

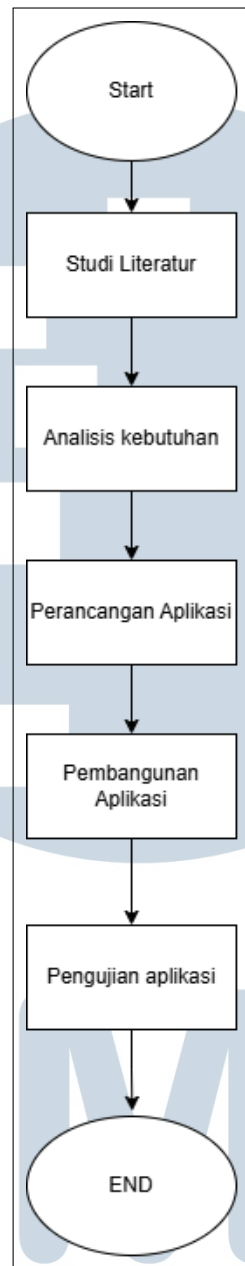
BAB 3

METODE PENELITIAN

3.1 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah metode *Simple Additive Weighting* (SAW). Metode ini dipilih untuk digunakan pada sistem rekomendasi dalam pemilihan karyawan Sales. Penggunaan metode SAW bertujuan untuk memberikan perhitungan yang sistematis dan objektif dalam menentukan karyawan yang memiliki kinerja unggul berdasarkan beberapa kriteria yang telah ditentukan. Gambar flowchart metode penelitian ini dapat dilihat pada Gambar 3.1





Gambar 3.1. Flowchart Metode Penelitian

3.1.1 Studi Literatur

Studi literatur merupakan langkah awal dalam penulisan laporan penelitian. Pada tahap ini, peneliti mengumpulkan informasi dan referensi yang relevan dengan topik penelitian yang dikembangkan. Sumber informasi yang digunakan meliputi karya ilmiah, jurnal, dan artikel yang digunakan mendukung pemahaman teori.

3.1.2 Analisis kebutuhan

Di tahap ini akan dilakukan analisis kebutuhan, peneliti melakukan pengumpulan data yang dibutuhkan sebagai dasar pengembangan sistem. Data yang dikumpulkan meliputi data karyawan, data pribadi, dan nilai kinerja karyawan tersebut. Analisis ini bertujuan untuk memahami kebutuhan dan spesifikasi sistem yang akan dibangun sehingga dapat berjalan dengan optimal sesuai tujuan penelitian.

3.1.3 Perancangan Aplikasi

Di tahap perancangan, sistem yang mulai dirancang dengan memperhatikan kebutuhan yang sudah dianalisis sebelumnya. Perancangan ini mencakup desain tampilan antarmuka serta fungsi-fungsi utama yang akan diimplementasikan.

3.1.4 Pembangunan Aplikasi

Setelah tahap perancangan selesai, dilakukan pembangunan aplikasi berbasis Web. Di tahap ini sistem berbasis Web yang sebelumnya sudah dirancang akan dibangun. Sistem yang dibangun akan menerapkan metode *Simple Additive Weighting* untuk menentukan karyawan terbaik secara otomatis.

3.1.5 Pengujian aplikasi

Pada tahap akhir, dilakukan pengujian sistem berbasis Web yang telah dikembangkan. Pengujian ini bertujuan untuk memastikan bahwa sistem berjalan dengan baik, sesuai dengan spesifikasi yang telah ditentukan, serta dapat memberikan hasil rekomendasi yang akurat dan tepercaya. Evaluasi dan perbaikan juga dilakukan berdasarkan hasil pengujian.

3.2 Perancangan Sistem

Perancangan sistem merupakan tahap penting dalam proses pengembangan sebuah sistem. Pada tahap ini menjelaskan beberapa komponen utama untuk merancang pembuatan sistem yang dilakukan secara mendetail agar proses pembangunan sistem dapat berjalan dengan lancar. Komponen-komponen yang

akan dirancang meliputi *flowchart*, struktur *database*, dan rancangan antarmuka sebagai bagian dari keseluruhan sistem

3.2.1 Teknologi

A Next.js

Next.js adalah sebuah kerangka kerja atau *framework* berbasis React yang dikembangkan oleh Vercel, dirancang untuk membangun aplikasi *web full-stack* yang modern dan berperforma tinggi. Sebagai kerangka kerja, Next.js menyediakan struktur dan serangkaian fitur siap pakai di atas pustaka React, yang secara fundamental hanya berfokus pada pembangunan antarmuka pengguna (UI). Keunggulan utama Next.js terletak pada kemampuannya untuk melakukan *Server-Side Rendering (SSR)* dan *Static Site Generation (SSG)*, yang mengatasi beberapa keterbatasan dari aplikasi React yang murni *Client-Side Rendering (CSR)*, seperti waktu muat awal yang lebih lambat dan tantangan dalam optimisasi mesin pencari (SEO) [18].

Dalam konteks pengembangan sistem ini, penggunaan Next.js memberikan keuntungan yang signifikan. Kemampuan *Server-Side Rendering* memastikan bahwa halaman awal yang berisi papan peringkat (*leaderboard*) dapat diindeks dengan baik oleh mesin pencari dan memiliki waktu muat yang cepat, karena halaman HTML telah dirender di sisi server sebelum dikirim ke klien. Selain itu, fitur bawaan seperti *file-system based routing* menyederhanakan proses pembuatan rute navigasi, dan API Routes memungkinkan pembuatan titik akhir atau *endpoint API* langsung di dalam proyek Next.js tanpa memerlukan server backend terpisah. Fitur-fitur ini secara kolektif mempercepat siklus pengembangan dan menghasilkan aplikasi yang lebih kokoh, skalabel, dan optimal [19].

B MongoDB

MongoDB adalah sistem manajemen basis data NoSQL (*Not Only SQL*) yang populer dan berorientasi pada dokumen. Berbeda dengan basis data relasional tradisional yang menggunakan skema tabel dengan baris dan kolom yang kaku, MongoDB menyimpan data dalam format BSON (*Binary JSON*), yang merupakan representasi biner dari dokumen JSON. Pendekatan ini memungkinkan penyimpanan data yang bersifat semi-terstruktur atau tidak terstruktur dalam satu

koleksi, memberikan fleksibilitas yang tinggi karena setiap dokumen tidak harus memiliki struktur atau set bidang yang sama [20].

Fleksibilitas skema yang ditawarkan oleh MongoDB menjadi pertimbangan utama dalam pemilihannya untuk proyek ini. Seiring dengan kemungkinan berkembangnya kriteria penilaian atau penambahan atribut data karyawan di masa depan, MongoDB memungkinkan perubahan ini diimplementasikan tanpa memerlukan migrasi skema yang kompleks. Model data dokumen juga secara alami memetakan objek dalam kode aplikasi misalnya, objek *JavaScript* atau *TypeScript*, yang menyederhanakan logika akses data dan mengurangi kompleksitas dalam lapisan persistensi. Didukung oleh kemampuannya untuk melakukan *horizontal scaling* melalui mekanisme *sharding*, MongoDB menjadi pilihan yang solid untuk mendukung pertumbuhan data aplikasi di masa depan [21].

C NextAuth.js

NextAuth.js adalah sebuah *library autentikasi* sumber terbuka yang dirancang khusus untuk aplikasi Next.js. Pustaka ini menyediakan solusi lengkap untuk menangani berbagai aspek keamanan dan manajemen sesi pengguna, menyederhanakan implementasi alur *autentikasi* yang kompleks. NextAuth.js dirancang untuk dapat diintegrasikan dengan mudah, mendukung berbagai strategi *autentikasi*, termasuk *autentikasi* berbasis kredensial (email dan kata sandi) serta penyedia layanan OAuth pihak ketiga seperti Google atau GitHub [22].

Penggunaan NextAuth.js dalam sistem ini bertujuan untuk mengabstraksi kerumitan dalam membangun sistem *autentikasi* dari awal. Pustaka ini secara otomatis menangani praktik keamanan fundamental, seperti proteksi terhadap *Cross-Site Request Forgery* (CSRF), penggunaan *server-only cookies* yang ditandatangani secara *kriptografis*, dan manajemen sesi berbasis JWT (*JSON Web Tokens*). Dengan mengimplementasikan NextAuth.js untuk mengelola sesi *login* administrator, pengembang dapat lebih fokus pada logika bisnis inti aplikasi, sambil memastikan bahwa mekanisme *autentikasi* yang digunakan telah teruji, aman, dan sesuai dengan standar industri.

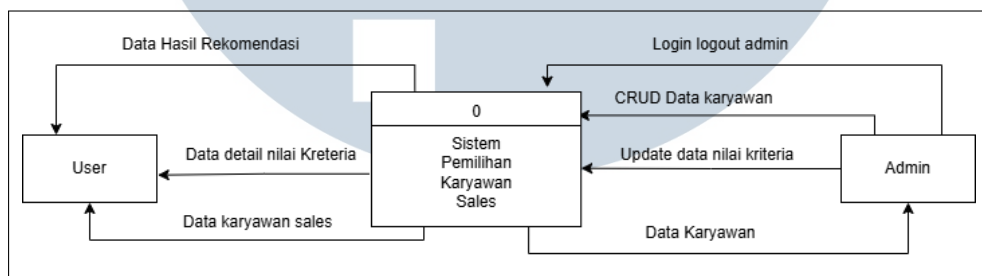
3.2.2 DFD

DFD atau *Data flow diagram* merupakan sebuah diagram yang menggambarkan alur informasi atau perpindahan data secara menyeluruh dalam

suatu sistem. Dengan menggunakan DFD, proses penyajian informasi menjadi lebih mudah dipahami karena disajikan dalam bentuk visualisasi yang sederhana namun informatif. Informasi yang diperoleh dari DFD dapat digunakan sebagai dasar dalam analisis sistem serta perancangan agar aliran data berjalan secara efisien dan sesuai dengan kebutuhan.

A DFD Level 0

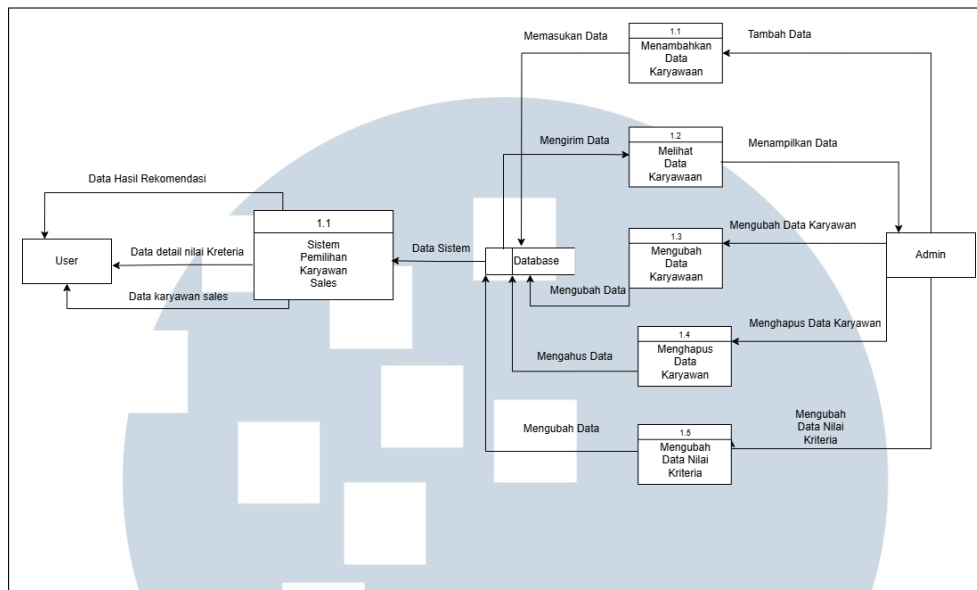
Pada DFD level 0, admin dan user menjadi 2 entitas pada sistem pemilihan yang dibuat. yang di mana dari sisi user, user mendapatkan data nilai hasil perhitungan serta mendapatkan data nilai karyawan lainnya. Admin juga mendapatkan data karyawan serta admin dapat mengirimkan data CRUD, data login, serta dapat melakukan *update* nilai karyawan yang dipilih. Hal ini dapat dilihat pada Gambar 3.2



Gambar 3.2. DFD Level 0

B DFD Level 1

Pada DFD level 1 ini, proses utama dilakukan oleh entitas admin dalam sistem, di sini admin dapat melakukan penambahan data karyawan, melihat data karyawan, mengubah data karyawan, menghapus data karyawan, serta mengubah data nilai karyawan. Setiap proses ini berfungsi untuk mengelola data karyawan yang tersimpan ke dalam *database*,



Gambar 3.3. DFD Level 1

Selanjutnya, data yang tersimpan dalam database digunakan oleh sistem pemilihan karyawan sales melalui proses yang dijalankan oleh user. Sistem ini mengolah data karyawan serta data nilai kriteria untuk menghasilkan rekomendasi karyawan terbaik berdasarkan metode *Simple Additive Weighting* (SAW). Hasil rekomendasi kemudian disajikan kembali kepada user sebagai output dari sistem. hal tersebut dapat dilihat pada Gambar 3.3

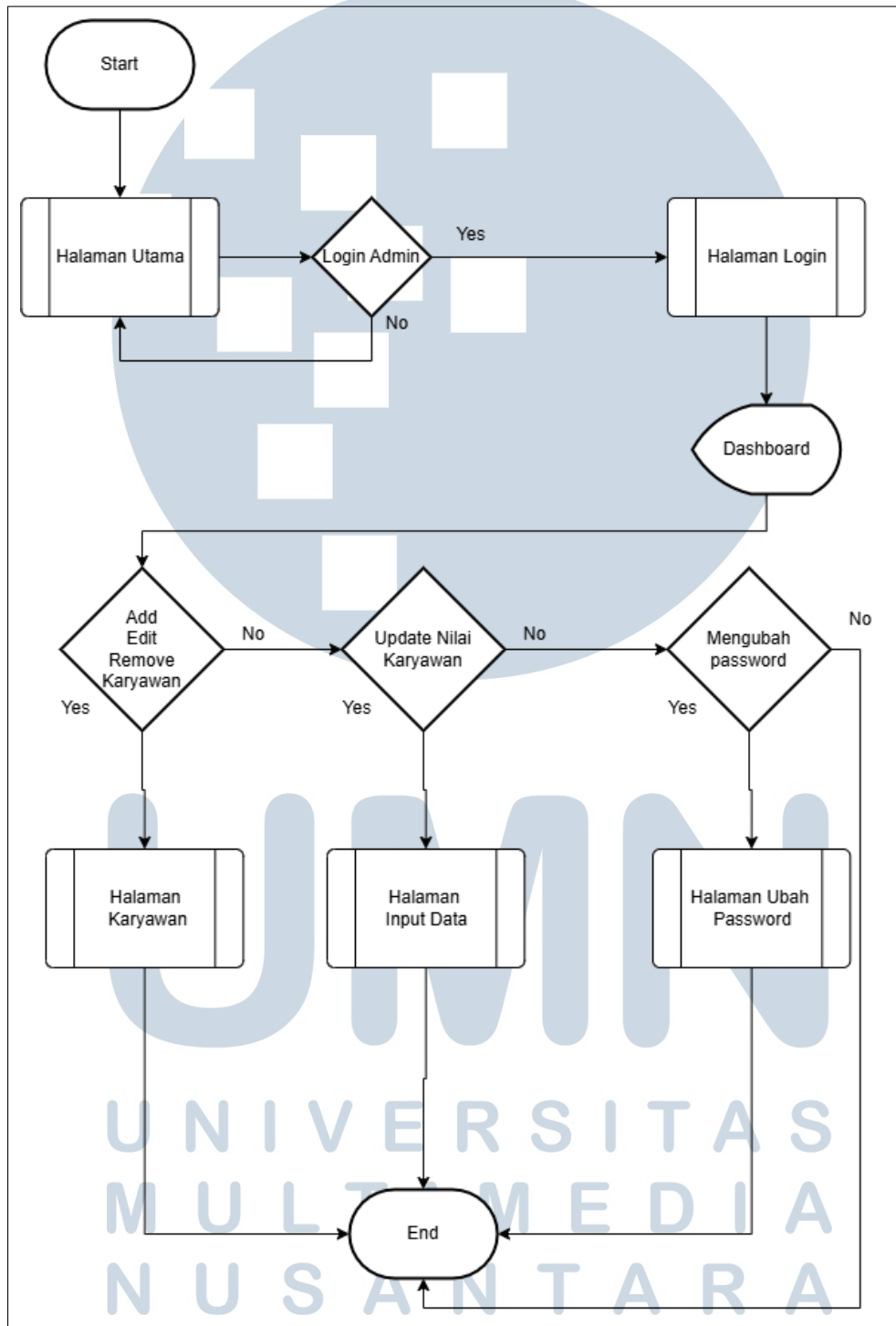
3.2.3 Flowchart

Flowchart adalah sebuah diagram yang menggambarkan urutan langkah-langkah atau proses dalam sebuah sistem secara sistematis dan visual. Dengan menggunakan flowchart dapat membantu dalam memahami alur kerja sistem secara menyeluruh Pada sistem ini,

A Flowchart Utama

Flowchart utama dari sistem pemilihan karyawan ini berfungsi untuk menggambarkan secara menyeluruh alur kerja dari program yang dirancang. Proses dimulai dengan tampilan halaman utama sebagai titik awal bagi user . Pada halaman utama ini, user bisa melihat secara langsung data hasil perhitungan rekomendasi karyawan terbaik yang telah diproses. Selain itu, di halaman ini juga disediakan fitur untuk melakukan proses *login*, user akan diarahkan ke halaman *login* untuk

melakukan verifikasi identitas. Jika proses *login* berhasil, user (admin) akan diteruskan ke halaman *dashboard*.



Gambar 3.4. Flowchart Utama

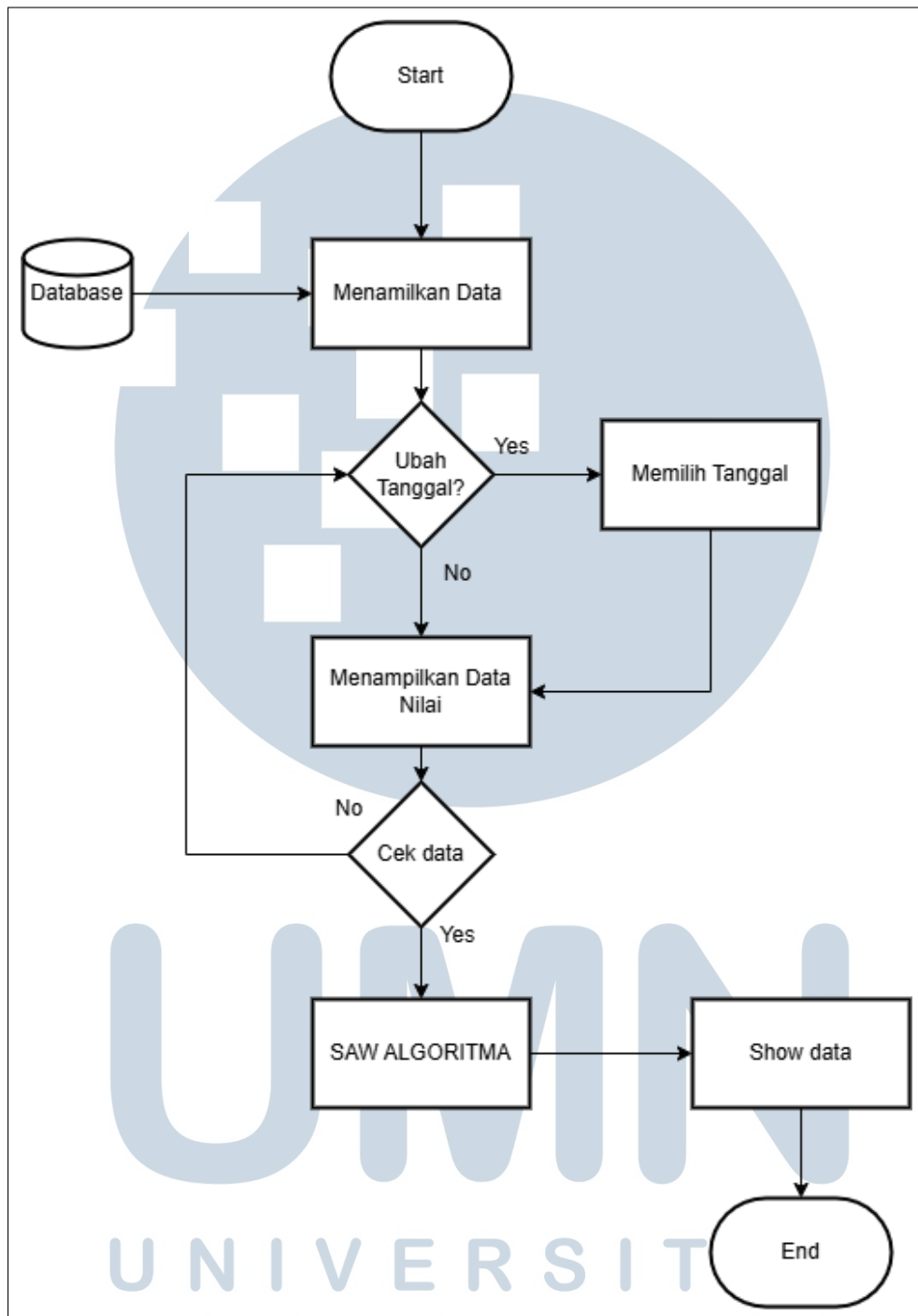
Di halaman ini menjadi pusat kendali dengan menyediakan tiga pilihan

menu utama, yaitu: halaman pengelolaan data karyawan, halaman untuk pengubahan nilai karyawan, serta halaman untuk mengubah *password* admin. Setelah seluruh aktivitas tersebut selesai, alur proses akan kembali berhenti atau keluar sebagai akhir dari flowchart. Gambar flowchart utama yang merepresentasikan alur ini dapat dilihat pada Gambar 3.4

B Flowchart Halaman Awal

Flowchart halaman awal ini menjelaskan secara detail proses yang terjadi ketika user pertama kali mengakses sistem pemilihan karyawan. Pada tahap awal, sistem akan secara otomatis menampilkan data hasil perhitungan yang tersimpan di dalam *database*. Data yang ditampilkan meliputi informasi tentang karyawan beserta nilai yang diperoleh oleh masing-masing karyawan.





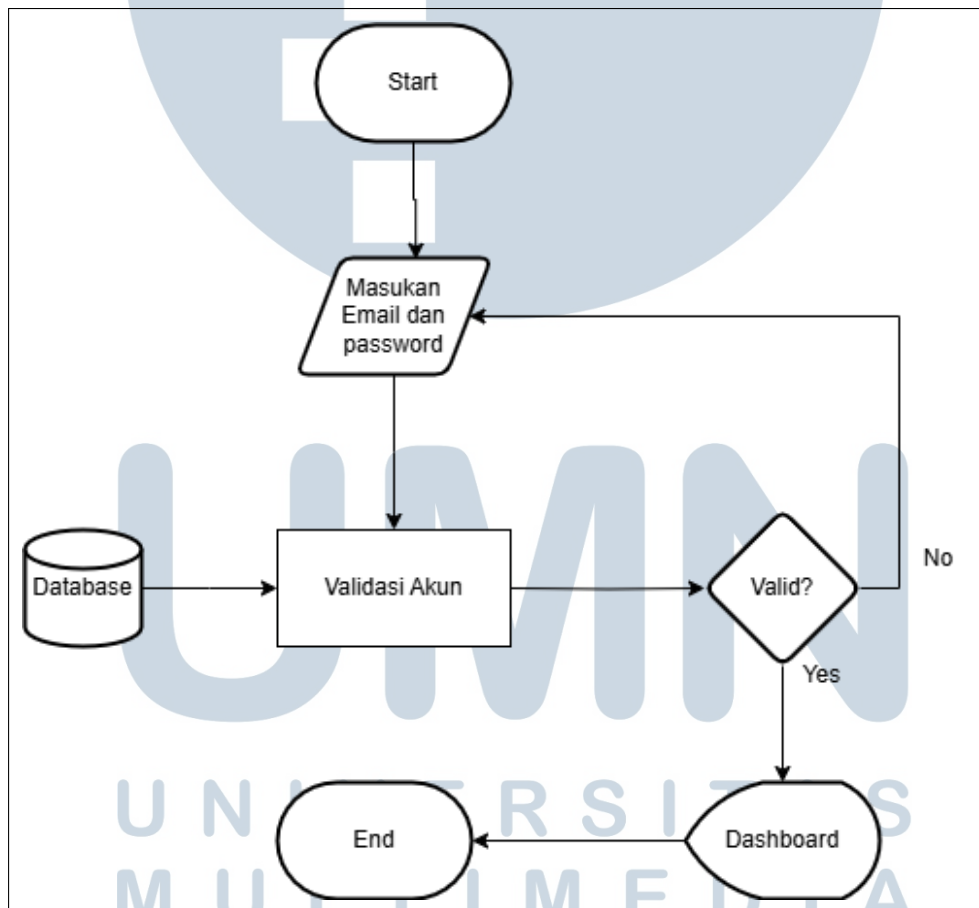
Gambar 3.5. Flowchart Halaman Awal

User diberikan opsi untuk mengubah tanggal tampilan data, seperti memilih bulan dan tahun tertentu, guna melihat hasil perhitungan pada periode waktu yang berbeda. Jika data pada bulan yang dipilih tidak tersedia, User dapat memilih tanggal lain karena mungkin pada bulan tersebut belum terdapat data yang di

input. Setelah tanggal dipilih dan data diperiksa, sistem akan menampilkan hasil skor yang telah dihitung menggunakan metode SAW. Gambar flowchart halaman awal ini dapat dilihat pada Gambar 3.5

3.2.4 Flowchart Login

Pada Flowchart *login* ini menggambarkan proses yang harus dilalui oleh user sebelum mengakses fitur di dalam sistem pemilihan karyawan. Pertama, user diminta untuk memasukkan informasi akun berupa *email* dan *password*. data yang di masukan ini kemudian dikirim ke sistem untuk dilakukan pemeriksaan melalui pengecekan pada database.



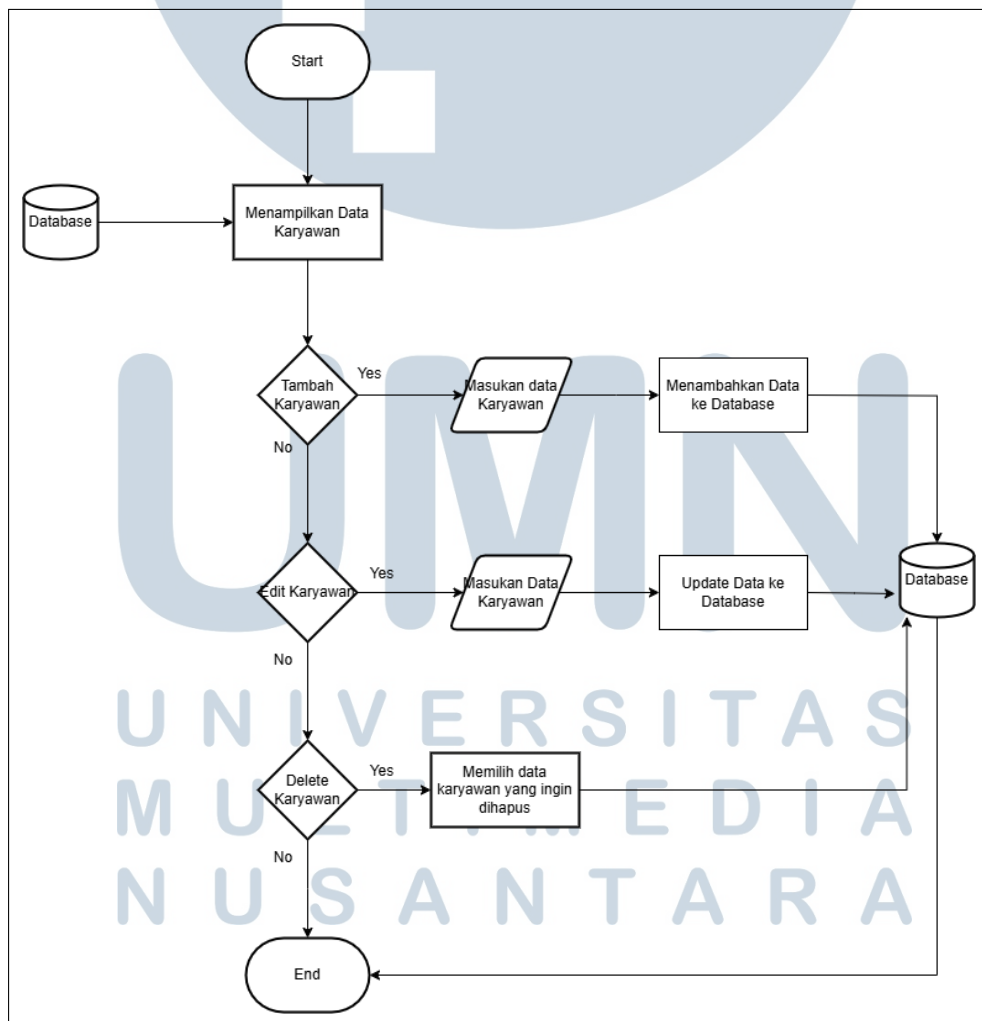
Gambar 3.6. Flowchart Login

Sistem akan melakukan validasi dengan memverifikasi kecocokan data akun yang dimasukkan dengan data yang tersimpan di dalam *database*. Jika data yang dimasukkan sesuai dengan yang ada di *database*, user akan diberikan akses

masuk ke halaman utama sistem yang disebut dashboard. Pada halaman dashboard tersebut, user dapat mengelola data karyawan serta melakukan proses lain yang diperbolehkan berdasarkan hak aksesnya. Bila data yang di masukan tidak sesuai, sistem tidak akan memberikan akses dan meminta user untuk mengulangi proses pengisian *email* dan *password*. Proses ini akan terus berulang hingga user berhasil memasukkan data yang benar atau keluar dari sistem. Gambar flowchart login ini dapat dilihat pada Gambar 3.6

A Flowchart Data Karyawan

Flowchart Halaman Data Karyawan menggambarkan proses pengelolaan data karyawan di dalam sistem. Pada awal proses, sistem akan mengambil dan menampilkan seluruh data karyawan yang tersimpan di dalam database.

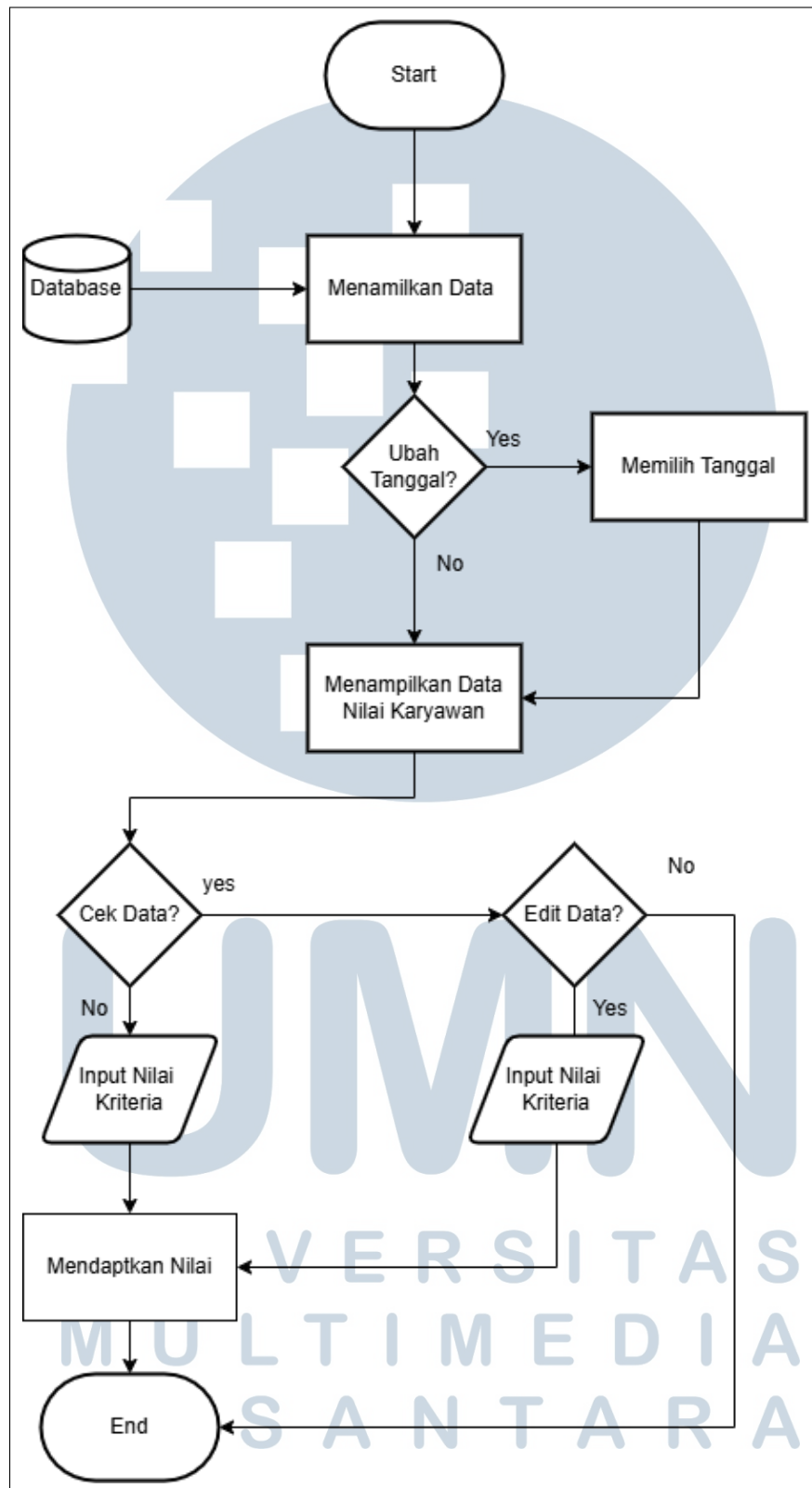


Gambar 3.7. Flowchart Data Karyawan

B Flowchart Nilai Karyawan

Pada Flowchart Nilai karyawan, pertama sistem akan mengambil data nilai karyawan yang ada dari *database*. user juga dapat mengubah bulan untuk memilih data dari bulan apa yang ingin di masukan, setelah itu user akan mendapatkan data karyawan, di sini sistem juga akan mengecek apakah data nilai dari karyawan tersebut sudah ada atau belum jika belum user dapat memasukkan nilai karyawan tersebut dan jika sistem sudah menampilkan nilai karyawan itu berarti, karyawan tersebut sudah dimasukkan nilainya sebelumnya. di sini user juga dapat mengubah nilai tersebut. Flowchart Nilai Karyawan ini dapat dilihat pada Gambar 3.8.





Gambar 3.8. Flowchart Nilai Karyawan

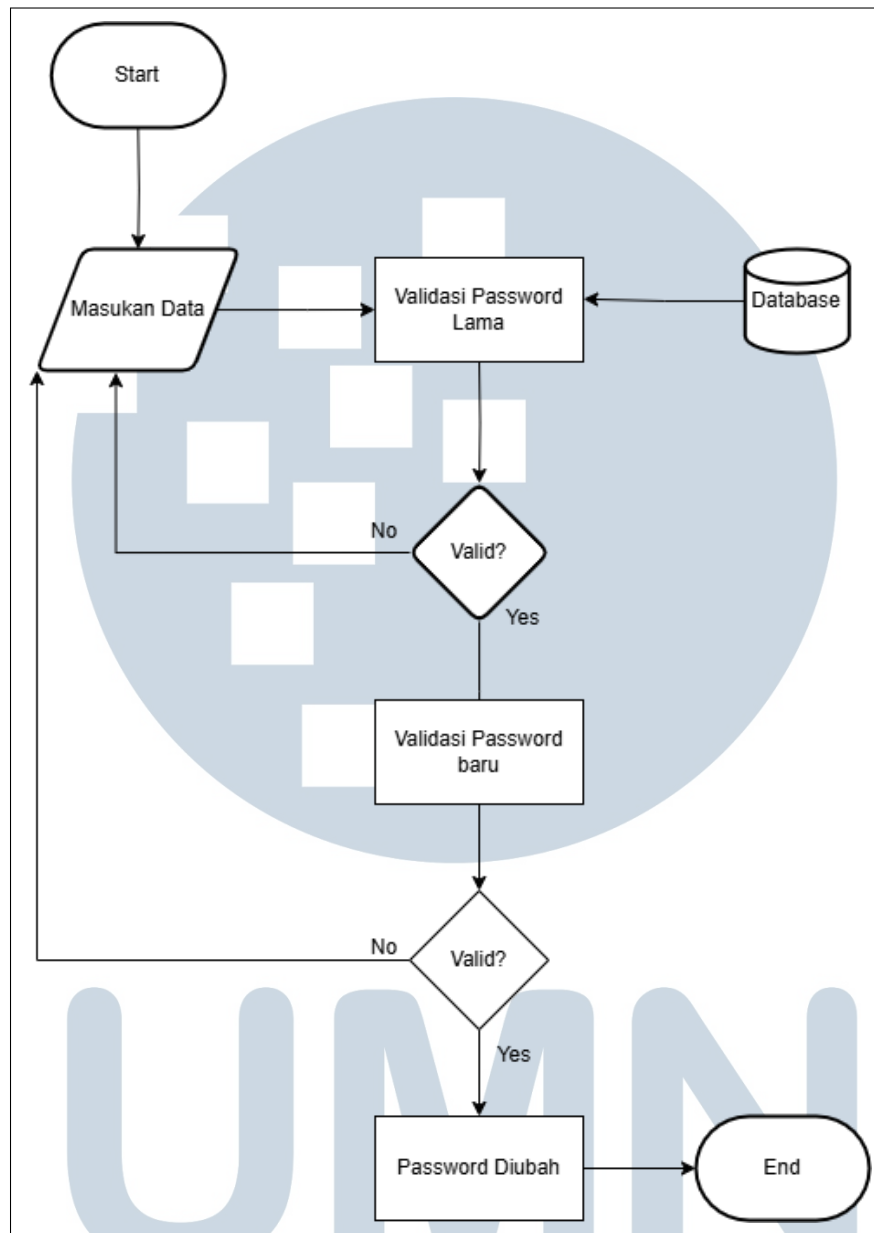
Flowchart pada halaman nilai karyawan menggambarkan proses pengelolaan data penilaian karyawan yang tersimpan didalam sistem. Pada awal proses, sistem akan mengambil data karyawan yang sudah tersimpan di *database* dan langsung menampilkannya kepada pengguna. Pengguna memiliki opsi untuk mengubah tanggal atau bulan untuk melihat data nilai yang ingin diakses, kemudian menampilkan data nilai karyawan sesuai tanggal tersebut.

Setelah data nilai karyawan ditampilkan, sistem melakukan pengecekan apakah nilai karyawan untuk periode tersebut sudah ada dalam database atau belum. Jika data nilai belum tersedia, maka pengguna dapat memasukkan nilai karyawan tersebut melalui input nilai kriteria yang disediakan sistem. tetapi, jika data nilai sudah ada dan ditampilkan oleh sistem, pengguna masih diberikan opsi untuk melakukan perubahan atau pembaruan nilai karyawan yang sudah ada. Apabila pengguna memilih untuk mengubah data, maka sistem akan menerima input nilai yang baru dan memperbarui data tersebut ke dalam database. Gambar flowchart nilai karyawan ini dapat dilihat pada Gambar 3.7.

C Flowchart Penggantian Password

Flowchart untuk halaman ganti *password* ini menggambarkan proses perubahan kata sandi di dalam sistem. pertama pengguna memasukkan data yang diperlukan, seperti *password* lama dan *password* baru, ke dalam sistem.





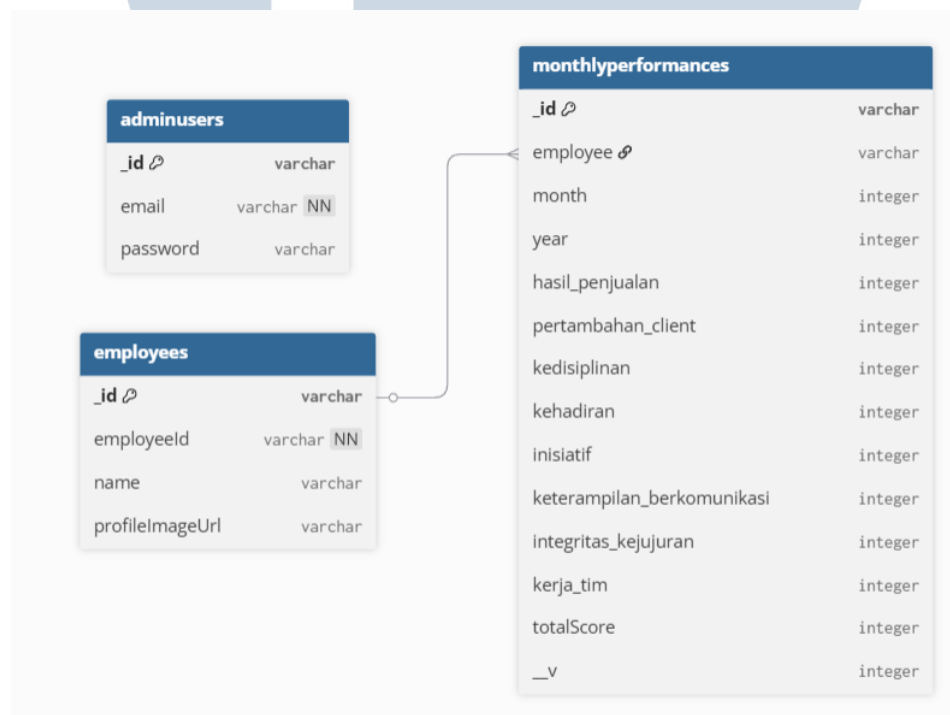
Gambar 3.9. Flowchart Ganti Password

Setelah data dimasukkan, sistem akan melakukan validasi terhadap *password* lama tersebut dengan membandingkannya dengan data yang tersimpan dalam *database*. Jika *password* lama yang dimasukkan tidak valid, maka proses akan kembali ke tahap input data. Jika *password* lama valid, maka sistem akan melanjutkan dengan validasi *password* baru yang dimasukkan oleh pengguna. user perlu memasukkan *password* yang baru untuk kedua kalinya dan sistem akan melakukan pengecekan untuk memastikan kedua *password* baru yang di masukan itu sama.

Apabila *password* baru tidak valid, sistem akan meminta pengguna untuk memasukkan kembali *password* baru yang memenuhi ketentuan. Namun, jika *password* baru valid, maka sistem akan melakukan perubahan *password* di *database* sesuai dengan input baru tersebut. Gambar flowchart ganti *password* ini dapat dilihat pada Gambar 3.9

3.2.5 Database Schema

Skema basis data merupakan fondasi dari struktur penyimpanan data pada sistem yang dibangun. Perancangan skema ini bertujuan untuk memastikan integritas dan efisiensi pengelolaan data. Sistem ini menggunakan tiga koleksi utama: *adminusers*, *employees*, dan *monthlyperformances*.



Gambar 3.10. Database Schema Sistem

Koleksi *adminusers* berfungsi untuk menyimpan data kredensial administrator, termasuk *email* dan *password* yang digunakan untuk proses otentikasi. Koleksi *employees* menyimpan data master karyawan, seperti ID, nama, dan URL foto profil.

Koleksi *monthlyperformances* menjadi pusat dari sistem penilaian, karena menyimpan seluruh data hasil evaluasi kinerja bulanan. Setiap dokumen dalam

koleksi ini terhubung ke satu karyawan melalui referensi *ObjectId* pada *field employee*. Selain itu, koleksi ini mencatat periode penilaian (bulan dan tahun) serta skor untuk setiap kriteria yang dievaluasi, seperti hasil penjualan, kedisiplinan, dan kerja tim, hingga menghasilkan skor total. Relasi antara koleksi *employees* dan *monthlyperformances* adalah satu-ke-banyak, di mana satu karyawan dapat memiliki banyak catatan kinerja bulanan. Struktur lengkap dari skema basis data ini dapat dilihat pada Gambar 3.10.

