

RANCANG BANGUN BACKEND WEBSITE PEMUNGUTAN SUARA DENGAN MENGGUNAKAN FRAMEWORK EXPRESS.JS

Muhammad Nauval Saiholau¹, Muhammad Bintang Pramadipta², Wahyu Eko Sulistiono³, Yessi Mulyani⁴

^{1, 2, 3, 4} Jurusan Teknik Elektro Universitas Lampung, Bandar Lampung

Riwayat artikel:

Received: 27 Maret 2024

Accepted: 30 Maret 2024

Published: 2 April 2024

Keywords:

Online Voting, Online Voting System, Iterative Development Model, Express.js.

Correspondent

nauvalsm@gmail.com

Email:

Abstrak. Pemungutan suara online merupakan solusi yang efisien dan modern untuk mengatasi permasalahan pemungutan suara tradisional. Penelitian ini bertujuan untuk merancang dan membangun backend situs web pemungutan suara online yang bernama PilihGeh dengan menggunakan framework Express.js. Backend situs web ini mendukung fitur-fitur seperti autentikasi, penyelenggaraan, pengelolaan, dan pelaksanaan pemungutan suara yang aman, valid, dan fleksibel. Metode penelitian yang digunakan adalah Iterative Development Model yang terdiri dari analisis, desain sistem, coding, testing, dan deployment. Hasil penelitian menunjukkan bahwa backend situs web ini berhasil dibangun dan diuji dengan menggunakan aplikasi Postman untuk 132 skenario. Backend situs web ini juga berhasil di deploy dengan menggunakan layanan Railway. Penelitian ini memberikan kontribusi untuk pengembangan sistem pemungutan suara online yang dapat digunakan oleh berbagai organisasi atau komunitas.

Abstract. The traditional voting method has many problems, such as the risk of error, high cost, and long time. Therefore, an online voting system that is more modern and efficient is needed. This study aims to design and build the backend website of the online voting system named PilihGeh using Express.js framework. The backend website of this system supports features such as authentication, organization, management, and execution of voting that are safe, valid, and flexible. The research method used is Iterative Development Model which consists of analysis, system design, coding, testing, and deployment. The research results show that the backend website of this system was successfully built and tested using Postman for 132 scenarios. The backend website of this system was also successfully deployed using Railway service. This study contributes to the development of an online voting system that can be used by various organizations or communities.

1. PENDAHULUAN

Pemungutan suara telah menjadi pilar utama dalam keputusan demokratis di berbagai organisasi dan lembaga. Di berbagai negara, termasuk Indonesia, pemilihan dengan menggunakan kertas suara dan kotak suara masih menjadi metode utama. Namun, cara tradisional ini memiliki kelemahan yang

signifikan. Risiko kesalahan dalam verifikasi suara, perhitungan yang rentan terhadap kesalahan, dan proses yang memakan waktu serta biaya yang tidak sedikit merupakan tantangan utama dalam sistem ini.

Pemilihan tradisional sering kali rentan terhadap kesalahan verifikasi suara dan perhitungan, yang dapat mengakibatkan

keraguan dalam hasil akhir. Di samping itu, proses manual yang diperlukan dalam pemilihan ini memakan waktu dan memerlukan biaya yang signifikan. Di Indonesia, seperti halnya di banyak negara lain, metode tradisional ini belum sepenuhnya terhindar dari potensi manipulasi dan kecurangan, seperti pemilih ganda atau kegiatan yang tidak jujur dalam proses perhitungan.

Sejumlah penelitian sebelumnya telah menjadi landasan kritis yang memotivasi pengembangan "PilihGeh". Contohnya pada penelitian [1] menyoroti risiko keamanan data akibat database yang tidak terenkripsi dan masalah verifikasi menggunakan QR Code sekali pakai. Demikian juga jurnal [2] menunjukkan keterbatasan dalam verifikasi menggunakan akun login. Begitu juga dengan jurnal [3] yang menyoroti kekurangan dalam verifikasi pemilik suara dan kurangnya informasi terkait keamanan enkripsi data. Pada skripsi [4] menunjukkan metode verifikasi awal melalui pengiriman token ke email yang memiliki celah terkait perwakilan suara. Begitu juga dalam jurnal [5] dimana sistem hanya menggunakan token sebagai verifikasi, namun terbuka terhadap perwakilan suara oleh orang lain.

Oleh karena itu, dalam penelitian ini, fokus utamanya adalah merancang dan membangun sebuah backend situs web yang dapat mengatasi permasalahan-permasalahan baru dalam pemungutan suara secara online. Situs web "PilihGeh" diharapkan menjadi platform pemungutan suara yang mampu mengatasi kendala-kendala yang disebutkan sebelumnya.

2. TINJAUAN PUSTAKA

2.1. Framework

Framework adalah sekumpulan fungsi, class, dan aturan-aturan. Framework bersifat menyeluruh mengatur bagaimana membangun aplikasi. Framework memungkinkan membangun aplikasi dengan lebih cepat karena developer akan lebih memfokuskan pada pokok permasalahan. Definisi lain dari Framework adalah kumpulan intruksi-intruksi yang dikumpulkan dalam class dan function-function dengan fungsi masing-masing untuk memudahkan developer dalam memanggilnya tanpa harus menuliskan syntax program yang

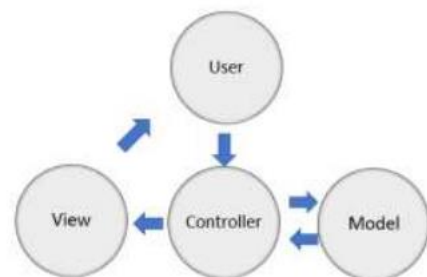
sama berulang-ulang serta dapat menghemat waktu [6].

2.2. NodeJs

Node.JS adalah sistem perangkat lunak yang didesain untuk pengembangan aplikasi web. Aplikasi ini ditulis dalam campuran Bahasa C++ dan juga JavaScript, mempunyai model event driven (basis event) dan asynchronous I/O. Tidak seperti kebanyakan bahasa JavaScript yang dijalankan pada web browser, Node.JS dieksekusi sebagai aplikasi server. Node.JS dapat berjalan di server karena dukungan dari V8 Engine buatan Google dan beberapa modul bawaan yang terintegrasi seperti modul http, modul filesystem, modul security dan beberapa modul penting lainnya [7].

2.3. ExpressJs

ExpressJs merupakan framework web paling populer di NodeJs. Berkat dokumentasinya yang lengkap dan mudah digunakan, membuat pengembang mudah dalam memakai ExpressJs untuk mengembangkan berbagai produk seperti aplikasi web dan RESTful API. [7]



Gambar 1 Arsitektur MVC

Pada gambar 1 terlihat bahwa MVC aplikasi dibagi menjadi tiga kategori: model sebagai domain utama aplikasi, view atau presentasi data dari model, dan interaksi user. Model adalah tempat semua logika bisnis suatu aplikasi disimpan. View atau tampilan adalah tempat semua elemen antarmuka pengguna aplikasi disimpan. Controller adalah komponen yang menghubungkan antara model dan view secara bersamaan. [8]

2.4. REST

REST (Representational State Transfer) merupakan gaya arsitektur dalam mendesain suatu web service. REST menggunakan metode

HTTP request untuk mengirimkan perintah kepada server. Sedangkan pada REST API hanya digunakan empat metode yaitu GET, POST, PUT, dan DELETE. [9]

2.5. PostgreSQL

PostgreSQL adalah sebuah Object-Relational Database Management System (ORDBMS) yang bersifat open source yang mendukung Structured Query Language (SQL) [6]. PostgreSQL yang merupakan pelopor berbagai aplikasi RDBMS menjadi komersil. PostgreSQL memiliki lisensi GPL (General Public License) yang membuat PostgreSQL dapat digunakan oleh setiap orang tanpa perlu membayar lisensi baik untuk keperluan pribadi maupun komersil. [10]

2.6. Unified Modeling Language

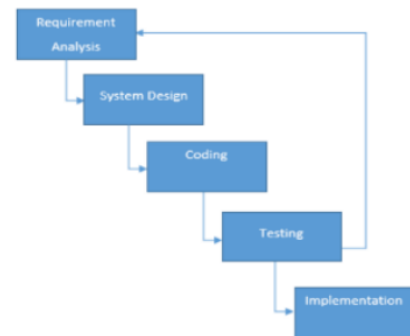
Unified Modeling Language (UML) merupakan sebuah metode pemodelan yang menggunakan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (Object-Oriented). Untuk dapat memahami UML membutuhkan bentuk konsep dari sebuah bahasa model, dan mempelajari 3 (tiga) elemen utama dari UML seperti building block, aturan-aturan yang menyatakan bagaimana building block diletakkan secara bersamaan, dan beberapa mekanisme umum. [11]

2.7. Iterative Development Model

Dengan pengembangan iteratif, proyek akan dibagi menjadi bagian-bagian kecil. Hal ini memungkinkan tim pengembang untuk menunjukkan hasil sebelumnya dan mendapatkan umpan balik dari pengguna sistem. Setiap iterasi merupakan proses mini-Waterfall dengan umpan balik dari satu fase yang memberikan informasi penting untuk desain tahap selanjutnya [12].

3. METODE PENELITIAN

Metode penelitian yang diimplementasikan dalam rancang bangun sistem *Backend* ini adalah metode Iterative Development Model. Berikut visualisasi model iteratif:



Gambar 2 Model Iteratif

Pada model iteratif, iterasi akan dilakukan berulang kali dengan tujuan untuk mencapai hasil yang diinginkan dari fase sebelumnya. Seperti yang terlihat pada gambar 2, model iteratif terdapat 5 tahapan yang menjadi inti dalam pengembangan, yaitu:

3.1. Analisis

Kegiatan analisis dimulai dengan mengumpulkan informasi yang dibutuhkan dalam rancang bangun sistem Backend ini. Hal ini dilakukan agar mudah memahami konteks bisnis perangkat lunak serta mendapatkan pandangan umum terhadap output dan fungsi utama dari sistem

3.2. Desain Sistem

Kegiatan perencanaan dilanjutkan dengan pembuatan desain sistem situs web PilihGeh. Rancang bangun Backend PilihGeh di desain dengan menggunakan Unified Modelling Language yaitu Use Case Diagram, Activity Diagram, dan menggunakan Entity Relationship Diagram.

3.3. Coding

Kegiatan perencanaan setelah desain kerangka sistem *Backend*, akan dilanjutkan dengan pembuatan serangkaian tes yang akan dijalankan sistem. Serangkaian tes yang dibuat akan menjadi acuan untuk pembuatan kode untuk melalui tes dengan sukses. Pembuatan rangkaian tes akan menggunakan aplikasi Postman. Sedangkan pembuatan kode akan menggunakan framework ExpressJs yang berbasis NodeJs dengan *software code editor* yaitu Visual Studio Code untuk melakukan rancang bangun sistem *Backend* website PilihGeh.

3.4. Testing

Proses testing dilakukan setelah penyelesaian kode untuk menemukan kesalahan dan memastikan bahwa kode memenuhi kebutuhan sistem. Fokus utama testing adalah keamanan dan kinerja sistem Backend tanpa mengurangi kebutuhan sistem. Testing dilakukan menggunakan aplikasi Postman untuk memeriksa jalannya API serta melakukan uji coba pada berbagai skenario yang mungkin terjadi.

3.5. Deployment

Proses implementasi dilakukan setelah proses testing selesai. Proses ini dilakukan dengan melakukan deployment ke layanan pihak ketiga. Pihak ketiga yang disebutkan akan ditentukan menyesuaikan dengan fitur yang dibutuhkan, dalam hal ini layanan yang akan digunakan adalah Railway

4. HASIL DAN PEMBAHASAN

4.1. Analisis

Tahap ini diperoleh informasi yang dibutuhkan untuk perancangan sistem Backend, yaitu kebutuhan fungsional dan non-fungsional dari situs web PilihGeh.

4.1.1. Kebutuhan Fungsional

Berikut adalah kebutuhan fungsional dari situs web ini:

- Backend dapat menyediakan fitur registrasi dan login ke akun
- Backend dapat mengirimkan OTP untuk autentikasi ke email/nomor Whatsapp
- Backend dapat memberikan akses kepada pengguna/pemilih untuk mengubah profil yaitu email, nama, nomor ponsel, NIM/NIK/NPM/NIS dan mengganti password
- Backend dapat mengelola proses pemungutan suara
- Backend dapat menyediakan fitur mendaftarkan kandidat
- Backend dapat menyediakan fitur mengelola timeline untuk mengatur waktu berjalannya pemungutan suara.
- Backend dapat menyediakan fitur mengelola pemilih.

- Backend dapat mengirimkan email maupun pesan Whatsapp berisi ajakan mengikuti/menjadi admin pemungutan suara
- Backend mendukung fitur menerima ajakan mengikuti/menjadi admin pemungutan suara
- Backend dapat mendukung pengiriman foto pengguna dan menyediakan data foto pemilih untuk verifikasi validitas suara yang masuk.
- Backend dapat mendukung sistem mengunduh hasil pemungutan suara
- Backend dapat mendukung sistem menghentikan pemungutan suara jika dalam keadaan darurat.
- Backend dapat mendukung sistem menghapus seluruh data suara yang masuk.
- Backend dapat mendukung sistem takedown pemungutan suara jika perlu.
- Backend dapat mendukung sistem yang dapat disesuaikan dan digunakan untuk berbagai organisasi *secara* bersamaan

4.1.2. Kebutuhan Non Fungsional

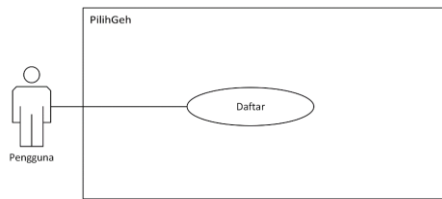
Berikut adalah kebutuhan non-fungsional dari situs web ini:

- Backend harus terhubung ke koneksi internet
- Backend dapat menjamin perlindungan data yang memadai dalam sisi kerahasiaan dan keamanan menggunakan enkripsi ataupun *hash*.
- Backend dapat memiliki skalabilitas yang tinggi.
- Backend dapat menahan serangan spam.
- Backend hanya dapat diakses oleh Frontend dari PilihGeh.

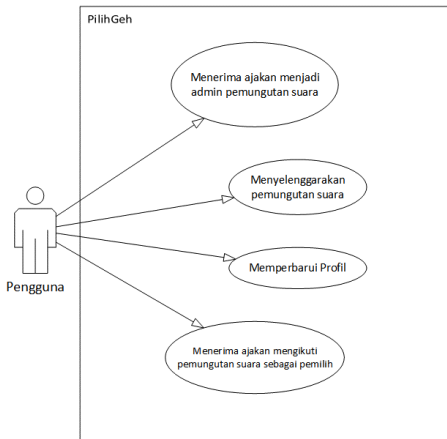
4.2. Desain Sistem

4.2.1. Use Case Diagram

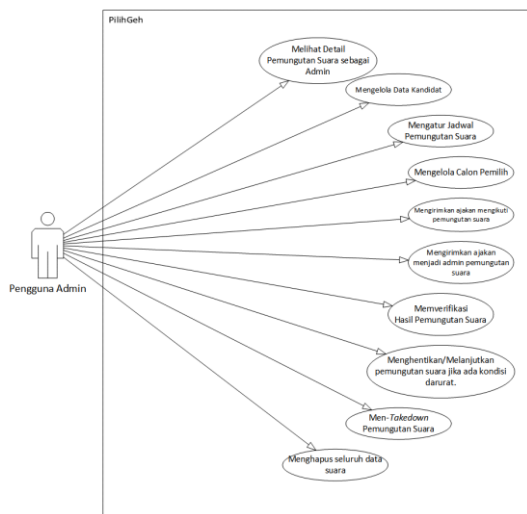
Use case diagram adalah sebuah diagram yang dapat digunakan untuk menggambarkan interaksi antara sistem dengan aktor yang terlibat dalam situs web ini. Use case diagram pada rancang bangun sistem ini dibagi menjadi 4 yaitu ketika user belum mendaftar, user belum login, user sesudah login dan user sebagai admin pemilihan. Berikut use case diagram dari PilihGeh :



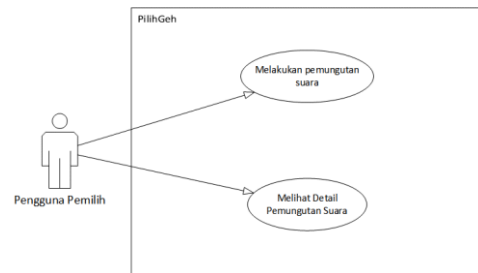
Gambar 3 Use Case Diagram Pengguna Sebelum Register



Gambar 4. Use Case Diagram Pengguna Setelah Login



Gambar 5 Use Case Diagram Pengguna Admin



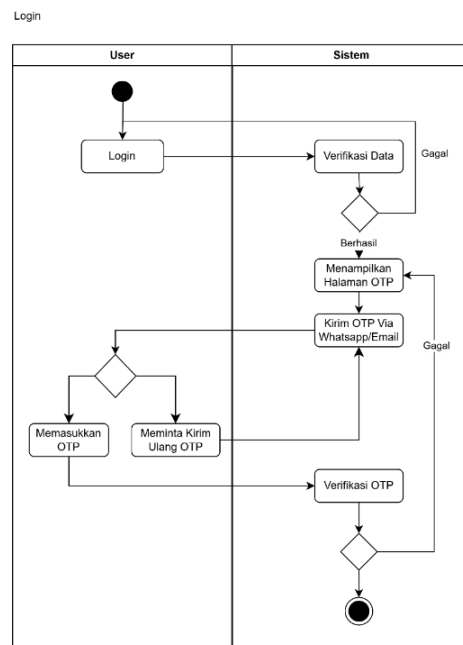
Gambar 6 Use Case Diagram Pengguna Pemilih

4.2.2. Activity Diagram

Activity diagram merupakan diagram yang menggambarkan alur kerja dari sebuah sistem. Diagram ini memudahkan untuk memahami proses pada sistem secara keseluruhan. Berikut beberapa activity diagram dari sistem pemilihan online PilihGeh:

a. Activity Diagram Login

Berikut adalah *activity diagram* aktivitas login pada situs web ini.

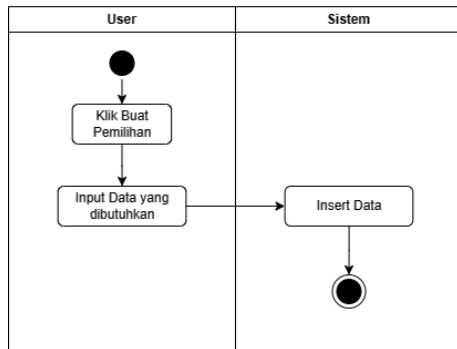


Gambar 7 Activity Diagram Login

b. Activity Diagram Menyelenggarakan Pemilihan

Berikut adalah *activity diagram* untuk menyelenggarakan pemilihan pada situs web ini.

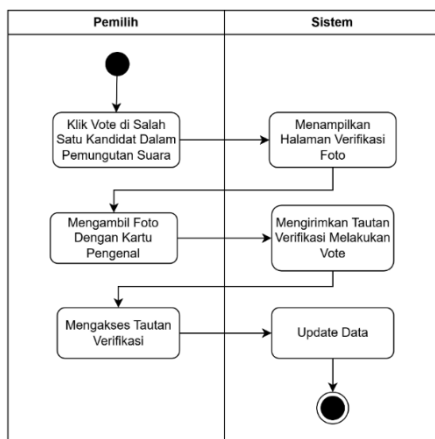
Menyelenggarakan Pemilihan



Gambar 8 Activity Diagram Menyelenggarakan Pemilihan

c. *Activity Diagram* Melaksanakan Pemilihan
Berikut adalah *activity diagram* untuk melaksanakan pemilihan yang sedang berlangsung pada situs web ini.

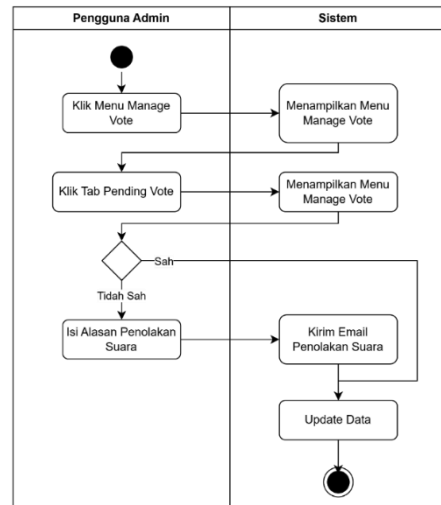
Melaksanakan Pemungutan Suara



Gambar 9 Activity Diagram Melaksanakan Pemilihan

d. *Activity Diagram* Memverifikasi Suara
Berikut adalah alur dari memverifikasi suara yang telah masuk pada situs web ini.

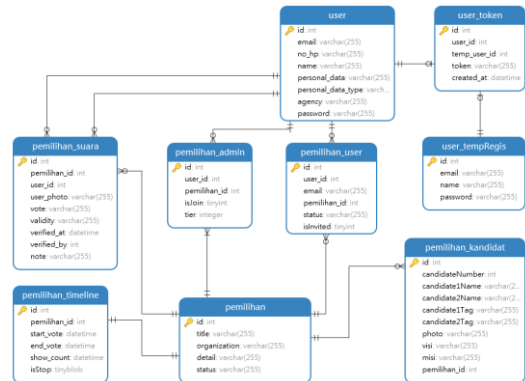
Verifikasi Suara



Gambar 10 Activity Diagram Memverifikasi Suara

4.2.3. Entity Relationship Diagram

Entity relationship diagram (ERD) merupakan diagram yang menggambarkan hubungan antar entitas dari suatu database. ERD terdiri dari 3 komponen yaitu entitas, atribut dan relasi.



Gambar 11 Entity Relationship Diagram Database PilihGeh

Berdasarkan gambar 11, ERD pada database PilihGeh berpusat pada dua entitas, yaitu user(pengguna) dan pemilihan dan satu entitas sementara yaitu user_tempRegis. Berikut jenis relasi dari tiap entitas ke entitas lainnya

Tabel 1 Relasi Dengan Referensi Field Dari Tabel User

Field	Table	Cardinality	Source	Relation
User_id	User_token	Zero or One	One and only one	1:1
	Pemilihan_admin	Zero or Many		1:M
	Pemilihan_user	Zero or Many		1:M
	Pemilihan_suara	Zero or Many		1:M
	Pemilihan_suara	Zero or Many		1:M

Berdasarkan Tabel 1, User_id memiliki hubungan 1:1 dengan User_token, 1:M dengan Pemilihan_admin dan Pemilihan_user, dan dua hubungan 1:M dengan Pemilihan_suara sebagai pengguna dan admin.

Tabel 2 Relasi Dengan Referensi Field Dari Tabel Pemilihan

Field	Table	Cardinality	Source	Relation
Pemilihan_id	Pemilihan_timeline	One and only one	One and only one	1:1
	Pemilihan_suara	Zero or Many		1:1
	Pemilihan_admin	One or Many		1:M
	Pemilihan_user	Zero or Many		1:M
	Pemilihan_kandidat	Zero or Many		1:M

Berdasarkan Tabel 2, Pemilihan_id memiliki hubungan 1:1 dengan Pemilihan_timeline dan Pemilihan_suara, serta hubungan 1:M dengan Pemilihan_admin, Pemilihan_user, dan Pemilihan_kandidat

Tabel 3 Relasi Dengan Referensi Field Dari Tabel User_tempRegis

Field	Table	Cardinality	Source	Relation
temp_user_id	User_token	Zero and One	One and only one	1:1

Berdasarkan Tabel 3, temp_user_id pada Tabel User_tempRegis di User_token memiliki hubungan 1:1

4.3. Pengodean (Coding)

Backend situs web PilihGeh menggunakan Express.js, memfasilitasi rute untuk autentikasi, pemungutan suara, profil pengguna, berinteraksi dengan database untuk menyimpan dan mengambil informasi, menerapkan fitur keamanan seperti enkripsi data, menangani kesalahan, dan menyediakan API untuk komunikasi dengan frontend, memungkinkan pertukaran data antara frontend dan backend secara efisien. Teknologi pendukung seperti middleware keamanan, manajemen sesi juga digunakan untuk menyempurnakan fungsi sistem.

- Fitur autentikasi pada backend PilihGeh dibagi menjadi dua, authenticationUser untuk memvalidasi token dan menyimpan info pengguna ketika pengguna ingin mengakses fitur verifikasi OTP, authenticationLogin untuk memeriksa validitas login sebelum akses ke *route* selanjutnya. Middleware yang terhubung dengan rute-rute di *authentication.js* memastikan hanya pengguna yang terotentikasi dan terverifikasi dapat mengakses fungsi-fungsi tertentu.
- Fitur pemungutan suara terbagi menjadi dua bagian: penyelenggaraan dan pelaksanaan. Fasilitas penyelenggaraan memungkinkan pembuatan pemungutan suara baru dengan pengisian data seperti nama pemungutan, nama organisasi, dan deskripsi. Di sisi lain, fitur pelaksanaan memungkinkan pemilih untuk memilih kandidat; diperlukan foto pemilih dan token verifikasi sebagai syarat untuk memberikan suara.
- Fitur kelola pemungutan suara sebagai admin dibagi menjadi tujuh fitur, di antaranya yaitu terdapat fitur CRUD kandidat, *timeline*, *manage voters*, *manage admin*, *manage vote*, dan *settings* untuk mengelola berjalannya pemungutan suara.
- Fitur kelola profil bertujuan untuk melakukan pembaruan informasi profil pengguna. Untuk melaksanakan fitur ini, dibutuhkan data pribadi pengguna, seperti nama, alamat email, nomor telepon, tipe

data pribadi, ID data pribadi, dan institusi terafiliasi.

Berikut adalah hasil implementasi dari pengodean yang sudah dilakukan yang merupakan dokumentasi dari masing-masing endpoint backend PilihGeh menggunakan Swagger:

All Users Endpoints available to all users		
POST	/users/register-no-device	Register a new user
POST	/users/login-no-device	Login a user without a device
POST	/users/verify-otp	Verify the OTP sent to the user
GET	/users/resend-otp	Resend the OTP to the user
GET	/users/get-profile	Get user profile
PATCH	/users/edit-profile	Edit a user's profile
PATCH	/users/edit-profile/verify/{token}	Verify a user's edited profile

Gambar 12 Dokumentasi Endpoint All User

Users Endpoints available to authenticated users		
POST	/users/create-pemilihan	Create a new pemilihan
GET	/users/get-pemilihan	Get a list of pemilihan that the user is following
GET	/users/pemilihan/{id}	Get a specific pemilihan by ID
GET	/users/get-kelola-pemilihan	Get a list of pemilihan that the user is managing
GET	/users/pemilihan/{id}/join/{token}	Join a pemilihan as a user
GET	/admin/pemilihan/{id}/join-admin/{token}	Join a pemilihan as an admin
POST	/users/pemilihan/{id}/add-vote/{no_kandidat}	Add a vote to a pemilihan

Gambar 13 Dokumentasi Endpoint Users

Pemilihan Data - Users Endpoints available to authenticated users for pemilihan data		
GET	/users/pemilihan/{id}/get-kandidat	Get a list of kandidat for a specific pemilihan by ID
GET	/users/pemilihan/{id}/get-kandidat/{no_kandidat}	Get a specific kandidat by candidate number for a specific pemilihan by ID
GET	/users/pemilihan/{id}/get-timeline	Get timeline for Pemilihan
GET	/users/pemilihan/{id}/count	Count vote

Gambar 14 Dokumentasi Endpoint Akses Users Ke Pemilihan

Kandidat - Admins Endpoints available to authenticated admins for kandidat		
GET	/admin/pemilihan/{id}/get-kandidat	Get a list of kandidat for a specific pemilihan by ID
GET	/admin/pemilihan/{id}/get-kandidat/{no_kandidat}	Get a specific kandidat by candidate number for a specific pemilihan by ID
POST	/admin/pemilihan/{id}/add-kandidat	Add a new kandidat to a pemilihan
PUT	/admin/pemilihan/{id}/add-edit-photo-kandidat/{no_kandidat}	Add or edit a photo for a kandidat in a pemilihan
PUT	/admin/pemilihan/{id}/edit-kandidat/{no_kandidat}	Edit kandidat
DELETE	/admin/pemilihan/{id}/delete-kandidat/{no_kandidat}	Delete kandidat by ID and no_kandidat

Gambar 15 Dokumentasi Endpoint Akses Admin Ke Data Kandidat

Timeline - Admins Endpoints available to authenticated admins for timeline		
GET	/admin/pemilihan/{id}/get-timeline	Get timeline for Pemilihan
POST	/admin/pemilihan/{id}/add-timeline	Add timeline
PUT	/admin/pemilihan/{id}/update-timeline	Update timeline by ID
DELETE	/admin/pemilihan/{id}/delete-timeline	Delete timeline by ID

Gambar 16 Dokumentasi Endpoint Akses Admin Ke Data Timeline

Manage User - Admins Endpoints available to authenticated admins for manage user		
GET	/admin/pemilihan/{id}/get-list	Get all list of users by ID
GET	/admin/pemilihan/{id}/get-user/{id_pemilihan_user}	Get user by ID and id_pemilihan_user
POST	/admin/pemilihan/{id}/add-user	Add user by email
POST	/admin/pemilihan/{id}/add-user-excel	Add users from excel file
DELETE	/admin/pemilihan/{id}/delete-user/{id_pemilihan_user}	Delete user by ID and id_pemilihan_user
GET	/admin/pemilihan/{id}/send-token-all	Send token to all users of Pemilihan
GET	/admin/pemilihan/{id}/send-token-pemilihan/{id_pemilihan_user}	Send token to user by ID and id_pemilihan_user

Gambar 17 Dokumentasi Endpoint Akses Admin Mengelola Data User(Pemilih)

Manage Pemilihan - Admins Endpoints available to authenticated admins for manage pemilihan		
POST	/admin/pemilihan/{id}/change-title	Change title of Pemilihan
DELETE	/admin/pemilihan/{id}/delete-all-votes	Delete all votes from Pemilihan
PUT	/admin/pemilihan/{id}/set-stop	Set status of Pemilihan to stop
PUT	/admin/pemilihan/{id}/set-emergency	Set emergency stop vote for Pemilihan

Gambar 18 Dokumentasi Endpoint Akses Admin Mengelola Pemilihan

Manage Vote - Admins Endpoints available to authenticated admins for manage vote		
GET	/admin/pemilihan/{id}/get-vote	Get all votes for Pemilihan
GET	/admin/pemilihan/{id}/get-vote/{vote_id}	Get specific vote by ID for Pemilihan
GET	/admin/pemilihan/{id}/pending-vote	Get all pending votes for Pemilihan
GET	/admin/pemilihan/{id}/verified-vote	Get all verified votes for Pemilihan
GET	/admin/pemilihan/{id}/invalid-vote	Get all invalid votes for Pemilihan
GET	/admin/pemilihan/{id}/count	Count votes for Pemilihan
PUT	/admin/pemilihan/{id}/verify-vote	Verify user's vote for Pemilihan

Gambar 19 Dokumentasi Endpoint Akses Admin Mengelola Suara Masuk

4.4. Testing

4.4.1. Pengujian Fungsionalitas

Pada tahap pengujian sistem ini, dilakukan pengujian langsung pada infrastruktur backend dengan metode blackbox testing. Tujuannya adalah memastikan bahwa sistem berfungsi sesuai kebutuhan yang telah ditentukan.

Sebanyak 132 skenario diuji untuk memverifikasi kesesuaian hasil fungsional sistem dengan yang diharapkan. Berikut beberapa hasil *blackbox testing* dari fitur utama backend :

a. Login

Tabel 4 Blackbox Testing Login

No	Kasus	Hasil Diharapkan		Hasil Pengujian	Status
		Status Code	JSON		
1	Email/Nomor Handphone tidak valid	400	"message": "Email/Nomor Handphone tidak valid"	Sesuai	Lulus
2	Salah satu data kosong	400	"message": "Email/Nomor Handphone/Password tidak boleh kosong"	Sesuai	Lulus
3	Email / Nomor Handphone tidak terdaftar	404	"message": "User Tidak Terdaftar!"	Sesuai	Lulus
4	Password tidak sesuai	401	"message": "Password Salah!"	Sesuai	Lulus
5	Email terdaftar dan password sesuai	200	"token": "jwt token"	Sesuai	Lulus

Berdasarkan Tabel 4, sistem berhasil mengembalikan respons yang tepat untuk berbagai skenario, termasuk input tidak valid atau kosong, user tidak terdaftar, password salah, dan autentikasi yang berhasil. Dalam semua kasus, sistem berfungsi sesuai harapan dan lulus pengujian.

b. Melaksanakan Pemungutan Suara

Tabel 5 Blackbox Testing Melaksanakan Pemungutan Suara

No	Kasus	Hasil Diharapkan		Hasil Pengujian	Status
		Status Code	JSON		
1	Kandidat dengan nomor yang diberikan dalam parameter tidak ditemukan	400	"message": "Kandidat tidak ditemukan!"	Sesuai	Lulus
2	User belum melengkapi data diri mereka	400	"message": "Anda belum melengkapi data diri!"	Sesuai	Lulus
3	User sudah melakukan vote	409	"message": "Anda sudah pernah melakukan vote!"	Sesuai	Lulus
4	Pemilihan dengan ID yang diberikan dalam parameter belum memiliki timeline	400	"message": "Pemilihan ini belum memiliki timeline!"	Sesuai	Lulus
5	Pemilihan sedang dihentikan sementara (Emergency Stop)	400	"message": "Pemilihan Sedang Dihentikan Sementara!(Emergency Stop)"	Sesuai	Lulus
6	Tanggal saat ini lebih awal dari tanggal mulai vote atau lebih akhir dari tanggal akhir vote	409	"message": "Vote belum dimulai!" atau "Vote sudah berakhir!"	Sesuai	Lulus
7	Tidak ada foto yang diunggah dalam request	400	"message": "Foto tidak boleh kosong!"	Sesuai	Lulus
8	Semua kondisi untuk melakukan vote terpenuhi, dan user berhasil melakukan vote	200	"message": "Berhasil melakukan vote!"	Sesuai	Lulus

Berdasarkan Tabel 5, sistem berhasil mengembalikan respons yang tepat untuk berbagai skenario, termasuk kandidat tidak ditemukan, user belum melengkapi data diri, user sudah melakukan vote, pemilihan belum memiliki timeline, pemilihan dihentikan

sementara, waktu vote belum dimulai atau sudah berakhir, dan tidak ada foto yang diunggah dalam request. Jika semua kondisi terpenuhi, user berhasil melakukan vote. Dalam semua kasus, sistem berfungsi sesuai harapan dan lulus pengujian.

c. Verifikasi Suara Masuk

Tabel 6 Blackbox Testing Verifikasi Suara Masuk

No	Kasus	Hasil Diharapkan		Hasil Pengujian	Status
		Status Code	JSON		
1	Jika validity bukan 0 atau 1	400	"message": "Validity harus 0 atau 1!"	Sesuai	Lulus
2	Jika validity adalah 0 dan note kosong	400	"message": "Note harus diisi!"	Sesuai	Lulus
3	Jika tidak ada suara dengan id tertentu, user_id, dan pemilihan_id	400	"message": "Tidak ada vote dengan id \${voteId}, user_id \${user_id} di pemilihan \${req.params.id}!"	Sesuai	Lulus
4	Jika suara ditemukan dan berhasil diverifikasi	200	"message": "Berhasil memverifikasi suara!"	Sesuai	Lulus

Berdasarkan Tabel 6, sistem verifikasi suara menangani kasus-kasus seperti validity tidak valid, note kosong, suara tidak ditemukan, dan suara berhasil diverifikasi. Semua skenario telah lulus pengujian.

4.5. Deployment

Backend PilihGeh saat ini sudah siap digunakan pada frontend PilihGeh untuk mendukung penyimpanan dan pemrosesan data pemilihan yang dapat digunakan oleh masyarakat seperti pelajar maupun masyarakat umum. Backend PilihGeh ini sudah dilakukan deployment di layanan Railway <https://backend-pilihgeh-production.up.railway.app>

5. KESIMPULAN

Berdasarkan hasil dan rancang bangun yang telah disampaikan, didapatkan kesimpulan sebagai berikut:

1. Rancang bangun yang dilakukan berhasil mengembangkan backend untuk sistem PilihGeh yang mendukung pemilihan secara fleksibel dengan menggunakan metode iterative development model dan menggunakan library ExpressJs.
2. Rancang bangun yang dilakukan berhasil menyediakan fitur yang sudah direncanakan seperti fitur autentikasi, otorisasi, mengelola profil, mengelola pemilihan, maupun melaksanakan pemilihan.
3. Backend web sudah diuji dalam aspek fungsionalitas dengan menggunakan metode

blackbox testing dengan menguji 132 skenario ketika mengakses endpoint dengan hasil sesuai dengan yang diharapkan.

4. Backend PilihGeh telah berhasil di deployment dengan menggunakan layanan Railway dengan tautan <https://backend-pilihgeh-production.up.railway.app/>.

DAFTAR PUSTAKA

- [1] A. Azis, I. Setiawan dan A. Risqiantoro, "Aplikasi E-Voting Untuk Pemilihan Kepala Desa Berbasis Website," *Jurnal Of Information System Management*, vol. 1, no. 1, pp. 1-7, 2019.
- [2] R. Prananda, H. Anra dan H. S. Pratiwi, "Rancang Bangun Aplikasi E-Voting Berbasis Android (Studi Kasus : Pemilihan Ketua Organisasi di Lingkungan Fakultas Teknik Universitas Tanjungpura)," *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, vol. 5, no. 1, pp. 17-21, 2017.
- [3] A. T. Siahaan dan H. N. Irmada, "Aplikasi Sistem e-Voting Ketua Umum UPN Band Veteran Jakarta Berbasis Website," *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya*, vol. 2, no. 2, pp. 742-751, 2021.
- [4] R. B. G. Maulida, "Perancangan Sistem Pemilihan Gubernur Mahasiswa Dengan Metode Electronic Voting Berbasis Website," *Teknik Informatika, Universitas Magelang, Magelang*, 2021.
- [5] J. Pratama dan R. Rachman, "Rancang Bangun Aplikasi E-Voting Ketua Organisasi Siswa Intra Sekolah (OSIS) Menggunakan Framework Laravel Berbasis Website (Studi Kasus: SMPN 39 Bandung)," *Jurnal Nasional Komputasi dan Teknologi Informasi*, vol. 5, no. 4, pp. 714-718, 2022.
- [6] D. Irawan dan Z. Novianto, "Perancangan E-Learning Pada SMAN 1 Kota Lubuklinggau Menggunakan Framework Codeigniter (CI)," *Jurnal Digital Teknologo Informasi*, vol. 3, no. 2, pp. 53-60, 2020.
- [7] R. Fajrin, "Pengembangan Sistem Informasi Geografis Berbasis Node.JS untuk Pemetaan Mesin dan Tracking Engineer dengan Pemanfaatan Geolocation pada PT IBM Indonesia," *Jurnal Komputer Terapan*, vol. 3, no. 1, pp. 33-40, 2017.
- [8] S. Munir, A. R. Adriansyah, R. Dhelika dan R. Wahyudi, "Implementasi Arsitektur Aplikasi MVC Pada Perancangan Aplikasi Bank Sampah Berbasis Web," *Jurnal Teknologi Terpadu*, vol. 4, no. 2, pp. 76-81, 2018.
- [9] M. I. Perkasa dan E. B. Setiawan, "Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Access Token," *ULTIMA Computing*, vol. X, no. 1, pp. 19-26, 2018.
- [10] B. Setiono, T. B. Adji dan D. Adhipta, "Komparasi Kinerja Basis Data Oracle dan PostgreSQL Berdasarkan Estimasi Cost dan Execution Time Serta TPC-C Benchmark," *Universitas Gadjah Mada, Yogyakarta*.
- [11] A. Mubarak, "Rancang Bangun Aplikasi Web Sekolah Menggunakan UML (Unified Modeling Language) dan Bahasa Pemrograman PHP(PHP Hypertext Preprocessor) Berorientasi Objek," *Jurnal Informatika dan Komputer*, vol. 2, no. 1, pp. 19-25, 2019.
- [12] N. M. A. Munassar dan A. Govardhan, "A Comparison Between Five Models Of Software Engineering," *International Journal of Computer Science Issues*, vol. 7, no. 5, pp. 94-101, 2010.