

BAB 3

METODE PENELITIAN

Penelitian ini merupakan penelitian rancang-bangun yang bertujuan untuk mengembangkan sistem *e-voting* berbasis web dengan metode *Waterfall*. Sistem ini dikembangkan untuk memberikan solusi pemungutan suara yang dapat membantu mahasiswa untuk memudahkan pelaksanaan Pemira dan juga aman dari manipulasi data dibandingkan metode penyewaan sistem *e-voting* yang mungkin memiliki keterbatasan dalam kontrol, biaya, dan keamanan data.

Dibandingkan dengan penyewaan sistem *e-voting stwawpoll*, pendekatan pengembangan sendiri memungkinkan:

1. Kustomisasi penuh terhadap fitur yang diinginkan tanpa ketergantungan pada vendor seperti menyimpan data mahasiswa kedalam *database e-voting*.
2. Keamanan lebih terjamin, karena sistem dikembangkan dengan standar keamanan yang dapat dikontrol langsung dengan mengurangi terjadinya manipulasi data dengan masuknya *email* yang tidak diinginkan.
3. Mengurangi terjadinya mahasiswa tidak memilih karena tidak mendapatkan akses untuk memilih atau tidak mendapatkan token pemilihan.
4. Kemampuan peningkatan sistem secara mandiri, sesuai dengan kebutuhan organisasi di masa depan seperti dibutuhkannya tempat untuk informasi para calon untuk mahasiswa.

Sistem ini akan menggunakan *One-Time Password* (OTP) atau token unik yang dikirimkan melalui email kepada pemilih. Token ini akan terisi otomatis ketika pemilih membuka tautan yang dikirim, dan akan hangus setelah satu kali pemakaian, sehingga setiap pemilih hanya dapat memilih satu kali.

3.1 BAHAN DAN ALAT PENELITIAN

Bahan yang digunakan dalam penelitian ini mencakup berbagai informasi dan sumber daya yang mendukung pengembangan sistem, antara lain:

1. Data Pemilih

Berisi daftar pemilih yang berhak mengikuti proses *e-voting*, termasuk email yang digunakan untuk mengirimkan OTP/token.

2. Data Calon

Meliputi informasi lengkap calon ketua dan wakil ketua organisasi mahasiswa, termasuk nama, NIM, visi-misi, serta program kerja yang diunggah sebagai bahan pertimbangan pemilih.

3. Data Pemungutan Suara

Berisi hasil pemilihan yang dilakukan oleh masing-masing pemilih. Data ini diolah untuk proses rekapitulasi, penghitungan suara otomatis, serta pelaporan hasil Pemira.

4. Data Waktu dan Jadwal Pemilihan

Digunakan untuk mengatur waktu mulai dan berakhirnya proses pemungutan suara serta menampilkan waktu hitung mundur dan pengumuman waktu secara otomatis dalam sistem.

5. Data Panitia Pemira

Digunakan untuk autentikasi akses admin atau panitia dalam mengelola sistem, seperti input data calon, pemilih, monitoring hasil suara, serta pengelolaan akun pengguna.

Adapun alat penelitian yang relevan dengan penelitian ini yaitu:

Perangkat Keras:

1. Laptop/PC dengan spesifikasi prosesor *Intel Core i5* ke atas, RAM 8GB, dan SSD 256GB.
2. *Server Cloud/VPS* untuk hosting aplikasi *e-voting*.
3. Koneksi Internet stabil untuk pengujian sistem berbasis web.

Perangkat Lunak:

1. Sistem Operasi: Windows 10 atau di atasnya.
2. Backend: Python (Django) untuk *server-side processing*.
3. Frontend: HTML, CSS, *JavaScript* untuk antarmuka pengguna.

4. *Database*: MySQL untuk menyimpan data pemilih dan hasil pemungutan suara.
5. Keamanan:
 - a) *JSON Web Token (JWT)* untuk manajemen sesi pengguna.
 - b) OTP Generator menggunakan fungsi dari `django.utils.crypto` yaitu `get_random_string()`.
6. Alat Pendukung Pengembangan:
 - a) XAMPP untuk menjalankan server lokal.
 - b) Visual Studio Code sebagai *code editor*.

Sistem operasi dan program-program aplikasi yang dipergunakan dalam dalam pengembangan aplikasi ini adalah:

1. Sistem Operasi: Windows 10 atau lebih baru (digunakan sebagai lingkungan pengembangan lokal).
2. Framework Backend: Python 3.x dengan Django Framework sebagai inti pengembangan aplikasi web.
3. *Database Engine*: MySQL 8.0 atau versi terbaru sebagai sistem manajemen basis data relasional untuk menyimpan data pemilih, kandidat, token voting, dan hasil suara.
4. SQL Manager: phpMyAdmin atau MySQL Workbench sebagai alat bantu manajemen *database* secara visual.
5. Frontend Development: HTML5, CSS3, dan JavaScript. Framework CSS opsional seperti Bootstrap 5 untuk tampilan responsif.
6. Tools Pendukung: Visual Studio Code sebagai code editor. Git dan GitHub/GitLab untuk version control. Dan Virtualenv untuk mengisolasi environment Python.

3.2 JALAN PENELITIAN

Penelitian ini menggunakan metode pengembangan perangkat lunak *Waterfall*. Model air terjun (*Waterfal Modell*) adalah suatu pendekatan klasik dalam pengembangan perangkat lunak yang bersifat linier dan bertahap. Keseluruhan fase ini menggambarkan siklus hidup dari perangkat lunak mulai dari tahap awal hingga

tahap pengiriman produk akhir. Setiap fase harus diselesaikan terlebih dahulu sebelum melangkah ke fase berikutnya, di mana hasil dari satu fase menjadi dasar atau input untuk fase berikutnya secara berurutan (Hasanah & Untari, 2020). Model ini dipilih karena cocok diterapkan dalam proyek pengembangan aplikasi individu, seperti tugas akhir atau skripsi, di mana tahapan-tahapan dapat dilakukan satu per satu secara terstruktur.

Secara garis besar, penelitian ini dilakukan dalam 5 tahap utama, yaitu:

1. Requirements

Tahap pertama dalam penelitian ini adalah mengidentifikasi kebutuhan dari sistem *e-voting* yang akan dikembangkan. Proses analisis kebutuhan akan dilakukan melalui wawancara langsung dengan panitia pelaksana Pemira serta beberapa mahasiswa sebagai calon pengguna sistem. Selain itu, peneliti juga menganalisis sistem *e-voting* sebelumnya, termasuk kelemahan dari aplikasi *Strawpoll* yang sempat digunakan dalam Pemira sebelumnya. Hasil yang diharapkan dari proses ini adalah perumusan kebutuhan fungsional seperti sistem login, pengiriman token OTP melalui email, fitur pemilihan calon, dan rekap hasil suara. Kebutuhan non-fungsional juga diidentifikasi, termasuk keharusan sistem yang aman, mudah digunakan, dan transparan. Semua informasi ini kemudian dirangkum ke dalam dokumen kebutuhan sistem (*System Requirement Specification*) yang akan menjadi dasar dalam tahap desain sistem.

2. Design

Tahapan desain dilakukan dengan mengacu pada hasil kebutuhan sistem dari tahap sebelumnya. Proses ini dimulai dengan merancang struktur basis data menggunakan MySQL, termasuk perancangan tabel untuk data pemilih, data calon, suara yang masuk, serta token OTP. Selanjutnya, peneliti menyusun arsitektur sistem berbasis web client-server yang menggambarkan alur komunikasi antara pengguna dan server. Untuk memvisualisasikan alur kerja dan fungsionalitas sistem, digunakan berbagai diagram UML seperti *Use case diagram* untuk menggambarkan interaksi pengguna dengan sistem, *Activity diagram* untuk menggambarkan alur pemungutan suara, dan

Sequence Diagram untuk alur pengiriman OTP hingga proses *voting*. Selain itu, antarmuka pengguna juga dirancang menggunakan *mockup UI/UX* sederhana agar sistem lebih mudah dipahami pengguna. Hasil akhir dari tahap ini berupa dokumen desain sistem yang mencakup rancangan teknis dan tampilan visual, yang akan digunakan sebagai panduan dalam tahap pengkodean.

3. *Coding*

Tahap implementasi atau pengkodean dimulai dengan mengkonversi desain sistem ke dalam bentuk kode program. Backend sistem dibangun menggunakan Django sebagai *framework* utama, dengan bahasa pemrograman Python, sementara frontend dikembangkan menggunakan HTML, CSS, dan JavaScript untuk menghasilkan tampilan web yang responsif. Peneliti juga mengintegrasikan sistem autentikasi berbasis OTP, yang akan mengirimkan kode ke email pemilih yang terdaftar agar dapat melakukan login dan memberikan suara. Seluruh logika pemrosesan, validasi, dan penyimpanan data dikembangkan sesuai alur yang telah dirancang pada tahap sebelumnya. Pada akhir tahap ini, diharapkan menghasilkan aplikasi *e-voting* berbasis web yang dapat diakses pengguna untuk melakukan simulasi Pemira secara digital.

4. *Pengujian Sistem*

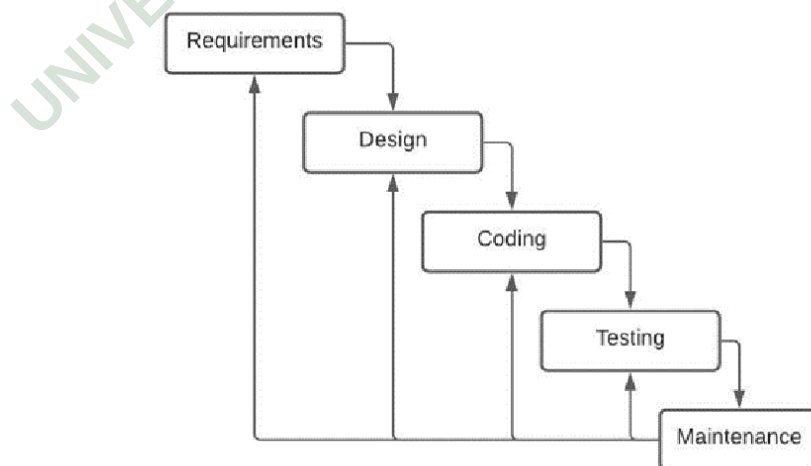
Setelah sistem selesai dikembangkan, tahap selanjutnya adalah melakukan pengujian. Pengujian dilakukan oleh peneliti secara langsung melalui *unit testing* untuk memastikan setiap komponen sistem, seperti pengiriman OTP, pemrosesan suara, dan penyimpanan data, berfungsi sesuai tujuan. Setelah itu, *integration testing* dilakukan untuk menguji integrasi antar modul dan memastikan tidak ada konflik ketika sistem dijalankan secara keseluruhan. Untuk memastikan sistem dapat diterima oleh pengguna akhir, peneliti juga melaksanakan *User Acceptance Testing (UAT)* dengan melibatkan sekitar 5 hingga 10 orang mahasiswa dan panitia Pemira. *UAT* ini bertujuan mengevaluasi kemudahan penggunaan, keamanan proses, serta respon pengguna terhadap sistem secara menyeluruh. *Feedback* dari

pengguna selama pengujian ini digunakan untuk melakukan perbaikan terhadap sistem sebelum masuk ke tahap implementasi akhir.

5. Implementasi dan Evaluasi

Tahap terakhir adalah implementasi sistem dalam skala kecil dan evaluasi. Sistem yang telah melewati tahap pengujian kemudian di-*deploy* pada server lokal atau *cloud* dan digunakan oleh sekitar 50 hingga 60 pengguna dalam simulasi Pemira internal. Dalam tahap ini, peneliti memantau performa sistem secara langsung, termasuk kestabilan server, kecepatan pengiriman OTP, dan keakuratan rekapitulasi suara. Evaluasi dilakukan berdasarkan pengamatan performa sistem selama digunakan dan melalui kuesioner feedback dari pengguna. Hasil evaluasi ini digunakan untuk mengetahui sejauh mana sistem memenuhi kebutuhan pengguna dan sejauh mana sistem sudah siap untuk digunakan dalam Pemira sesungguhnya. Jika masih ditemukan kekurangan, sistem akan diperbaiki dan diperbarui sesuai saran dan kebutuhan yang muncul dari proses ini. Tahapan *maintenance* ini menjadi penutup dari siklus *waterfall* dan dapat dilanjutkan dengan iterasi baru jika sistem akan dikembangkan lebih lanjut di masa depan.

Penerapan siklus metode *waterfall* tahap demi tahap dapat diilustrasikan pada Gambar 3.1.



Gambar 3.1 Siklus Metode *Waterfall*

3.3 IDENTIFIKASI KEBUTUHAN

Identifikasi kebutuhan dilakukan untuk menentukan secara rinci fitur-fitur dan komponen yang harus dimiliki oleh sistem *e-voting* yang akan dikembangkan. Kebutuhan ini dibagi menjadi dua kategori utama dan satu kebutuhan tambahan, yaitu kebutuhan fungsional, kebutuhan non-fungsional, dan kebutuhan AI dan keamanan tambahan berdasarkan hasil observasi, wawancara dengan panitia Pemira, dan analisis terhadap sistem sebelumnya seperti *Strawpoll* dan *Google Form*.

3.3.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan fitur utama yang harus dimiliki oleh sistem agar proses *e-voting* dapat berjalan dengan baik dan sesuai dengan prosedur Pemira. Tabel 3.1 menunjukkan kebutuhan fungsional dari sistem yang akan dibuat.

Tabel 3.1 Tabel Fungsional Sistem

No.	Aktor	Fitur / Kebutuhan Sistem	Deskripsi Fungsional
1	Admin, Pemilih	Autentikasi Login	Admin masuk ke sistem menggunakan NIM dan <i>password</i>
2	Admin	Manajemen Pemilihan	CRUD data pemilihan, menentukan jadwal mulai dan selesai.
3	Admin	Input Data Pemilih	Mengimpor data pemilih dari file Excel dan CRUD secara manual
4	Admin	Atur Pemilih per Pemilihan	Menentukan siapa saja yang berhak memilih untuk tiap pemilihan tertentu.
5	Admin	Manajemen Kandidat	Input ketua & wakil berdasarkan NIM, unggah visi-misi dan program kerja.
6	Admin	Kirim Token Voting	Sistem mengirim token <i>voting</i> via <i>email</i> secara otomatis.
7	Admin	Monitoring Suara Masuk	Melihat status suara terkumpul, total pemilih.

8	Admin, Pemilih	Lihat Hasil <i>Voting</i>	Menampilkan grafik hasil pemilihan secara <i>real-time</i> .
9	Admin	Deteksi Suara Mencurigakan	Menampilkan log suara mencurigakan untuk analisis kecurangan.
10	Admin	Ekspor Data	Ekspor hasil <i>voting</i> dan log suara mencurigakan ke CSV / gambar.
11	Pemilih	Autentikasi via Token	Request Token dan masuk ke halaman <i>voting</i> menggunakan token yang dikirim ke email.
12	Pemilih	Melihat Daftar Kandidat	Melihat informasi ketua & wakil, visi, misi, dan program kerja.
13	Pemilih	Memberikan Suara	Memilih satu pasangan kandidat dan mengirimkan suara.
14	Pemilih	Melihat Status Pemilihan	Melihat apakah sudah memilih atau belum, serta waktu mulai & selesai <i>voting</i> .

3.3.2 Kebutuhan *Non-Fungsional*

Kebutuhan non-fungsional berkaitan dengan kualitas dan karakteristik sistem agar dapat berjalan secara optimal. Tabel 3.2 menunjukkan kebutuhan *non-fungsional* dari sistem yang akan dibuat.

Tabel 3.2 Tabel *Non-Fungsional* Sistem

No.	Kategori	Kebutuhan Non-Fungsional	Deskripsi
1	Usability	Antarmuka Mudah Digunakan	Sistem memiliki desain antarmuka yang sederhana, responsif, dan ramah pengguna.
2	Security	One-Time Token <i>Voting</i>	Token hanya bisa digunakan satu kali oleh satu pemilih untuk satu pemilihan.
3	Security	Enkripsi Data	Data suara, <i>password</i> , dan token dienkripsi untuk menjaga kerahasiaan dan integritas data.

4	Performance	Respon Cepat	Sistem merespon input pengguna dengan waktu < 3 detik dalam kondisi normal.
5	Availability	Akses 24/7 Selama Pemilihan	Sistem harus selalu online selama masa Pemira berlangsung agar bisa diakses kapan saja.
6	Compatibility	Dukungan Multi-Perangkat	Sistem dapat diakses dari berbagai perangkat (laptop, PC, smartphone).
7	Maintainability	Mudah Dipelihara dan Ditingkatkan	Kode sistem modular, terdokumentasi, dan mudah dikembangkan lebih lanjut.
8	Scalability	Skalabilitas untuk Banyak Pengguna	Sistem mampu menangani ratusan pemilih secara bersamaan tanpa crash atau delay.
9	Auditability	Log Aktivitas	Sistem menyimpan log aktivitas penting untuk keperluan audit dan deteksi kecurangan.
10	Reliability	Rekam Suara Otomatis	Setiap suara disimpan langsung dan validasi suara dilakukan secara otomatis oleh sistem.

3.3.3 Kebutuhan AI dan Keamanan Tambahan

Menjabarkan fitur-fitur yang berkaitan dengan penerapan kecerdasan buatan (AI) dan mekanisme keamanan tambahan dalam sistem *e-voting* untuk mendeteksi dan mencegah kecurangan, serta meningkatkan integritas sistem. Tabel 3.3 menunjukkan kebutuhan AI dan Keamanan Tambahan dari sistem yang akan dibuat.

Tabel 3.3 Tabel Kebutuhan AI dan Keamanan Tambahan

No.	Komponen	Fitur / Kebutuhan Khusus	Deskripsi
1	AI / Machine Learning	Deteksi Suara Mencurigakan	Sistem memanfaatkan model klasifikasi (misalnya Random Forest) untuk mendeteksi pola suara anomali.

2	AI / Rule-Based	Aturan Deteksi Kecurangan	Sistem mendeteksi kecurangan berbasis aturan, seperti IP berbeda, waktu tidak wajar, perangkat ganda.
3	AI / Dataset	Ekspor Data untuk Pelatihan Model	Admin dapat mengekspor suara mencurigakan ke format CSV untuk pelatihan model AI secara berkelanjutan.
4	Logging	Pencatatan Log <i>Voting</i>	Sistem mencatat log <i>voting</i> : NIM, IP Address, waktu <i>voting</i> , user-agent, lokasi (jika memungkinkan).
5	Token Security	Token Sekali Pakai (One-Time Use)	Token dikodekan acak, otomatis kadaluarsa setelah digunakan, tidak bisa dipakai ulang.
6	Token Delivery	Enkripsi Token Saat Pengiriman	Token dikirim melalui email dengan ID yang dienkripsi dan divalidasi saat link diklik.
7	<i>Voting Integrity</i>	Verifikasi Identitas Saat <i>Voting</i>	<i>Voting</i> hanya bisa dilakukan oleh pemilih terverifikasi berdasarkan NIM yang terdaftar.
8	<i>Database Security</i>	Hashing dan Enkripsi Suara	Suara yang tersimpan dalam <i>database</i> dienkripsi dan diverifikasi menggunakan hashing (misal SHA-256).
9	System Monitoring	Dashboard Anomali Suara	Admin dapat memantau log dan daftar suara mencurigakan melalui halaman monitoring khusus.
10	Recovery	Backup Otomatis Data <i>Voting</i>	Sistem secara berkala menyimpan backup data <i>voting</i> untuk keperluan pemulihan jika terjadi error.

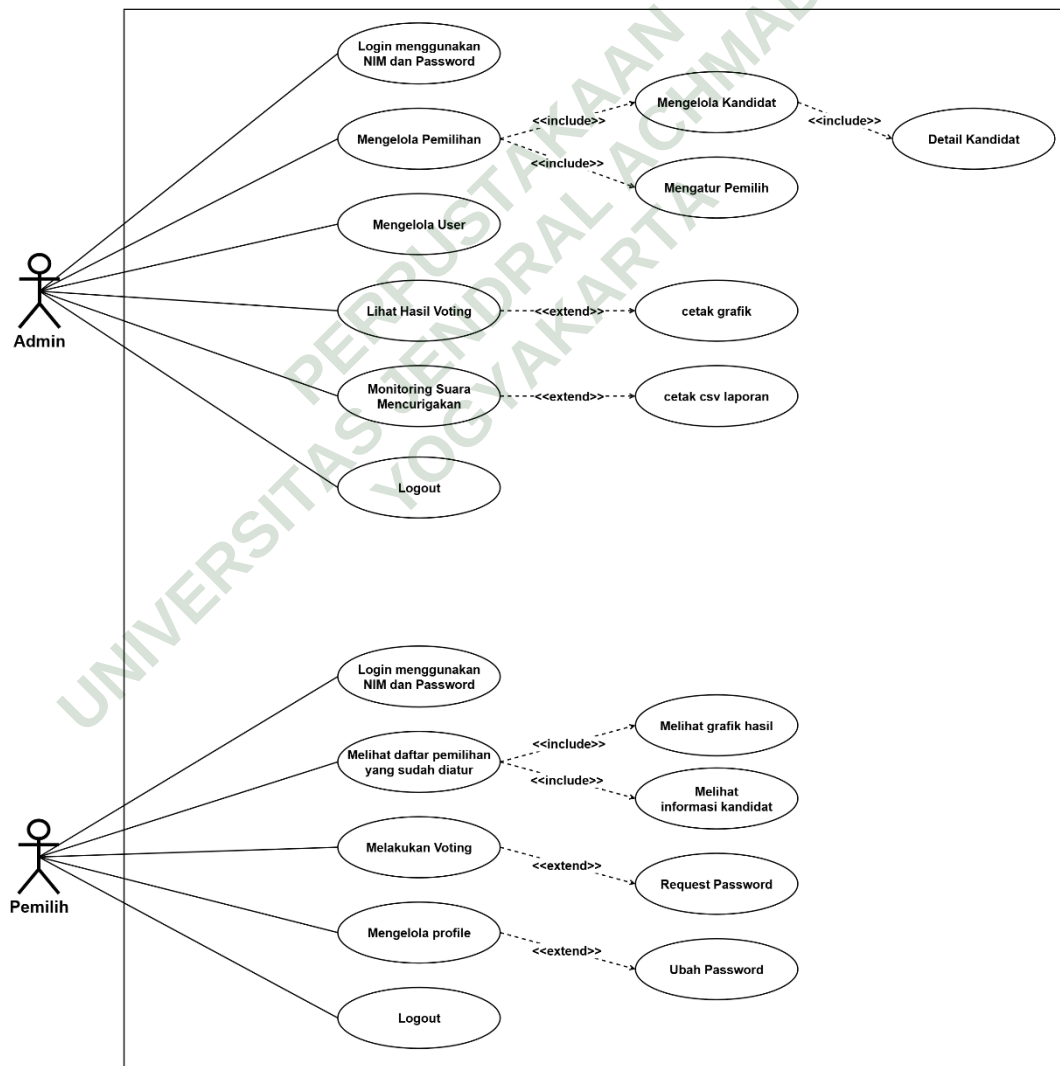
3.4 DESAIN SISTEM

Desain sistem adalah tahap penting dalam pengembangan perangkat lunak yang bertujuan untuk menggambarkan bagaimana sistem akan dibangun berdasarkan kebutuhan yang telah diidentifikasi. Pada penelitian ini, desain sistem mencakup arsitektur sistem, *use case diagram*, *activity diagram*, perancangan

struktur basis data serta lainnya yang akan digunakan dalam implementasi aplikasi *e-voting*. Berikut merupakan perancangan desain sistem aplikasi *e-voting* pemira Unjaya :

3.4.1 Use case diagram

Use case diagram aplikasi *e-voting* Pemira ini menggambarkan interaksi antara aktor (admin dan pemilih) dengan fitur utama sistem. Diagram ini menjadi dasar dalam pengembangan alur proses dan antarmuka pengguna sesuai kebutuhan fungsional yang telah ditentukan sebelumnya. *Use case diagram* akan ditampilkan pada gambar 3.2.



Gambar 3.2 *Use case diagram*

Uraian dari rangkaian aktivitas yang dilakukan oleh para aktor pada *use case* dikemas dalam tabel 3.4.

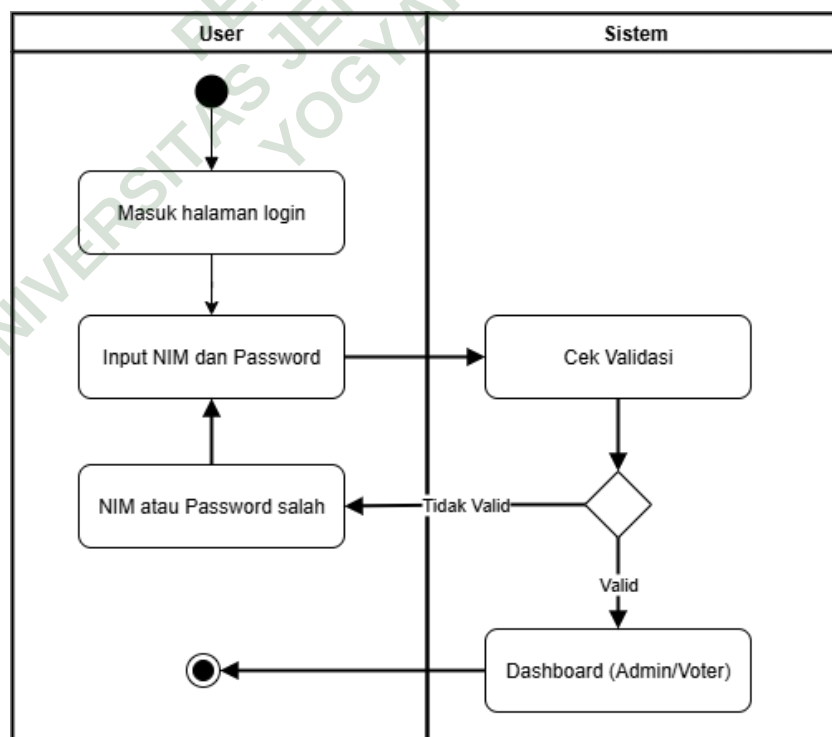
Tabel 3.4 *Uraian Use case diagram*

No.	Aktor	Use Case	Penjelasan
1	Admin, Pemilih	Login menggunakan NIM dan <i>Password</i>	Masuk kedalam sistem menggunakan NIM dan <i>Password</i> yang sudah terenkripsi.
2	Admin	Mengelola Pemilihan	Admin dapat melihat, membuat, mengatur, dan mengedit data pemilihan.
3	Admin	Mengelola Kandidat	Fitur untuk melihat, menambah atau mengedit pasangan calon ketua & wakil.
4	Admin	Detail Kandidat	Menampilkan informasi lengkap kandidat.
5	Admin	Mengatur Pemilih	Menentukan siapa saja mahasiswa yang bisa memilih.
6	Admin	Mengelola User	Manajemen data user (akun admin dan pemilih).
7	Admin	Lihat Hasil <i>Voting</i>	Melihat rekap suara secara <i>real-time</i> .
8	Admin	Cetak Grafik	Fitur tambahan untuk mencetak hasil dalam bentuk grafik.
9	Admin	Monitoring Suara Mencurigakan	Meninjau log suara yang dianggap anomali.
10	Admin	Cetak CSV Laporan	Ekspor laporan suara mencurigakan.
11	Admin	Logout	Keluar dari sistem.
12	Pemilih	Melihat daftar pemilihan yang sudah diatur	Menampilkan pemilihan yang dapat diikuti.
13	Pemilih	Melihat grafik hasil	Akses untuk melihat hasil sementara dari pemilihan.
14	Pemilih	Melihat informasi kandidat	Melihat profil, visi, misi dan program kerja pasangan calon .

15	Pemilih	Melakukan <i>Voting</i>	Fitur utama untuk memberikan suara pada kandidat pilihan.
16	Pemilih	Request <i>Password</i>	Permintaan token untuk dapat memilih .
17	Pemilih	Mengelola profil	Mengubah data akun pribadi.
18	Pemilih	Ubah <i>Password</i>	Mengubah <i>password</i> .
19	Admin, Pemilih	Logout	Keluar dari sistem.

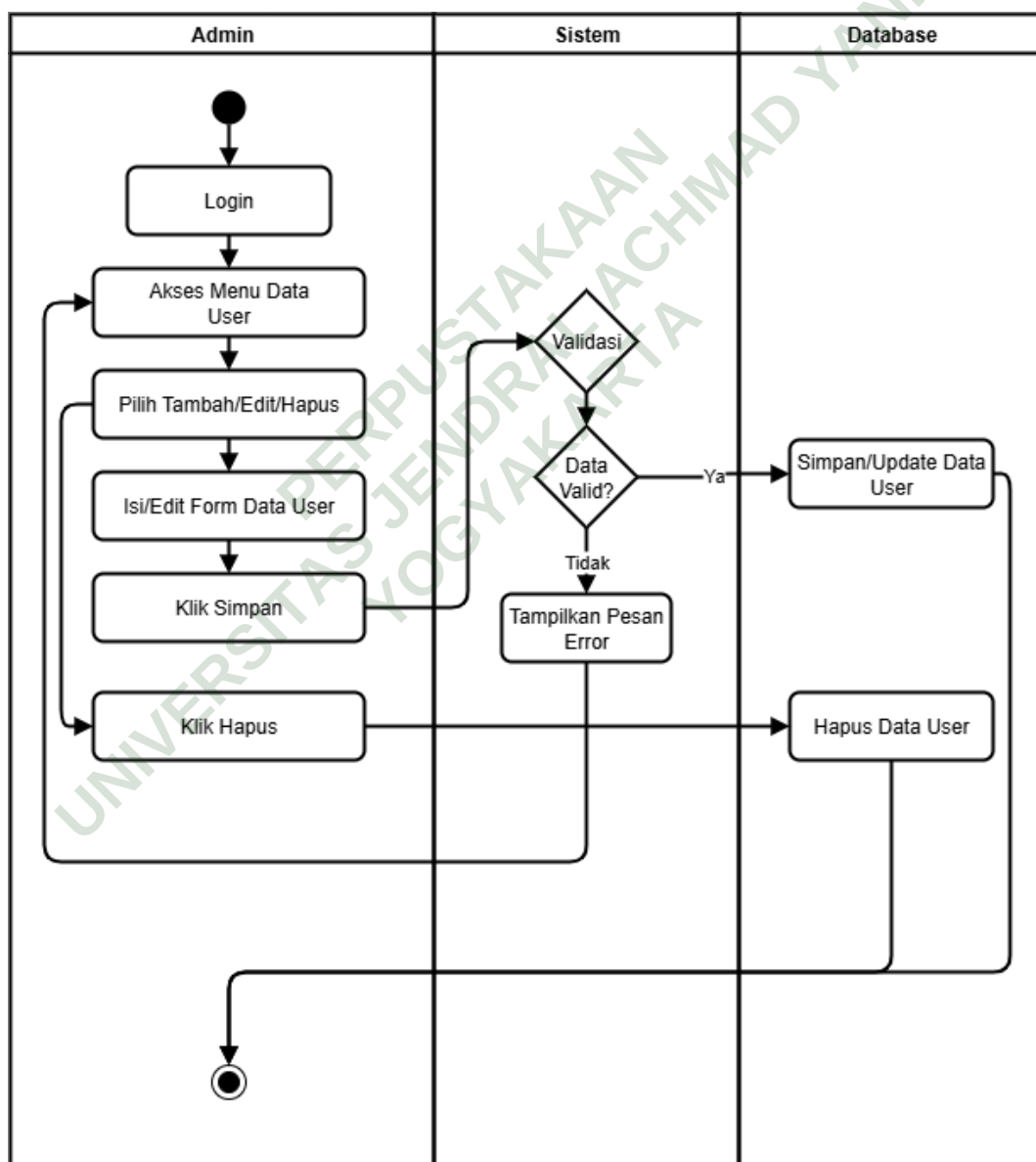
3.4.2 Activity diagram

Activity diagram digunakan untuk menggambarkan alur proses dalam sistem secara terperinci dari sudut pandang pengguna. Diagram ini menunjukkan urutan aktivitas dan keputusan yang terjadi selama proses interaksi pengguna dengan sistem, baik oleh Admin maupun Pemilih. Dengan memodelkan aktivitas dalam bentuk diagram ini, pengembangan sistem menjadi lebih terarah dan memudahkan implementasi alur fungsional dalam aplikasi *e-voting*. Berikut merupakan *activity diagram* untuk sistem aplikasi *e-voting*.



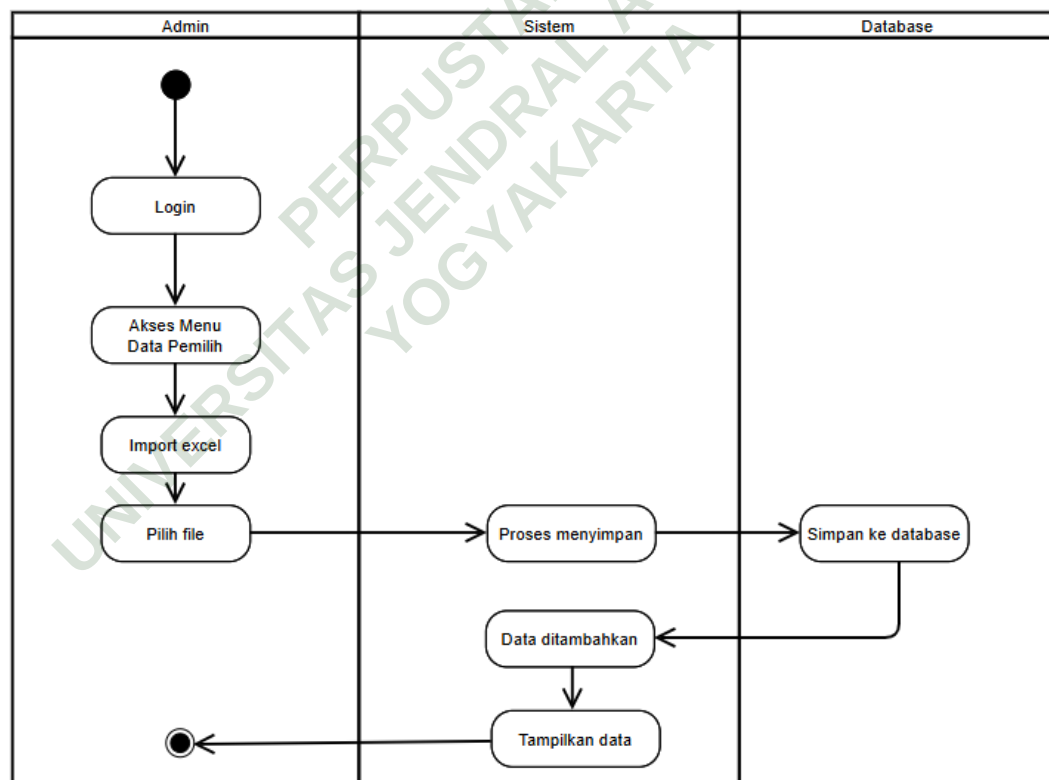
Gambar 3.3 Activity diagram Login User

Activity diagram pada gambar 3.3 menunjukkan alur proses login pengguna (Admin atau *Voter*) ke dalam sistem *e-voting*. Pengguna memulai dengan memasukkan NIM dan *password*, lalu sistem melakukan proses validasi. Jika data yang dimasukkan tidak sesuai, sistem menampilkan pesan kesalahan "NIM atau *Password* salah" dan pengguna diminta mengulangi proses login. Namun jika validasi berhasil, pengguna langsung diarahkan ke halaman dashboard sesuai dengan perannya.



Gambar 3.4 Activity diagram Kelola User

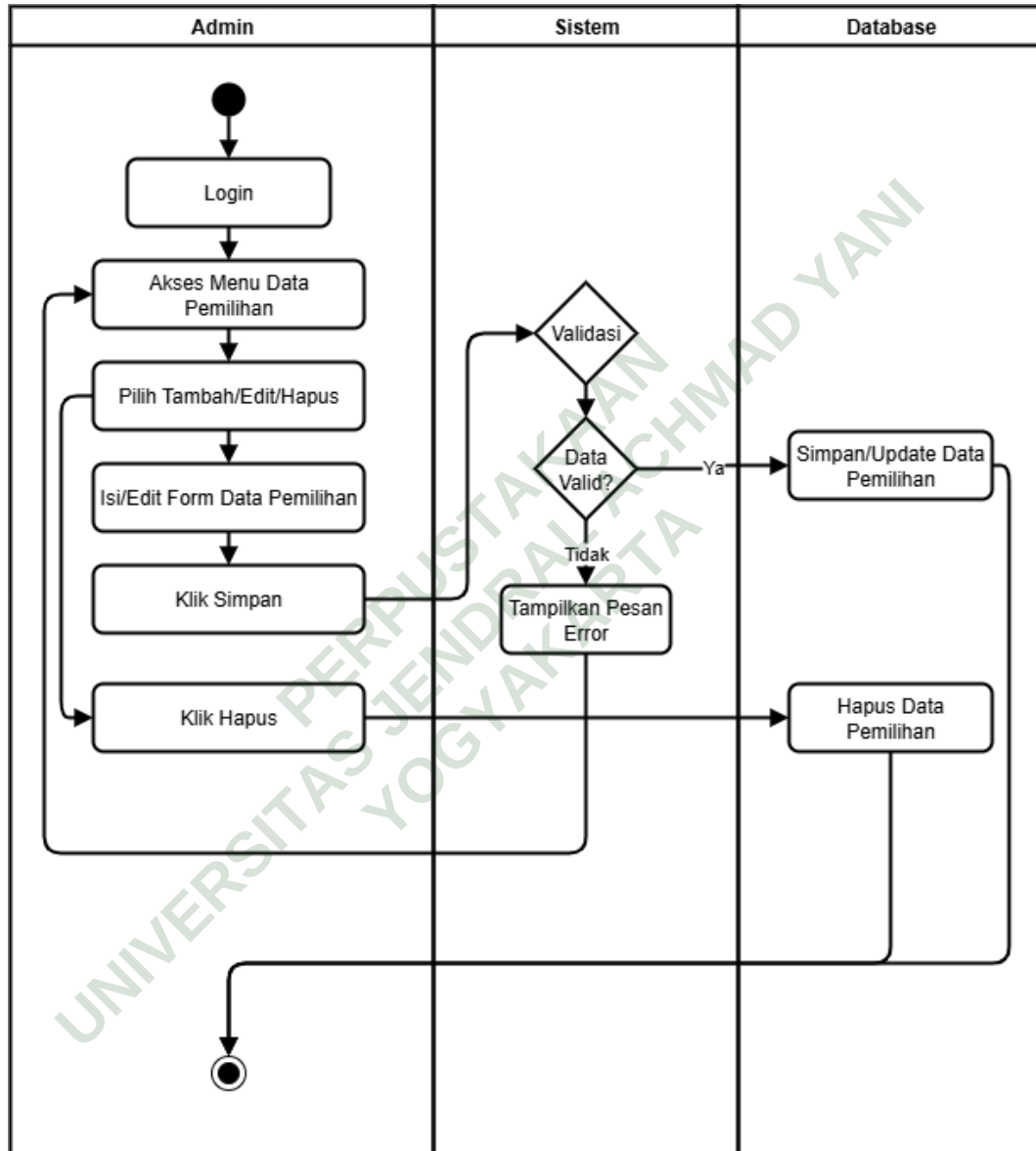
Activity diagram pada gambar 3.4 menunjukkan alur proses pengelolaan data user oleh Admin pada sistem. Admin memulai dengan login ke dalam sistem, kemudian mengakses menu pengelolaan data user. Selanjutnya, Admin memilih tindakan yang ingin dilakukan, yaitu tambah, edit, atau hapus data. Jika Admin memilih tambah atau edit, Admin akan mengisi atau mengubah form data user, kemudian mengklik simpan. Sistem kemudian melakukan validasi data. Apabila data tidak valid, sistem akan menampilkan pesan *error*, dan Admin diminta untuk memperbaiki data tersebut. Namun, jika data sudah valid, sistem akan menyimpan atau memperbarui data tersebut ke dalam *database*. Sedangkan jika Admin memilih untuk menghapus data user, sistem akan langsung menghapus data tersebut dari *database*. Setelah semua aktivitas selesai, proses kembali ke menu utama pengelolaan data atau Admin dapat mengakhiri proses.



Gambar 3.5 *Activity diagram Import Data User*

Activity diagram pada gambar 3.5 menggambarkan proses import data pemilih oleh Admin, dimulai dari login, mengakses menu data pemilih, melakukan

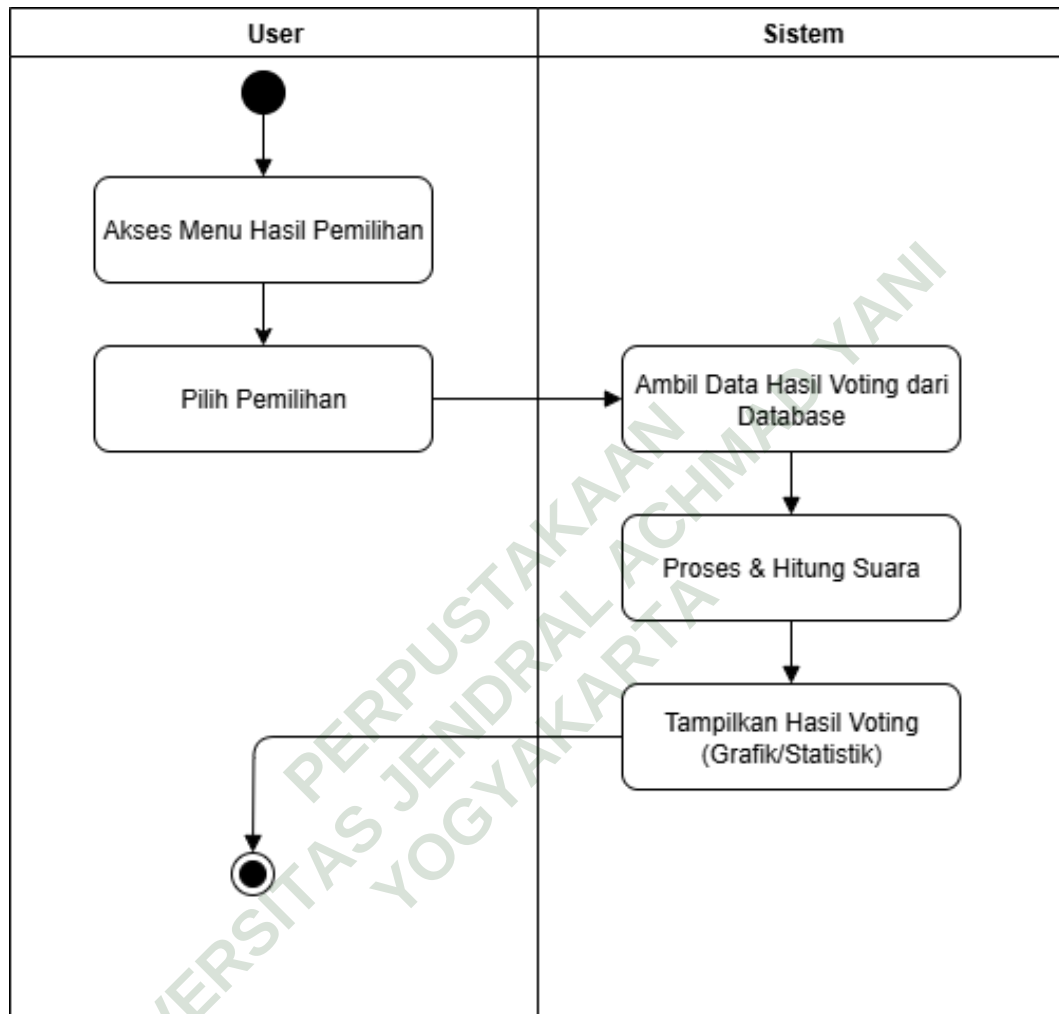
import file excel, hingga proses penyimpanan data ke database. Setelah data tersimpan, sistem akan menampilkan data yang baru ditambahkan, memastikan proses *import* berjalan dengan baik dan data tersedia dalam sistem.



Gambar 3.6 Activity diagram Kelola Pemilihan

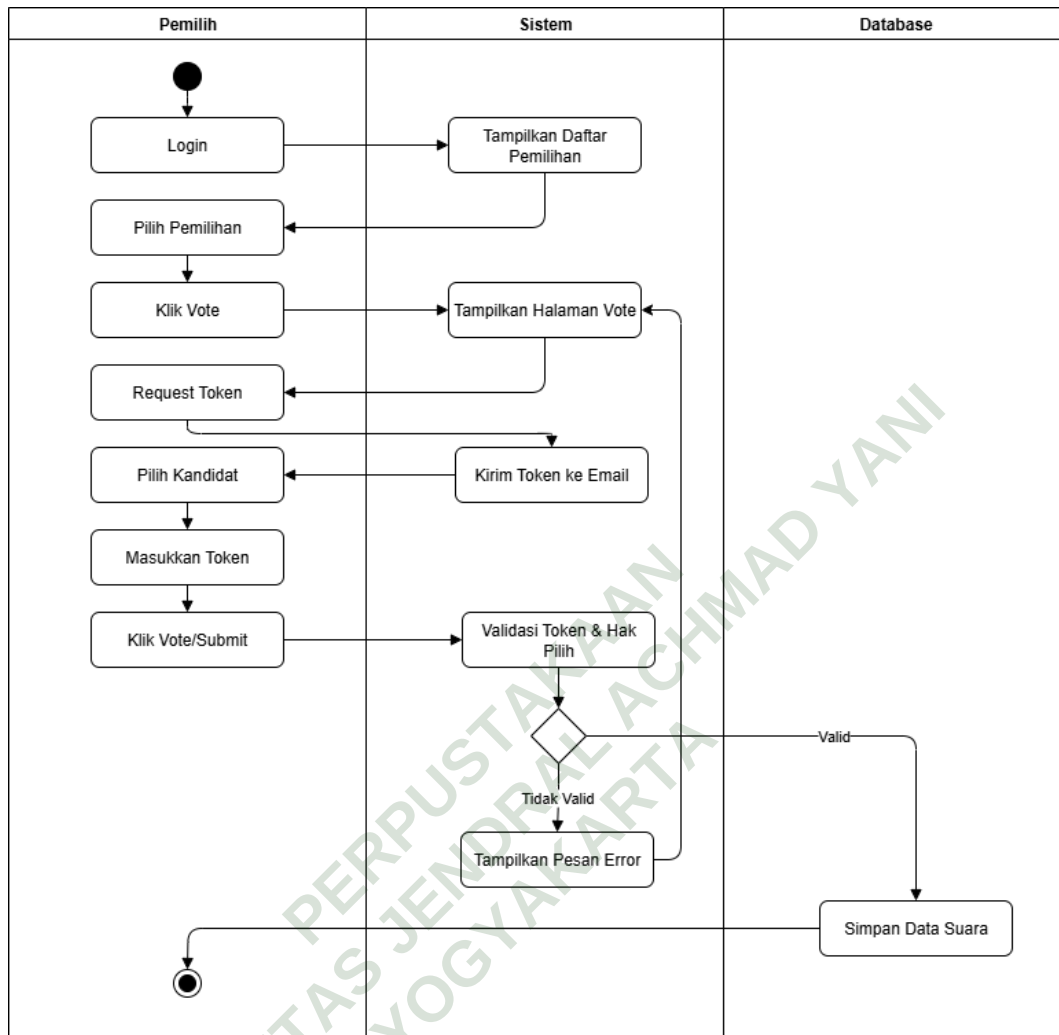
Activity diagram pada gambar 3.6 ini menggambarkan proses lengkap pengelolaan data pemilihan oleh admin, mulai dari login, akses menu, memilih aksi (tambah, edit, hapus), validasi data, hingga proses penyimpanan atau penghapusan data di *database*. Diagram ini memastikan bahwa setiap perubahan data pemilihan

dilakukan secara terstruktur dan divalidasi dengan baik oleh sistem sebelum data benar-benar masuk ke *database*.



Gambar 3.7 Activity diagram hasil Pemilihan

Activity diagram pada gambar 3.7 ini menjelaskan proses melihat hasil pemilihan oleh admin. Admin mengakses menu hasil pemilihan, kemudian memilih pemilihan yang ingin dilihat hasilnya. Sistem lalu mengambil data hasil *voting* dari database, melakukan perhitungan suara, dan menampilkan hasil *voting* dalam bentuk grafik atau statistik. Diagram ini memastikan admin mendapatkan informasi hasil pemilihan secara akurat dan terstruktur.



Gambar 3.8 Activity diagram Voting

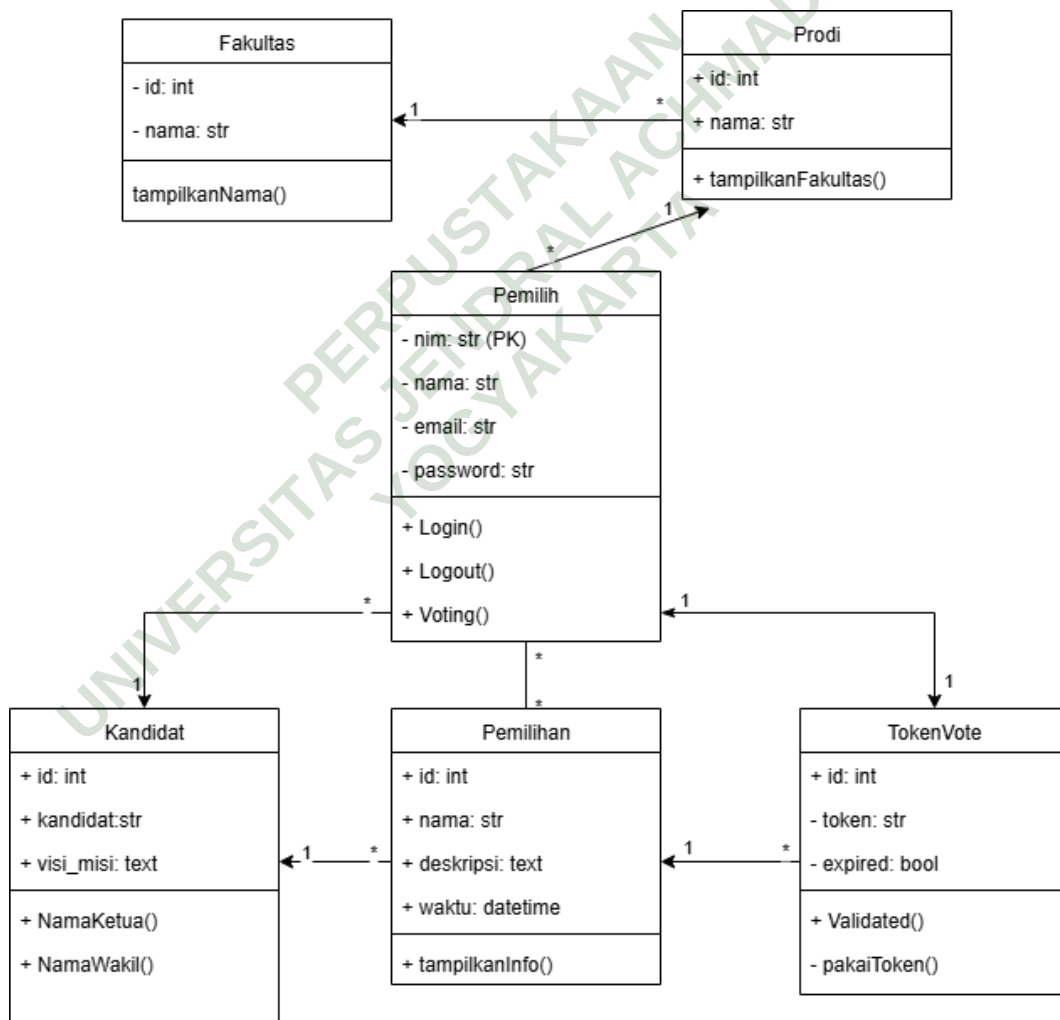
Activity diagram pada gambar 3.8 menggambarkan alur proses pemungutan suara (*voting*) oleh user dalam sistem *e-voting* mulai dari login, memilih pemilihan, meminta token, memilih kandidat, validasi token hingga penyimpanan suara di *database*. Penggunaan token sebagai autentikasi bertujuan untuk menjaga keamanan dan mencegah kecurangan dalam proses *voting*. Jika terjadi kesalahan pada token atau hak pilih, sistem akan memberikan pesan error agar user dapat memperbaiki prosesnya.

3.4.3 Class diagram

Class diagram merupakan salah satu bagian dari *Unified Modeling Language* (UML) yang digunakan untuk memodelkan struktur sistem secara statis,

yaitu dengan menggambarkan kelas-kelas utama beserta atribut, method, dan relasi antar kelas dalam aplikasi. Pada sistem *e-voting* Pemira Unjaya, *class diagram* dirancang untuk merepresentasikan struktur data dan hubungan antar entitas utama seperti Fakultas, Prodi, Pemilih, Kandidat, Pemilihan, dan TokenVote.

Dengan adanya *class diagram* ini, pengembangan sistem dapat dilakukan secara lebih terstruktur dan terorganisir, sehingga proses implementasi maupun pengembangan lanjutan menjadi lebih mudah dan sistematis. Dapat dilihat pada gambar 3.9 dibawah merupakan presentasi dari *class diagram e-voting*.



Gambar 3.9 *Class diagram E-voting*

3.5 DESAIN INTERFACE

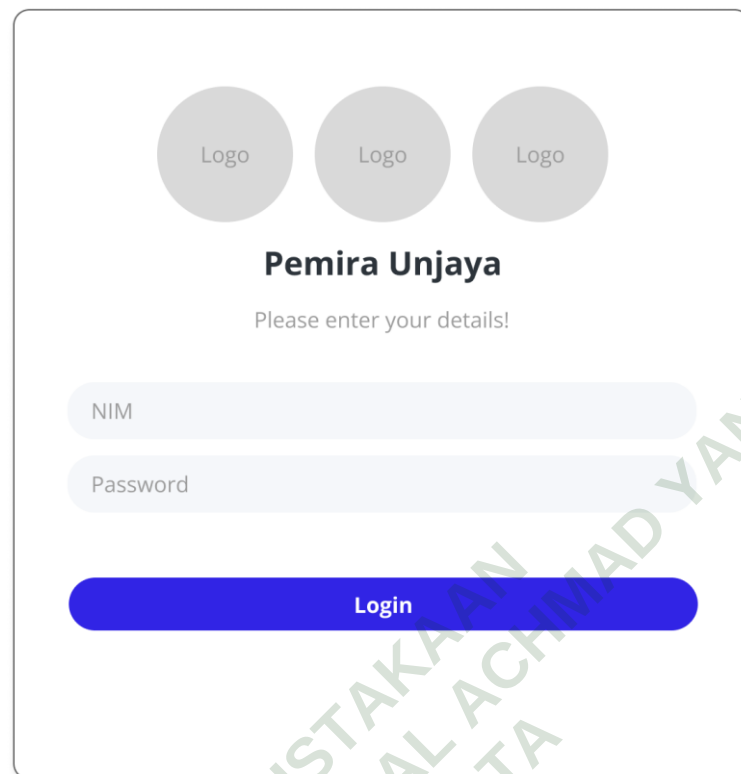
Desain interface atau perancangan antarmuka pengguna (*user interface/UI*) merupakan tahap penting dalam pengembangan sistem *e-voting* Pemira Unjaya. Tujuan dari tahap ini adalah menghasilkan tampilan aplikasi yang mudah digunakan, informatif, dan sesuai dengan kebutuhan pengguna, baik admin maupun pemilih. Antarmuka yang dirancang harus mendukung prinsip *usability*, yaitu kemudahan akses, konsistensi tampilan, navigasi yang jelas, serta responsif pada berbagai perangkat.

Desain interface sistem ini meliputi halaman utama (dashboard), halaman login, halaman pengelolaan data (pemilih, kandidat, pemilihan), halaman *voting* untuk pemilih, serta tampilan hasil pemilihan. Setiap elemen antarmuka disesuaikan agar pengguna dapat dengan mudah memahami alur aplikasi dan melaksanakan fungsinya tanpa kebingungan.

Dalam tahap ini, mockup atau wireframe digunakan sebagai gambaran awal sebelum pengembangan antarmuka sebenarnya. Desain interface yang baik akan meningkatkan pengalaman pengguna (*user experience/UX*) dan memastikan sistem dapat digunakan secara efektif dan efisien oleh seluruh pengguna.

3.5.1 Desain Halaman Login

Gambar 3.10 memperlihatkan desain antarmuka halaman login Pemira Unjaya yang sederhana dan *user-friendly*. Pada bagian atas terdapat placeholder logo institusi, diikuti judul aplikasi dan petunjuk singkat bagi pengguna. Terdapat dua *field* input untuk NIM dan *password*, serta tombol login berwarna biru yang jelas terlihat. Tata letak ini dirancang agar mudah digunakan baik di desktop maupun perangkat mobile.

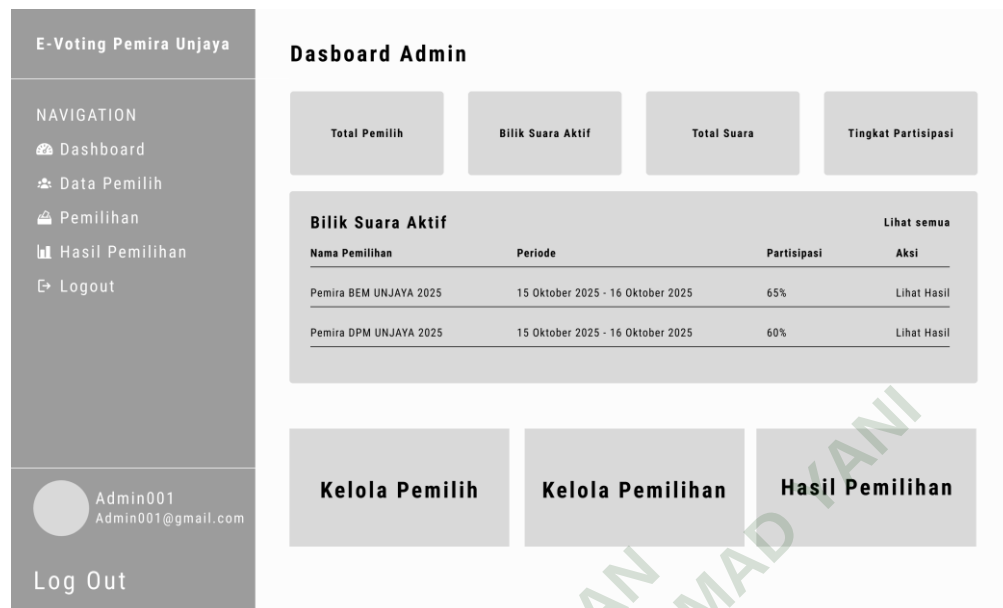


The image shows a user login page design for 'Pemira Unjaya'. At the top, there are three gray circular placeholders, each labeled 'Logo'. Below them is the title 'Pemira Unjaya' in a bold, dark font, followed by the instruction 'Please enter your details!'. The form consists of two light gray input fields: the first is labeled 'NIM' and the second is labeled 'Password'. Below these fields is a prominent blue button with the text 'Login' in white. A large, diagonal watermark reading 'UNIVERSITAS JENDERAL ACHMAD YANI' is overlaid across the center of the page.

Gambar 3.10 Desain Halaman *Login User*

3.5.2 Halaman Dashboard Admin

Pada Gambar 3.11 menampilkan desain antarmuka dashboard admin untuk sistem *e-voting* Pemira Unjaya. Layout terdiri dari sidebar navigasi di sebelah kiri dengan menu utama seperti Dashboard, Data Pemilih, Pemilihan, dan Hasil Pemilihan. Di bagian utama, terdapat ringkasan data statistik (total pemilih, pemilihan, suara, partisipasi) serta tabel daftar pemilihan aktif. Fitur akses cepat untuk mengelola user, pemilihan, dan hasil juga tersedia agar admin dapat memantau dan mengelola sistem secara efisien.



Gambar 3.11 Desain Halaman Dashboard Admin

3.5.3 Desain Halaman Data Pemilih

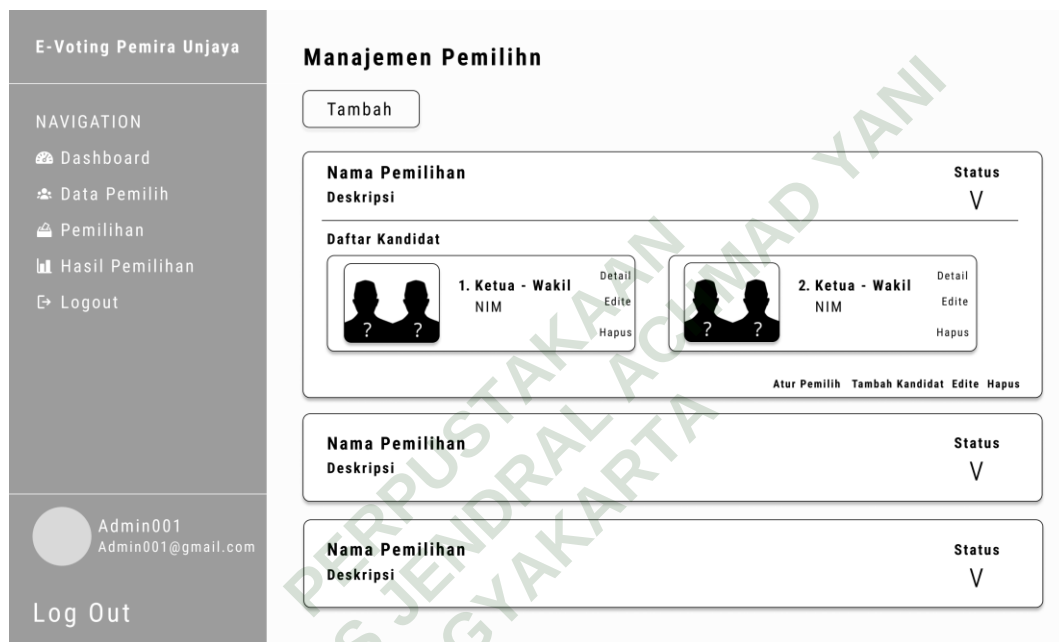
Pada gambar 3.12 memperlihatkan rancangan antarmuka untuk pengelolaan data pemilih oleh admin. Halaman ini menampilkan tabel daftar pemilih yang dapat dicari, ditambah, atau diimpor melalui tombol khusus. Kolom-kolom tabel meliputi NIM, Nama, Email, Prodi, Fakultas, dan aksi. Tampilan ini mendukung admin dalam mengelola data pemilih secara efisien dan terorganisir.



Gambar 3.12 Desain Halaman Data Pemilih

3.5.4 Desain Halaman Manajemen Pemilihan

Pada Gambar 3.13 menampilkan desain halaman manajemen pemilihan, di mana admin dapat melihat, menambah, atau mengelola daftar pemilihan beserta statusnya. Pada setiap pemilihan, admin dapat mengelola kandidat, mengatur pemilih, dan melakukan aksi detail, edit, maupun hapus. Desain ini memudahkan admin dalam mengelola proses pemilihan secara terstruktur dan interaktif.



Gambar 3.13 Desain Halaman Manajemen Pemilihan

3.5.5 Desain Modal Detail Kandidat

Gambar 3.14 menampilkan desain halaman detail kandidat yang berisi informasi lengkap tentang pasangan calon, termasuk nama, NIM, prodi, dan fakultas ketua serta wakil. Halaman ini juga menyediakan ruang untuk menampilkan visi, misi, dan program kerja (proker) kandidat secara terperinci. Tampilan ini memudahkan pemilih maupun admin dalam mengenal profil setiap kandidat sebelum melakukan *voting*.

Detail kandidat

X

2



Ketua : Nama Ketua (NIM)
Prodi - Fakultas

Wakil : Nama Wakil (NIM)
Prodi - Fakultas

Visi:

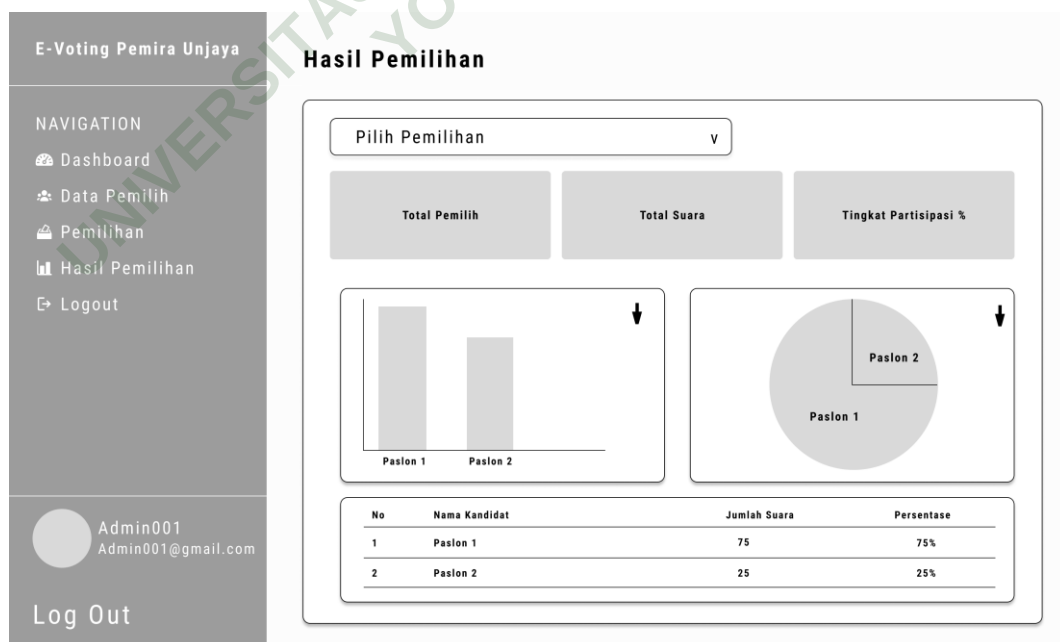
Proker:

Misi:

Gambar 3.14 Desain Modal Detail Kandidat

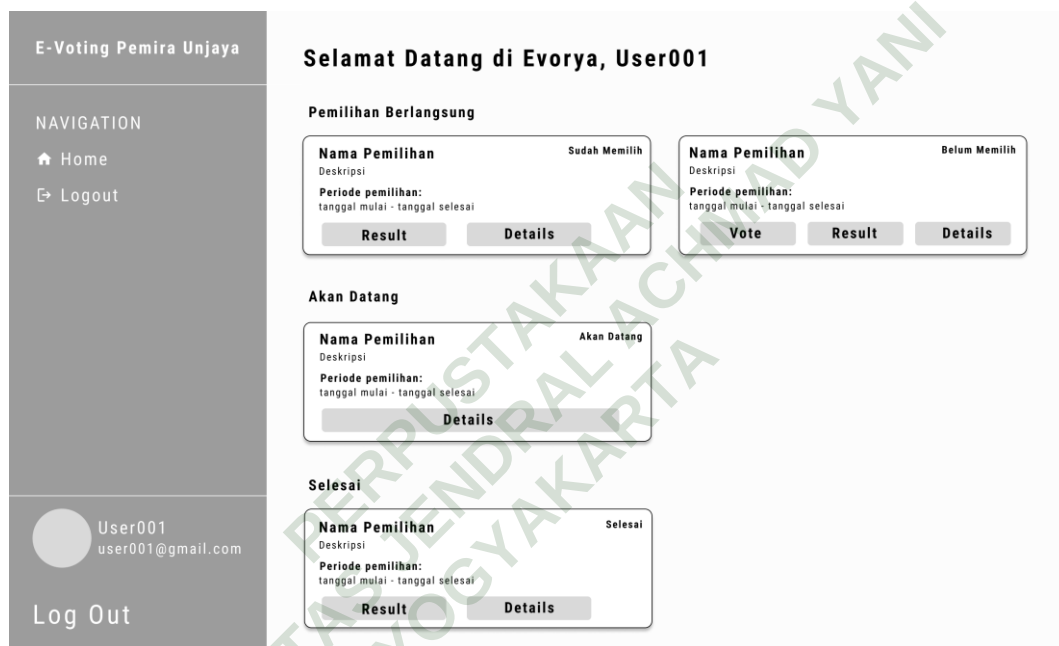
3.5.6 Desain Hasil Pemilihan

Pada Gambar 3.15 memperlihatkan desain halaman hasil pemilihan yang menampilkan rekapitulasi suara untuk setiap pemilihan. Terdapat fitur pemilihan periode, informasi total pemilih, total suara masuk, serta tingkat partisipasi. Visualisasi hasil disajikan dalam bentuk grafik batang dan diagram pie, dilengkapi dengan tabel rekap jumlah suara dan persentase untuk masing-masing kandidat. Tampilan ini memudahkan admin dalam memantau hasil *voting* secara *realtime*.



3.5.7 Desain Halaman Dashboard Voter

Pada Gambar 3.16 menampilkan desain halaman utama bagi pengguna setelah login ke aplikasi *e-voting*. Terdapat daftar pemilihan yang dikelompokkan berdasarkan status berlangsung, akan datang, atau sudah selesai dengan penjelasan singkat dan periode waktu masing-masing. Pengguna dapat melihat detail, hasil, atau melakukan *voting* jika statusnya aktif. Antarmuka ini memudahkan pengguna untuk memantau dan berpartisipasi dalam berbagai pemilihan secara efisien.



Gambar 3.16 Desain Halaman *Dashboard Voter*

3.5.8 Desain Details *Voting* dan Result

Pada Gambar 3.17 menunjukkan tampilan detail pemilihan yang berisi informasi periode, deskripsi, dan daftar pasangan kandidat secara lengkap. Setiap kandidat menampilkan foto, nama ketua dan wakil beserta prodi dan fakultas, serta visi, misi, dan program kerja. Pada bagian bawah, tersedia tombol “Vote” untuk pemilihan aktif dan belum memilih dipilihan tersebut.

Dan pada Gambar 3.18 menampilkan antarmuka hasil pemilihan yang memperlihatkan total suara masuk, total pemilih terdaftar, dan tingkat partisipasi dalam satu pemilihan. Data visualisasi hasil suara disajikan menggunakan diagram lingkaran (*pie chart*) untuk membandingkan perolehan suara antar pasangan calon

secara jelas dan informatif. Tampilan ini memudahkan admin maupun pengguna melihat hasil pemilihan secara ringkas dan *real-time*.

Detail Pemilihan: Nama Pemilihan X

Periode : waktu mulai - waktu selesai

Deskripsi:

No urut 1
Ketua : Nama Ketua (NIM)
 Prodi - Fakultas
Wakil : Nama Wakil (NIM)
 Prodi - Fakultas

Visi:

Misi:

Proker:

No urut 2
Ketua : Nama Ketua (NIM)
 Prodi - Fakultas
Wakil : Nama Wakil (NIM)
 Prodi - Fakultas

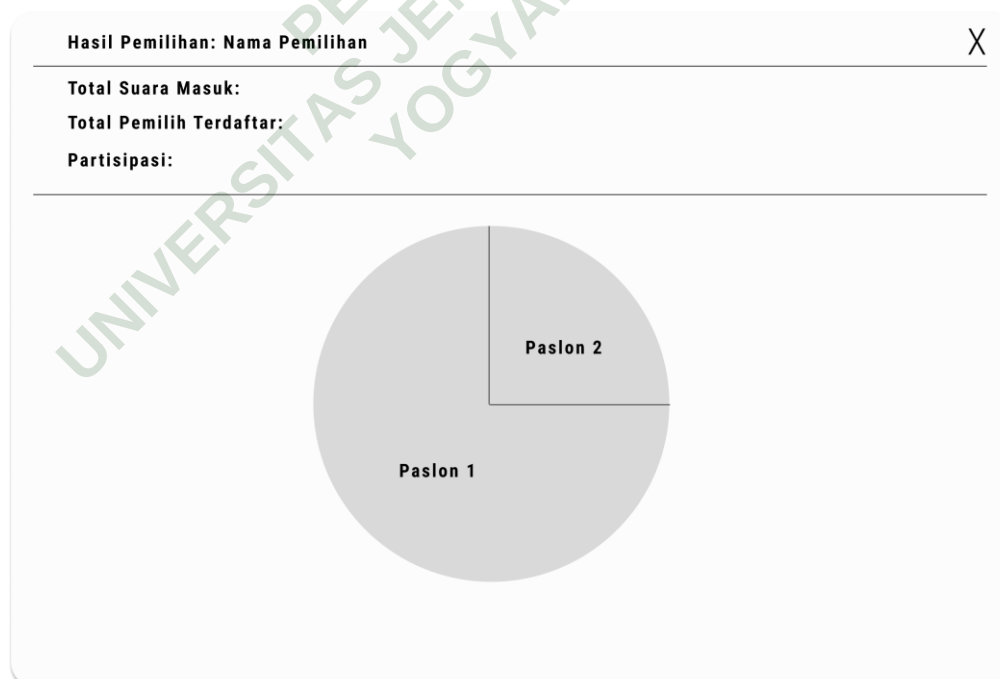
Visi:

Misi:

Proker:

Tutup Vote

Gambar 3.17 Desain Modal *Detail Voting*



Gambar 3.18 Desain Modal *Result*

3.5.9 Desain Halaman Pemilihan Kandidat

Pada Gambar 3.19 memperlihatkan desain antarmuka halaman pemilihan kandidat. Pada halaman ini, pengguna dapat melihat detail pemilihan, memilih pasangan calon dengan radio button, serta memasukkan token *voting* yang diterima via email. Tersedia juga fitur request token jika belum menerima token sebelumnya. Setelah memilih, pemilih dapat menekan tombol “Kirim Suara” untuk mengirimkan pilihannya secara aman.

E-Voting Pemira Unjaya

NAVIGATION

- Home
- Logout

User001
user001@gmail.com
Log Out

Pemilihan : Nama Pemilihan

Deskripsi:

Periode : waktu mulai - waktu selesai

☐ No urut 1

Ketua : Nama Ketua (NIM)
Prodi - Fakultas

Wakil : Nama Wakil (NIM)
Prodi - Fakultas

☒ No urut 2

Ketua : Nama Ketua (NIM)
Prodi - Fakultas

Wakil : Nama Wakil (NIM)
Prodi - Fakultas

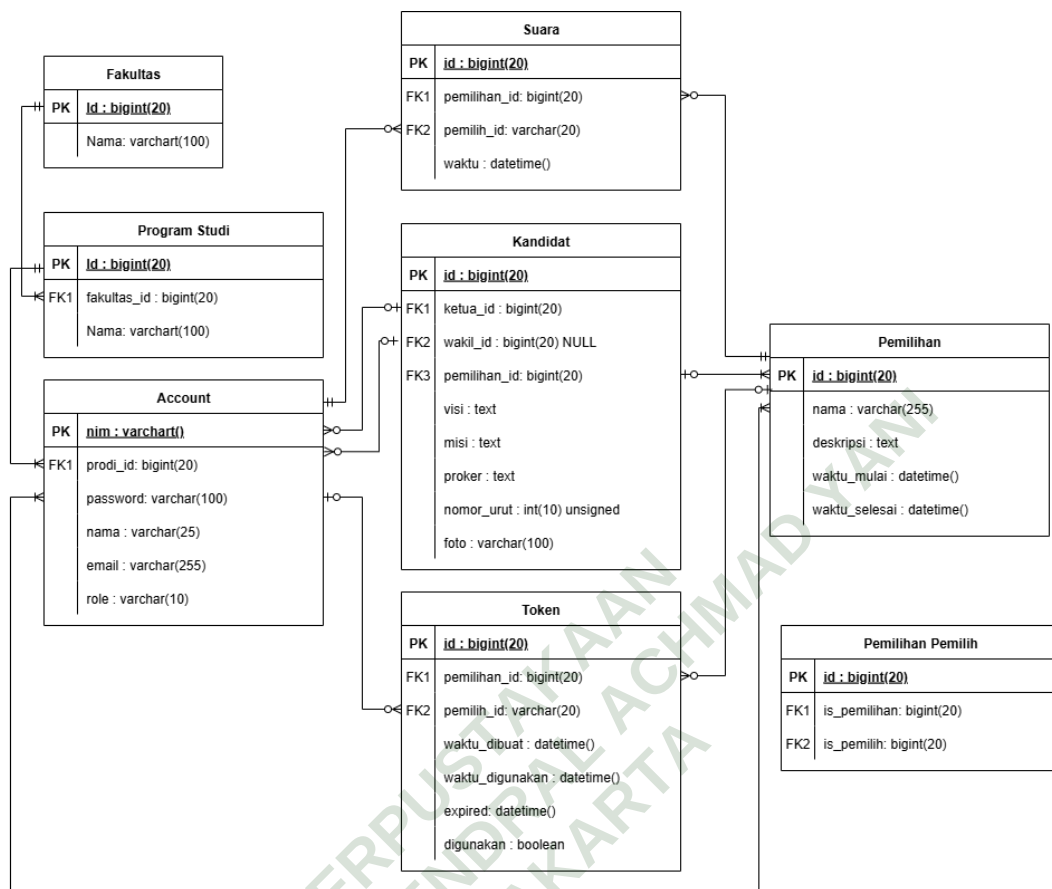
Masukan Toke Disini

Request token dan kirim ke email Kirim Suara

Gambar 3.19 Desain Halaman Pemilihan Kandidat

3.6 DESAIN STRUKTUR DATABASE

Desain struktur *database* merupakan bagian penting dalam pengembangan aplikasi *e-voting*, karena menjadi dasar dalam pengelolaan data pengguna, pemilih, kandidat, pemilihan, serta proses pencatatan suara. Pada sistem ini, rancangan *database* dibuat dengan memperhatikan keterkaitan antar-entitas agar proses bisnis berjalan efektif dan efisien. Setiap tabel seperti *Account*, *Fakultas*, *Prodi*, *Pemilihan*, *Kandidat*, *Suara*, dan *TokenVote* dihubungkan dengan relasi yang jelas untuk menjaga integritas dan keamanan data. Struktur ini juga memudahkan pengelolaan data, pencarian, dan pembuatan laporan hasil pemilihan. Ilustrasi hubungan antar tabel dalam sistem dapat dilihat pada Gambar 3.20.



Gambar 3.20 Entity Relationship Diagram (ERD)

Pada sistem *e-voting* ini, tabel *Account* berperan sebagai basis data utama untuk menyimpan seluruh pengguna sistem, baik itu admin maupun pemilih. Relasi pertama terjadi antara *Account* dan *Kandidat*, di mana setiap kandidat ketua dan wakil ketua pada tabel *Kandidat* diambil langsung dari data pengguna pada tabel *Account*. Artinya, hanya mahasiswa yang terdaftar di tabel *Account* yang bisa didaftarkan sebagai kandidat. Selanjutnya, *Account* juga berelasi dengan *Program Studi* dan *Fakultas*, di mana setiap pengguna atau mahasiswa memiliki atribut prodi dan fakultas untuk keperluan data administratif dan pelaporan. Tabel *Pemilihan* mendefinisikan setiap event pemilihan yang berjalan, dan memiliki relasi *One-to-Many* dengan tabel *Kandidat*, sehingga satu pemilihan dapat memiliki banyak kandidat yang mengikuti. Untuk proses pemilihan suara, *Account* juga berelasi dengan *Pemilihan* melalui entitas *Suara* dan *Token*. Tabel *Suara* merekam setiap pilihan yang dilakukan oleh pemilih (*Account*) pada suatu *Pemilihan* untuk satu

Kandidat tertentu, sehingga setiap suara yang masuk dapat dilacak berdasarkan pemilih, kandidat, dan waktu *voting*. Selain itu, Token diberikan ke masing-masing pemilih (*Account*) untuk satu pemilihan sebagai mekanisme verifikasi agar setiap pemilih hanya dapat memberikan suara satu kali. Relasi antara Token dengan *Account* dan Pemilihan memastikan bahwa token hanya berlaku untuk satu pemilih dan satu event pemilihan saja. Terakhir, Kandidat juga menyimpan referensi ke Pemilihan, sehingga kandidat yang didaftarkan selalu terkait dengan pemilihan tertentu, dan referensi ke *Account* sebagai ketua dan wakil. Secara keseluruhan, setiap relasi antar tabel di desain untuk menjaga keamanan data, keaslian suara, serta memudahkan integrasi proses administrasi dan pelaporan dalam sistem *e-voting* Pemira Unjaya. Dan struktur masing masing akan dijelaskan di poin-poin berikutnya.

3.6.1 Tabel Fakultas, Prodi, dan *Account*

Tabel 3.5 Struktur Tabel Fakultas

<i>Field</i>	Type	Extra	Action
id	bigint(20)	<i>Auto increment</i>	PK
nama	varchar(100)		

Tabel 3.6 Struktur Tabel Prodi

<i>Field</i>	Type	Extra	Action
id	bigint(20)	<i>Auto increment</i>	PK
nama	varchar(100)		
fakultas_id	bigint(20)		FK → main_fakultas(id)

Tabel 3.7 Struktur Tabel *Account*

<i>Field</i>	Type	Extra	Action
<i>password</i>	varchar(128)		
last_login	datetime(6)	NULL	

is_superuser	tinyint(1)		
first_name	varchar(150)		
last_name	varchar(150)		
is_staff	tinyint(1)		
is_active	tinyint(1)		
date_joined	datetime(6)		
nim	varchar(20)		PK
email	varchar(254)		UNIQUE
photo_profile	varchar(100)	NULL	
role	varchar(10)		
prodi_id	bigint(20)	NULL	FK main_prodi(id) →

Tabel Fakultas (Tabel 3.5) berfungsi untuk menyimpan data seluruh fakultas yang ada di institusi. Setiap fakultas memiliki *id* sebagai primary key.

Tabel Prodi (Tabel 3.6) berfungsi untuk menyimpan data program studi yang berada di bawah fakultas tertentu. Tabel ini memiliki *fakultas_id* (bigint) yang menjadi *foreign key* (FK) dan merujuk ke *id* pada tabel Fakultas. Dengan demikian, setiap prodi terhubung dengan satu fakultas.

Tabel Account (Tabel 3.7) digunakan untuk menyimpan data pengguna, baik admin maupun pemilih. *Field* utama pada tabel ini antara lain NIM sebagai primary key. Relasi *foreign key* pada *prodi_id* memastikan setiap pengguna terhubung ke salah satu prodi yang ada.

Struktur dan relasi antar ketiga tabel ini bertujuan untuk mendukung validasi data pengguna berdasarkan struktur organisasi kampus, serta memudahkan proses administrasi dan integrasi data dalam sistem *e-voting*.

3.6.2 Tabel Pemilihan

Tabel Pemilihan digunakan untuk menyimpan seluruh data pemilihan yang berlangsung di sistem *e-voting*. Setiap pemilihan memiliki *id* bertipe bigint (*auto increment*) sebagai *primary key* (PK), sehingga setiap pemilihan dapat diidentifikasi secara unik. Tabel ini menjadi pusat acuan untuk seluruh proses

pemilihan, pengaturan jadwal, serta pengelolaan kandidat dan suara dalam aplikasi *e-voting*. Tabel 3.8 menunjukkan struktur tabel dari pemilihan.

Tabel 3.8 Struktur Tabel Pemilihan

<i>Field</i>	<i>Type</i>	<i>Extra</i>	<i>Action</i>
id	bigint(20)	<i>Auto increment</i>	PK
nama	varchar(255)		
deskripsi	longtext	NULL	
waktu_mulai	datetime(6)		
waktu_selesai	datetime(6)		

3.6.3 Tabel Pemilihan Pemilih

Tabel Pemilihan Pemilih berfungsi sebagai penghubung (*junction table*) antara pemilihan dengan para pemilih yang berhak berpartisipasi dalam suatu pemilihan. Setiap entri pada tabel ini memiliki *id* bertipe *bigint* (*auto increment*) sebagai *primary key* (PK), serta *field pemilihan_id* bertipe *bigint* yang merupakan *foreign key* (FK) dan mengacu pada *field id* di tabel *main_pemilihan*. Dengan demikian, tabel ini memungkinkan sistem untuk mencatat daftar pemilih pada setiap pemilihan yang berbeda, sehingga hanya mahasiswa yang terdaftar di tabel ini yang dapat mengikuti proses *voting* pada pemilihan tertentu. Tabel ini terbentuk karena adanya relasi *many to many* antara tabel pemilih dan pemilihan. Tabel 3.9 merupakan struktur tabel pemilihan pemilih.

Tabel 3.9 Struktur Tabel Pemilihan Pemilih

<i>Field</i>	<i>Type</i>	<i>Extra</i>	<i>Action</i>
id	bigint(20)	<i>Auto increment</i>	PK
pemilihan_id	bigint(20)		FK → main_pemilihan(id)
account_id	varchar(20)		FK → main_account(nim)

3.6.4 Tabel Kandidat

Tabel Kandidat menyimpan data pasangan calon yang mengikuti setiap pemilihan. Setiap kandidat memiliki *id* sebagai *primary key*.

Field ketua_id adalah *foreign key* yang mengacu ke *field* nim pada tabel *account* untuk menentukan mahasiswa sebagai ketua kandidat. *Field* wakil_id juga merupakan *foreign key* ke *account* (opsional/NULL), digunakan bila ada wakil dalam pasangan calon. Dengan struktur ini, tabel Kandidat memungkinkan sistem mengelola semua data pasangan calon secara lengkap dan terhubung ke pemilihan serta data mahasiswa secara relasional. Tabel 3.10 merupakan struktur dari tabel pengguna.

Tabel 3.10 Struktur Tabel Kandidat

Field	Type	Extra	Action
id	bigint(20)	<i>Auto increment</i>	PK
visi	longtext	NULL	
misi	longtext	NULL	
proker	longtext	NULL	
foto	varchar(100)	NULL	
ketua_id	varchar(20)		FK → main_account(nim)
wakil_id	varchar(20)	NULL	FK → main_account(nim)
pemilihan_id	bigint(20)		FK → main_pemilihan(id)
nomor_urut	int(10) unsigned		

3.6.5 Tabel Token

Tabel Token digunakan untuk mengelola token unik yang diperlukan oleh setiap pemilih agar dapat memberikan suara dalam sebuah pemilihan. *Field* id sebagai primary key.

Dengan struktur ini, setiap token hanya dapat digunakan satu kali oleh pemilih tertentu dalam satu pemilihan, serta dapat dipantau status dan riwayat penggunaannya secara jelas. Tabel 3.11 merupakan struktur dari tabel token.

Tabel 3.11 Struktur Tabel Token

<i>Field</i>	Type	Extra	Action
id	bigint(20)	<i>Auto increment</i>	PK
token	varchar(12)		UNIQUE
digunakan	tinyint(1)		
waktu_dibuat	datetime(6)		
waktu_digunakan	datetime(6)	NULL	
expired	datetime(6)	NULL	
pemilihan_id	bigint(20)		FK → main_pemilihan(id)
pemilih_id	varchar(20)		FK → main_account(nim)

3.6.6 Tabel Suara

Tabel Suara menyimpan setiap suara yang diberikan oleh pemilih pada saat proses pemilihan berlangsung. Struktur ini memastikan setiap suara dapat dilacak ke waktu, pemilih, kandidat, dan pemilihan terkait, serta mencegah terjadinya duplikasi suara untuk satu pemilih dalam satu pemilihan. Tabel 3.12 merupakan struktur dari tabel suara.

Tabel 3.12 Struktur Tabel Suara

<i>Field</i>	Type	Extra	Action
id	bigint(20)	<i>Auto increment</i>	PK
waktu_vote	datetime(6)		
kandidat_id	bigint(20)		FK → main_kandidat(id)
pemilihan_id	bigint(20)		FK → main_pemilihan(id)
pemilih_id	varchar(20)		FK → main_account(nim)

3.7 PENGUJIAN SISTEM

Pada tahap pengujian sistem, dilakukan serangkaian tes untuk memastikan seluruh fitur aplikasi *e-voting* berjalan sesuai dengan kebutuhan dan bebas dari error yang dapat mengganggu proses pemilihan. Pengujian meliputi *unit testing* pada setiap modul utama seperti login, manajemen data pemilih, pengelolaan kandidat, pemilihan, dan proses *voting*, termasuk juga pengujian pada fitur keamanan seperti token dan deteksi kecurangan. Selain itu, dilakukan *integration testing* untuk memastikan modul-modul dapat berfungsi dengan baik secara terintegrasi. User Acceptance Testing (UAT) juga dilakukan dengan melibatkan beberapa pengguna (admin dan pemilih) guna mengevaluasi kemudahan penggunaan, keamanan, dan kepuasan pengguna terhadap sistem.

3.7.1 *Unit Testing*

Unit testing adalah metode pengujian dalam pengembangan perangkat lunak yang bertujuan memastikan bahwa setiap unit kode bekerja sesuai harapan dan memberikan hasil konsisten setiap kali dijalankan. Unit adalah komponen terkecil dalam perangkat lunak yang dapat diuji secara terpisah, biasanya menerima satu atau beberapa *input* serta menghasilkan satu *output*. Praktik *unit testing* menawarkan berbagai manfaat seperti mendeteksi bug sejak dini, mempercepat proses pengembangan, mencegah munculnya regresi bug, serta membantu developer memahami struktur dan perilaku kode program (Valerian et al., 2025).

3.7.2 *Integration Testing*

Integration testing merupakan tahap pengujian lanjutan setelah *unit testing* selesai dilakukan. Jika unit-unit yang diuji secara terpisah telah berjalan dengan baik, maka tahap ini menguji kombinasi dua atau lebih unit yang digabungkan menjadi satu kesatuan. Sering kali, pengujian integrasi menemukan permasalahan baru karena adanya perbedaan jenis *error* pada tiap level pengujian. Tujuan utama *integration testing* adalah memastikan bahwa unit-unit yang telah dibuat dapat berinteraksi dengan benar dan bekerja sesuai harapan dalam lingkungan sesungguhnya (Valerian et al., 2025).

3.7.3 User Acceptance Testing (UAT)

Menurut Anam et al., (2023) Untuk memastikan kelayakan program, dilakukan pengujian lebih lanjut menggunakan metode *user acceptance testing* (UAT). *User acceptance testing* merupakan tahapan di mana pengguna secara langsung mencoba sistem guna mendapatkan bukti hasil pengujian dan memastikan bahwa sistem telah berjalan sesuai dengan kebutuhan pengguna. Pengujian ini termasuk dalam kategori beta testing, karena sistem diuji langsung oleh pengguna sebagai bagian dari proses pengembangan perangkat lunak. Dalam proses UAT, pengguna diberikan kesempatan untuk mencoba langsung Sistem *E-voting* dan selanjutnya diminta mengisi kuesioner berdasarkan bobot penilaian yang tertera pada tabel 3.13.

Tabel 3.13 Keterangan Bobot

Bobot	Keterangan	Persentase
1	Sangat Tidak Sesuai	0% - 20%
2	Tidak Sesuai	21% - 40%
3	Kurang Sesuai	41% - 60%
4	Sesuai	61% - 80%
5	Sangat Sesuai	81% - 100%

Data yang terkumpul nantinya akan dihitung dengan rumus :

$$\text{Mean/Rata - Rata} = \frac{\text{Jumlah Bobot Penilaian Responden}}{\text{Total Responden}}$$

$$\text{Persentase} = \frac{\text{Nilai Mean}}{\text{Bobot Maksimum}} \times 100\%$$

Pengujian UAT untuk pemilih(*voters*) dengan daftar pertanyaan pada tabel 3.14:

Tabel 3.14 Pernyataan Kuesoner Untuk Pemilih

Kategori	No	Pernyataan
<i>Usability</i>	1	Saya mudah memahami menu dan tampilan aplikasi sebagai pemilih.

	2	Saya dapat dengan mudah melihat informasi setiap kandidat sebelum melakukan <i>voting</i> .
Kemudahan	3	Saya dapat meminta token <i>voting</i> dan menerima token tersebut di email saya.
	4	Saya mudah memilih kandidat yang diinginkan.
Fungsionalitas	5	Saya dapat login ke aplikasi menggunakan NIM dan <i>password</i> .
	6	Sistem memastikan saya hanya dapat <i>voting</i> satu kali untuk setiap pemilihan yang saya ikuti.
	7	Saya dapat melihat hasil <i>voting</i> setelah selesai memilih.
Keamanan	8	Sistem menolak token yang salah, sudah digunakan, atau sudah kedaluwarsa.
	9	Data pribadi dan hasil <i>voting</i> saya terjaga kerahasiaannya.
Reliabilitas	10	Proses <i>voting</i> berjalan lancar tanpa error meskipun banyak pemilih yang aktif secara bersamaan.
	11	Saya dapat logout dan sistem mengakhiri sesi saya dengan benar.

Pengujian UAT untuk Admin dengan daftar pertanyaan pada tabel 3.15:

Tabel 3.15 Pernyataan Kuesoner Untuk Admin

Kategori	No	Pernyataan
Usability	1	Saya mudah memahami tampilan dan menu pada dashboard admin.
	2	Saya mudah menambah, mengedit, dan menghapus data pengguna.
Kemudahan	3	Saya dapat dengan mudah mengimpor data pemilih dari file Excel atau CSV.
	4	Saya dapat menggunakan fitur pencarian dan filter data pada daftar pemilih dan kandidat.
Fungsionalitas	5	Saya dapat membuat pemilihan baru dan mengatur jadwal periode pemilihan dengan benar.
	6	Saya dapat menambah dan mengelola kandidat untuk setiap pemilihan.
	7	Saya dapat mengatur daftar pemilih pada setiap pemilihan.

	8	Saya dapat memantau jumlah suara yang masuk secara <i>real-time</i> .
Keamanan	9	Sistem mengingatkan saya untuk mengganti <i>password</i> jika masih menggunakan <i>password</i> default.
	10	Saya dapat melihat log aktivitas kecurangan pada halaman monitoring.
Reliabilitas	11	Sistem tetap berjalan stabil ketika saya mengelola banyak data atau pengguna.
	12	Saya dapat logout dan sesi admin saya langsung berakhir dengan baik.

PERPUSTAKAAN
JENDRAL ACHMAD YAN
YOGYAKARTA