
- **filteredwords** – menyimpan data filter kata-kata yang tidak pantas.
- **mediashares** – menyimpan data media share berupa request video YouTube.
- **monthlyleaderboards** – menyimpan data peringkat donor per bulan.
- **notifications** – menyimpan data notifikasi donasi yang ditampilkan pada overlay.
- **payouts** – mencatat pengajuan dan proses pencairan dana oleh kreator.
- **profileimages** – menyimpan informasi gambar profil kreator.

BAB V

KESIMPULAN DAN SARAN

0.26 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa platform Nyumbangin berhasil dikembangkan sebagai prototipe platform donasi digital berbasis web yang fungsional. Sistem mampu menangani alur utama donasi, mulai dari autentikasi pengguna, pencatatan transaksi, integrasi pembayaran, notifikasi real-time melalui overlay, hingga mekanisme pencairan dana (payout) bagi kreator secara tersusunan.

Fitur-fitur inti yang dirancang pada tahap analisis telah diimplementasikan dan berjalan sesuai kebutuhan. Penggunaan arsitektur aplikasi web modern serta pemodelan sistem menggunakan UML dan ERD membantu memastikan alur proses dan struktur data berjalan konsisten. Hasil pengujian menunjukkan

bahwa proses kritis seperti donasi, pembayaran, webhook, dan payout dapat berjalan stabil dan menghasilkan data yang valid.

Pengujian sistem difokuskan pada modul-modul utama yang bersifat kritis, sehingga hasil *code coverage* yang diperoleh hanya merepresentasikan pengujian fitur inti dan tidak mencakup seluruh modul aplikasi, khususnya antarmuka pengguna dan utilitas pendukung. Meskipun demikian, sistem telah memenuhi tujuan pengembangan sebagai platform donasi digital yang operasional. Pengembangan lanjutan masih dapat dilakukan, terutama pada perluasan cakupan pengujian, peningkatan keamanan, dan optimasi performa untuk penggunaan pada skala yang lebih besar.

Bibliografi