

BAB 3

METODE PENELITIAN

Penelitian ini termasuk dalam kategori penelitian kuantitatif, karena berfokus pada analisis numerik dan pemrosesan data menggunakan algoritma KNN. Selain itu, penelitian ini juga mengukur akurasi sistem yang merupakan metode kuantitatif. Meskipun dominan kuantitatif, penelitian ini juga memiliki unsur pendekatan kualitatif, khususnya dalam penyusunan soal tes RIASEC dan proses validasi kuesioner melalui wawancara dengan dosen psikologi. Unsur kualitatif hanya digunakan sebagai pendukung dalam proses pengembangan instrumen dan validasi data, serta tidak menjadi bagian utama dalam analisis data. Inti analisis tetap berbasis data numerik, sehingga penelitian ini tetap dikategorikan sebagai penelitian kuantitatif, bukan penelitian campuran. Berikut ini adalah bahan, alat, tahapan penelitian, serta metode pengembangan perangkat lunak yang digunakan dalam pengembangan SPK pemilihan jurusan kuliah berbasis web.

3.1 BAHAN DAN ALAT PENELITIAN

Bahan penelitian berisi bahan-bahan yang digunakan untuk menciptakan solusi. Bahan penelitian yang digunakan dalam pengembangan sistem ini meliputi:

1. Literatur Ilmiah dan Penelitian Terdahulu

Penelitian ini mengacu pada literatur ilmiah dan penelitian terdahulu terkait SPK, algoritma KNN, dan teori kepribadian RIASEC, yang digunakan sebagai dasar dalam perancangan sistem.

2. Kuesioner RIASEC

Merupakan instrumen yang digunakan untuk mengumpulkan data kepribadian mahasiswa Unjaya berdasarkan enam tipe kepribadian RIASEC serta program studi yang mereka pilih.

3. Dataset RIASEC Mahasiswa Unjaya

Data ini dikumpulkan melalui kuesioner yang diberikan kepada mahasiswa Unjaya. Data ini mencakup jumlah skor untuk setiap tipe kepribadian RIASEC serta program studi yang dipilih oleh mahasiswa. Dataset ini akan

digunakan sebagai referensi dalam proses klasifikasi menggunakan algoritma KNN.

Alat yang digunakan dalam penelitian ini meliputi perangkat keras dan perangkat lunak yang mendukung proses pengembangan sistem. Perangkat yang digunakan harus memiliki spesifikasi yang cukup untuk menjalankan framework pengembangan, basis data, serta algoritma klasifikasi KNN. Perangkat keras yang digunakan dalam penelitian ini adalah laptop/PC dengan spesifikasi minimal:

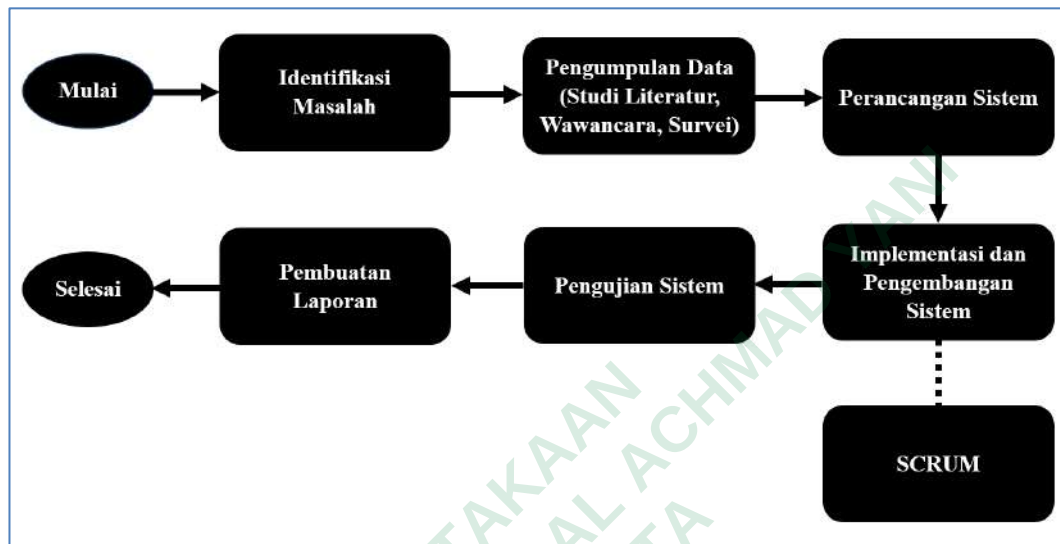
- a. Prosesor: Intel core i3 atau yang setara.
- b. RAM: Minimal 4 GB untuk mendukung eksekusi server lokal dan pengolahan data.
- c. Penyimpanan: SSD 256 GB untuk mempercepat akses data dan pengolahan sistem.

Adapun perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Google Formulir: Digunakan untuk mengumpulkan dataset RIASEC mahasiswa Unjaya.
2. Sistem Operasi: Windows 11.
3. Bahasa Pemrograman: Python.
4. *Framework* Pengembangan: Django, sebagai *framework backend* berbasis Python.
5. Basis Data: MySQL, sebagai sistem manajemen database untuk menyimpan data mahasiswa dan hasil klasifikasi.
6. Server Lokal: XAMPP, digunakan sebagai server lokal yang menyediakan Apache (web server), MySQL, dan PHPMyAdmin (antarmuka untuk mengelola database MySQL).
7. Editor Kode: Visual Studio Code.
8. Web *Browser*: Google Chrome, untuk menguji tampilan dan fungsionalitas sistem berbasis web.

3.2 JALAN PENELITIAN

Penelitian ini dilakukan melalui serangkaian tahapan yang sistematis untuk mengidentifikasi masalah, merancang, mengembangkan, serta menguji sistem yang dikembangkan. Tahapan-tahapan penelitian ditunjukkan pada Gambar 3.1.



Gambar 3.1 Tahapan Penelitian

1. Identifikasi Masalah

Tahap ini bertujuan untuk memahami permasalahan utama, yaitu banyaknya mahasiswa yang merasa salah jurusan akibat kesulitan dalam memilih jurusan yang sesuai dengan minat dan kepribadian mereka. Selain itu, belum adanya sistem berbasis web di Unjaya yang dapat membantu calon mahasiswa dalam menentukan jurusan sesuai dengan kepribadian dan minat mereka.

2. Pengumpulan Data

Setelah identifikasi masalah dilakukan, tahap berikutnya adalah mengumpulkan data yang diperlukan untuk pengembangan sistem. Proses pengumpulan data dilakukan melalui tiga metode utama, yaitu studi literatur, survei, dan wawancara.

a. Studi Literatur

Studi literatur dilakukan untuk mendapatkan dasar teori dan penelitian terdahulu yang relevan dengan penelitian ini.

b. Wawancara

Untuk mendapatkan data yang komprehensif mengenai pertanyaan-pertanyaan atau soal yang akan digunakan dalam tes RIASEC pada sistem dan kuesioner, dilakukan wawancara dengan dosen dari program studi psikologi Unjaya. Hal ini bertujuan untuk menggali pertanyaan-pertanyaan yang relevan dan efektif guna mengukur aspek kepribadian serta minat mahasiswa. Selain itu, wawancara juga berfungsi sebagai proses validasi data yang dikumpulkan dari hasil kuesioner. Hal ini bertujuan untuk mendapatkan pemahaman lebih dalam terkait korelasi antara hasil kuesioner RIASEC dengan pemilihan program studi. Data yang telah divalidasi kemudian digunakan sebagai dataset utama dalam sistem guna meningkatkan akurasi rekomendasi yang diberikan.

c. Wawancara

Survei dilakukan dengan menggunakan kuesioner untuk mengumpulkan data kepribadian mahasiswa Unjaya berdasarkan enam tipe kepribadian RIASEC serta program studi yang mereka pilih. Data ini digunakan sebagai dasar dalam penyusunan dataset yang akan diterapkan pada sistem untuk memberikan rekomendasi jurusan yang sesuai dengan minat dan bakat pengguna.

3. Perancangan Sistem

Tahap selanjutnya adalah merancang sistem berdasarkan hasil analisis kebutuhan dan data yang telah dikumpulkan. Perancangan sistem meliputi desain arsitektur sistem, model basis data, alur kerja sistem, serta model algoritma KNN. Pada tahap ini, sistem dirancang untuk memungkinkan pengguna melakukan tes kepribadian RIASEC, memproses data dengan algoritma KNN, dan menampilkan rekomendasi jurusan yang sesuai. Selain itu, dibuat pula rancangan tampilan antarmuka (UI) yang *user-friendly* agar pengguna dapat dengan mudah mengakses dan memahami hasil rekomendasi yang diberikan oleh sistem.

4. Implementasi dan Pengembangan Sistem

Setelah perancangan sistem selesai, tahap selanjutnya adalah implementasi dan pengembangan sistem menggunakan metode SCRUM. Meskipun SCRUM umumnya diterapkan dalam tim, tetapi tetap dapat disesuaikan untuk pengembangan individu dengan menyesuaikan beberapa tahapannya. Berikut adalah penerapan tahapan-tahapan SCRUM dalam pengembangan SPK Pemilihan Jurusan Kuliah ini:

a. *Product Backlog*

Proses dimulai dengan menyusun *product backlog*, di mana daftar fitur utama sistem disusun, seperti tes kepribadian RIASEC, informasi jurusan, serta dashboard admin untuk mengelola data kuesioner dan daftar jurusan.

b. *Sprint Planning*

Setelah *product backlog* tersusun, dilakukan *sprint planning*, yang membagi pengembangan ke dalam beberapa *sprint* dengan target penyelesaian fitur dalam kurun waktu tertentu.

c. *Daily Scrum (Self Check)*

Selama proses pengembangan, dilakukan *daily scrum*. Karena dikerjakan secara individu, *daily scrum* diadaptasi menjadi *self check* harian di mana setiap hari dilakukan evaluasi mandiri terhadap progres pengembangan, penentuan pekerjaan untuk hari berikutnya, serta pencatatan kendala yang dihadapi dan cara mengatasinya.

d. *Sprint Review*

Setelah setiap *sprint* selesai, dilakukan *sprint review* untuk menguji fitur yang telah dikembangkan dan memastikan bahwa fitur tersebut sesuai dengan yang direncanakan dalam *product backlog*. Jika terdapat kekurangan, perbaikan dilakukan sebelum fitur tersebut dianggap selesai.

e. *Sprint Retrospective*

Selanjutnya, dilakukan *sprint retrospective*, di mana setiap akhir sprint dilakukan evaluasi terhadap proses kerja untuk mengidentifikasi kendala yang dihadapi dan menemukan cara untuk meningkatkan efisiensi pengembangan pada *sprint* berikutnya.

5. Pengujian Sistem

Setelah implementasi selesai, tahap berikutnya adalah pengujian sistem untuk memastikan bahwa seluruh fitur berfungsi dengan baik. Pengujian dilakukan menggunakan metode *black box testing*, di mana setiap fitur diuji berdasarkan skenario penggunaan untuk memastikan bahwa sistem dapat berjalan sebagaimana mestinya. Selain itu, dilakukan pengujian akurasi dengan membandingkan hasil rekomendasi sistem dengan data yang telah terlabel dalam dataset untuk menilai tingkat akurasi algoritma yang diterapkan. Jika ditemukan kesalahan atau ketidaksesuaian, maka perbaikan dilakukan sebelum sistem resmi digunakan.

6. Pembuatan Laporan

Tahap terakhir dalam penelitian ini adalah penyusunan laporan yang mendokumentasikan seluruh tahapan penelitian dan pengembangan sistem. Laporan mencakup latar belakang penelitian, metode yang digunakan, hasil pengujian, serta kesimpulan yang diperoleh dari penelitian ini. Dokumentasi yang baik diharapkan dapat menjadi referensi bagi pengembangan lebih lanjut serta memberikan kontribusi bagi penelitian di bidang sistem pendukung keputusan.

3.3 PERANCANGAN SISTEM

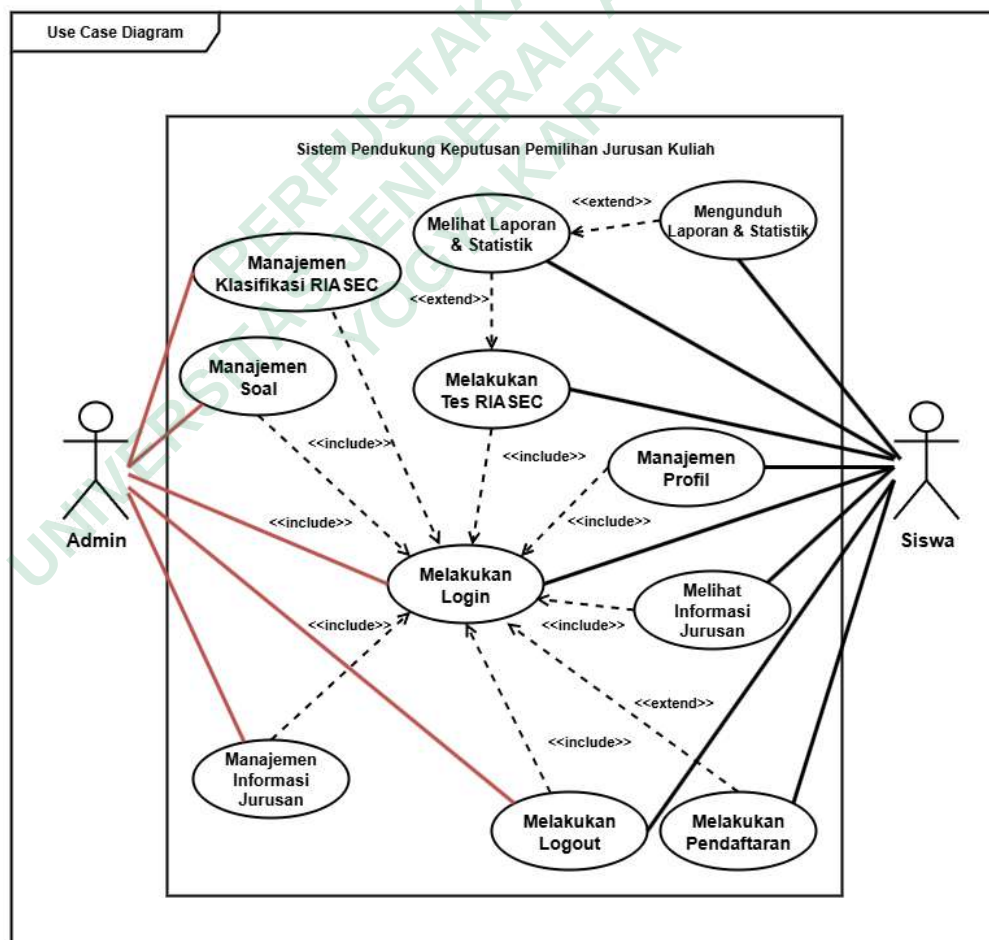
Perancangan sistem merupakan tahapan penting dalam pengembangan yang bertujuan untuk memodelkan dan menggambarkan bagaimana sistem akan bekerja sebelum diimplementasikan. Pada penelitian ini, perancangan sistem dilakukan melalui pendekatan Unified Modeling Language (UML), perancangan basis data, dan perancangan antarmuka pengguna (*user interface*).

3.3.1 Perancangan Sistem Menggunakan UML

Unified Modeling Language (UML) merupakan suatu metode dalam pengembangan sistem yang memanfaatkan bahasa grafis sebagai sarana untuk mendokumentasikan dan menentukan spesifikasi sistem (Mulyani, 2017). Beberapa diagram yang digunakan antara lain: *use case diagram*, *activity diagram*, dan *class diagram*.

3.3.1.1 Use Case Diagram

Use case diagram merupakan diagram yang digunakan untuk memodelkan bagaimana suatu sistem berperilaku, dengan menunjukkan interaksi antara satu atau beberapa aktor yang berperan sebagai pengguna sistem tersebut (Oktarino et al., 2024). *Use case diagram* pada sistem yang sedang dibangun dapat dilihat pada Gambar 3.2 berikut.



Gambar 3.2 Use Case Diagram

Use case diagram pada Gambar 3.2 menggambarkan interaksi antara aktor (Admin dan Siswa) dengan sistem yang dikembangkan. Diagram ini menunjukkan fungsionalitas sistem dari sudut pandang pengguna. Berikut adalah penjelasan dari *use case* pada Gambar 3.2:

1. Melakukan Pendaftaran

Digunakan oleh siswa untuk membuat akun pada sistem.

2. Melakukan *Login*

Digunakan oleh admin dan siswa untuk masuk ke dalam sistem. *Use case* ini *include* ke semua fitur lain karena semua fungsi utama memerlukan autentikasi terlebih dahulu. Untuk admin akan login ke admin *site* bawaan Django.

3. Melihat Informasi Jurusan

Digunakan siswa untuk membaca deskripsi dan prospek kerja dari jurusan yang tersedia.

4. Manajemen Profil

Fitur yang digunakan siswa untuk memperbarui data diri seperti nama, asal sekolah, dan nomor telepon.

5. Melakukan Tes RIASEC

Fitur utama bagi siswa untuk mengerjakan tes minat berdasarkan teori kepribadian RIASEC. *Use case* ini *extend* ke fitur “Melihat Laporan dan Statistik”. Karena laporan dan statistik hanya muncul setelah tes dilakukan.

6. Melihat Laporan dan Statistik

Siswa dapat melihat hasil tes RIASEC berupa skor RIASEC dan jurusan yang direkomendasikan. Terdapat *extend* ke “Mengunduh Laporan dan Statistik”.

7. Mengunduh Laporan dan Statistik

Siswa dapat mengunduh hasil tes dalam format PDF.

8. Manajemen Soal

Admin dapat menambahkan, mengedit, atau menghapus soal tes RIASEC.

9. Manajemen Informasi Jurusan

Admin mengelola data jurusan seperti nama jurusan, deskripsi, dan prospek kerja.

10. Manajemen Klasifikasi RIASEC

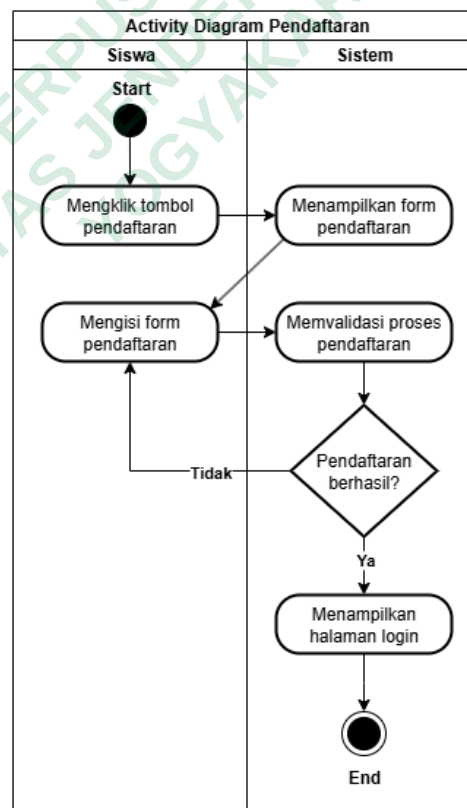
Admin dapat menentukan atau memperbarui vektor RIASEC yang menjadi dasar rekomendasi (*dataset*) untuk masing-masing jurusan.

11. Melakukan *Logout*

Digunakan oleh semua pengguna untuk keluar dari sistem.

3.3.1.2 Activity Diagram

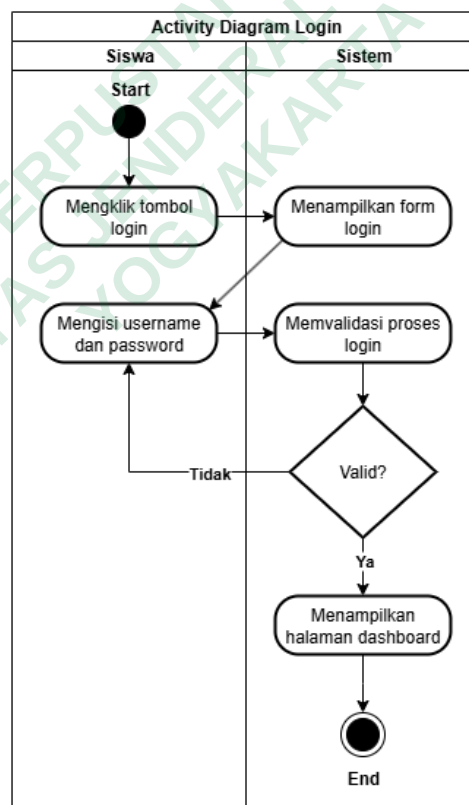
Activity diagram merupakan diagram yang digunakan untuk menggambarkan alur kerja dari suatu proses atau aktivitas. Diagram ini juga dapat merepresentasikan proses bisnis, prosedur, atau alur kerja dalam suatu sistem perangkat lunak (Zein et al., 2023). *Activity* diagram pendaftaran pada sistem ini dapat dilihat pada Gambar 3.3 berikut.



Gambar 3.3 Activity Diagram Pendaftaran

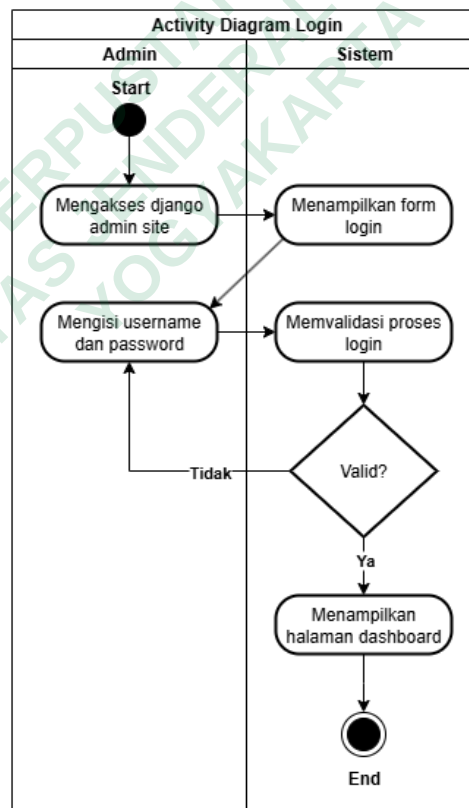
Activity diagram pada Gambar 3.3 menggambarkan alur aktivitas proses pendaftaran calon pengguna (dalam hal ini, siswa) pada SPK pemilihan jurusan kuliah berbasis web. Proses dimulai ketika siswa mengakses sistem dan menekan tombol pendaftaran. Setelah itu, sistem akan merespons dengan menampilkan *form* pendaftaran. Selanjutnya, siswa mengisi *form* pendaftaran sesuai dengan data yang dibutuhkan.

Setelah *form* dikirimkan, sistem melakukan validasi terhadap data pendaftaran. Jika proses validasi berhasil, sistem akan menampilkan halaman login sebagai indikasi bahwa proses pendaftaran selesai dan berhasil dilakukan. Namun jika proses validasi gagal (misalnya, karena data tidak lengkap atau format tidak sesuai), maka sistem akan mengembalikan *form* kepada siswa untuk diperbaiki. Kemudian *Activity* diagram *login* siswa pada sistem ini dapat dilihat pada Gambar 3.4 berikut.



Gambar 3.4 *Activity* Diagram *Login* Siswa

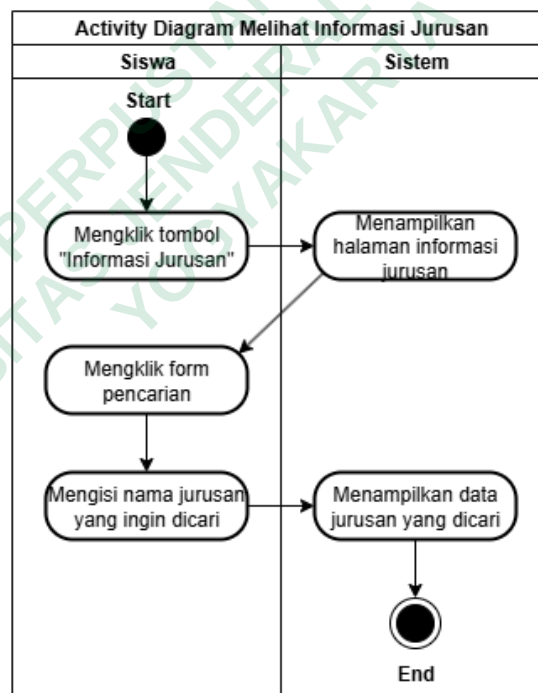
Activity diagram pada Gambar 3.4 menunjukkan alur aktivitas dalam proses *login* yang dilakukan oleh aktor siswa pada sistem. Alur dimulai ketika siswa melakukan tindakan mengklik tombol *login*. Setelah itu, sistem akan menampilkan *form login* yang berisi kolom input untuk memasukkan *username* dan *password*. Pengguna kemudian mengisi kedua kolom tersebut dan mengirimkan data ke sistem. Selanjutnya, sistem melakukan validasi terhadap data *login* yang dimasukkan. Proses validasi mencakup pemeriksaan apakah kombinasi *username* dan *password* sesuai dengan yang tersimpan di dalam basis data sistem. Jika validasi berhasil, sistem akan mengarahkan pengguna ke halaman *dashboard*. Namun, apabila validasi gagal (misalnya karena *username* atau *password* salah), maka sistem akan mengembalikan pengguna ke *form login* untuk melakukan perbaikan data. *Activity* diagram *login* admin pada sistem ini dapat dilihat pada Gambar 3.5 berikut.



Gambar 3.5 *Activity* Diagram *Login* Admin

Activity diagram pada Gambar 3.5 menjelaskan alur proses *login* untuk admin melalui admin *site* bawaan Django. Tidak seperti siswa yang mengakses *form login* dari antarmuka utama sistem, admin harus mengakses halaman *login* admin *site* secara manual dengan menambahkan “/admin” pada URL aplikasi.

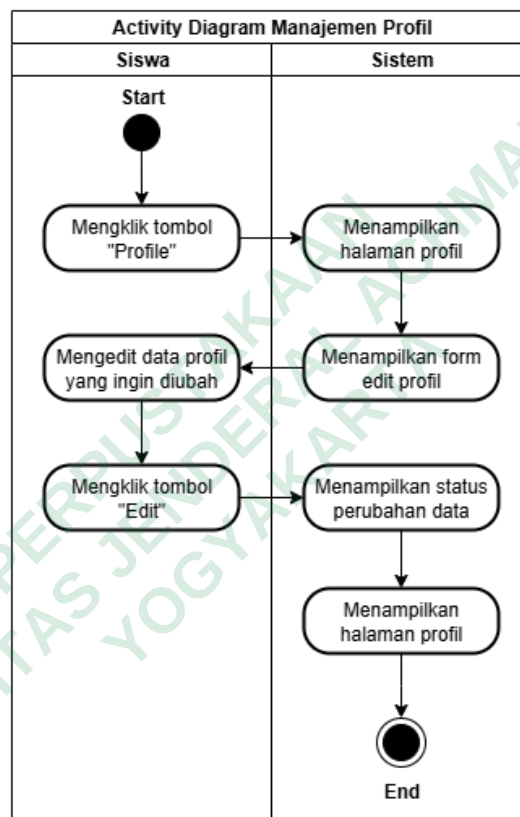
Proses dimulai ketika admin mengakses halaman Django admin *site*, kemudian sistem akan menampilkan *form login* bawaan Django. Admin selanjutnya mengisi *username* dan *password*. Setelah data *login* dikirimkan, sistem akan melakukan proses validasi terhadap informasi yang dimasukkan. Jika data *login* valid, maka admin akan diarahkan menuju halaman *dashboard* admin *site*. Namun, jika proses validasi gagal (misalnya karena *username* atau *password* salah), maka sistem akan menampilkan kembali *form login* disertai notifikasi kesalahan, untuk melakukan perbaikan data. *Activity* diagram melihat informasi jurusan pada sistem ini dapat dilihat pada Gambar 3.6 berikut.



Gambar 3.6 *Activity* Diagram Melihat Informasi Jurusan

Activity diagram pada Gambar 3.6 menggambarkan alur aktivitas yang dilakukan oleh siswa dalam proses mengakses dan mencari informasi jurusan yang tersedia pada sistem. Proses dimulai ketika siswa mengklik tombol "Informasi

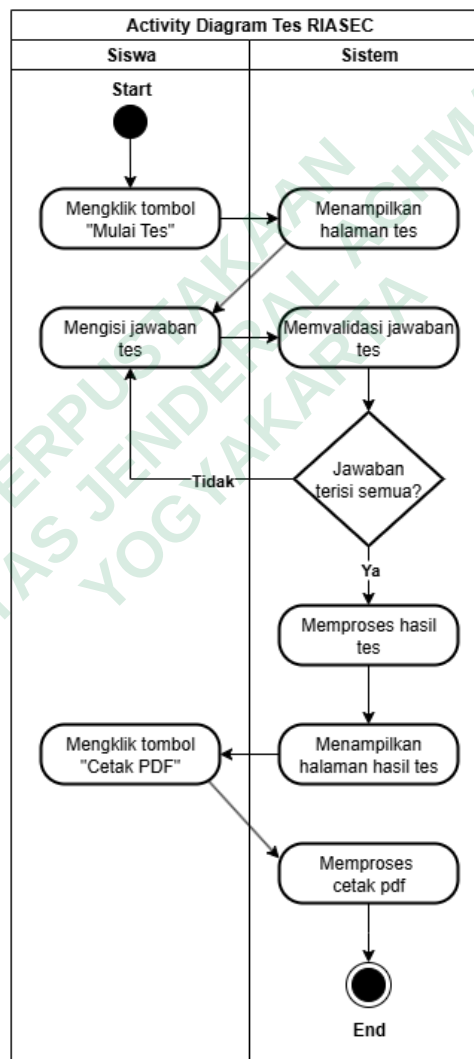
Jurusan" yang terdapat pada *sidebar*. Setelah itu, sistem akan merespons dengan menampilkan halaman informasi jurusan. Selanjutnya, siswa dapat mengakses *form* pencarian untuk mencari jurusan tertentu yang ingin diketahui. Pada *form* tersebut, siswa mengisi nama jurusan yang ingin dicari, kemudian sistem akan memproses permintaan dan menampilkan data jurusan yang sesuai dengan kata kunci pencarian. *Activity* diagram manajemen profil pada sistem ini dapat dilihat pada Gambar 3.7 berikut.



Gambar 3.7 Activity Diagram Manajemen Profil

Activity diagram pada Gambar 3.7 menggambarkan alur aktivitas yang dilakukan oleh siswa dalam melakukan manajemen atau pengelolaan data profil pribadi di dalam sistem. Proses dimulai ketika siswa mengklik tombol "*Profile*" pada *sidebar*. Sistem kemudian akan menampilkan halaman profil yang berisi informasi data diri siswa yang tersimpan.

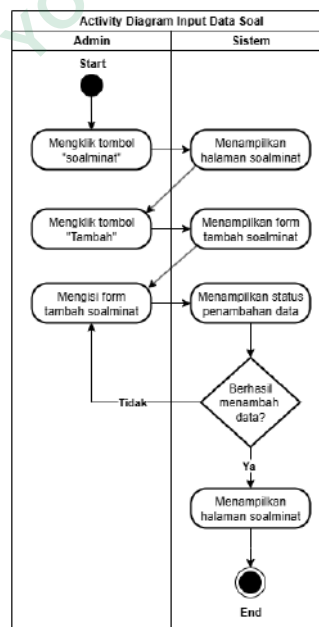
Untuk melakukan perubahan, siswa dapat mengedit data yang ingin diperbarui, seperti nama, email, atau informasi lainnya, melalui *form* yang ditampilkan. Sistem akan merespons dengan menampilkan *form* edit profil yang dapat diisi ulang oleh siswa. Setelah perubahan dilakukan, siswa menekan tombol "Edit" untuk menyimpan perubahan data. Sistem kemudian akan menampilkan status perubahan data (berhasil atau gagal), lalu kembali menampilkan halaman profil yang telah diperbarui. *Activity* diagram Tes RIASEC pada sistem ini dapat dilihat pada Gambar 3.8 berikut.



Gambar 3.8 Activity Diagram Tes RIASEC

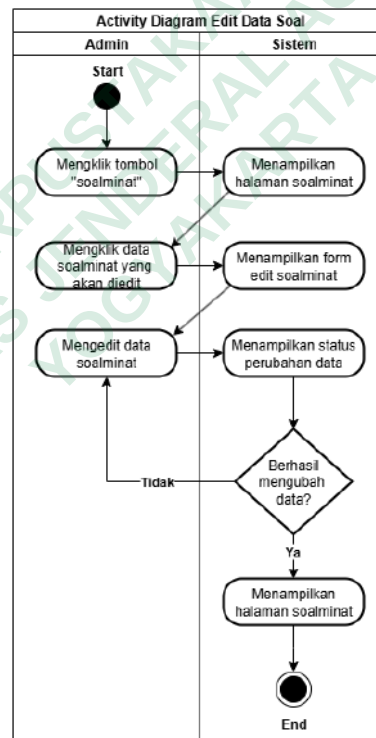
Activity diagram pada Gambar 3.8 menggambarkan alur aktivitas siswa dalam mengikuti tes RIASEC yang merupakan komponen utama dalam SPK pemilihan jurusan kuliah berbasis web. Proses dimulai ketika siswa mengklik tombol "Mulai Tes", kemudian sistem akan menampilkan halaman tes RIASEC yang berisi daftar pernyataan terkait minat dan kepribadian. Siswa kemudian mengisi jawaban tes berdasarkan preferensi masing-masing. Setelah itu, sistem melakukan proses validasi terhadap isian jawaban. Jika ditemukan ada pernyataan yang belum terisi, sistem akan mengembalikan pengguna ke *form* untuk melengkapi jawaban yang masih kosong.

Jika semua jawaban telah terisi dengan lengkap, sistem akan melanjutkan dengan memproses hasil tes menggunakan algoritma klasifikasi yang telah dirancang (K-Nearest Neighbor), lalu menampilkan halaman hasil tes yang menunjukkan kecenderungan kepribadian siswa menurut model RIASEC. Setelah hasil ditampilkan, siswa memiliki opsi untuk mengklik tombol "Cetak PDF". Sistem kemudian akan memproses pencetakan hasil tes dalam format PDF, sehingga siswa dapat menyimpan atau mencetak hasil tersebut sebagai referensi. *Activity* diagram input data soal pada sistem ini dapat dilihat pada Gambar 3.9 berikut.



Gambar 3.9 *Activity* Diagram Input Data Soal

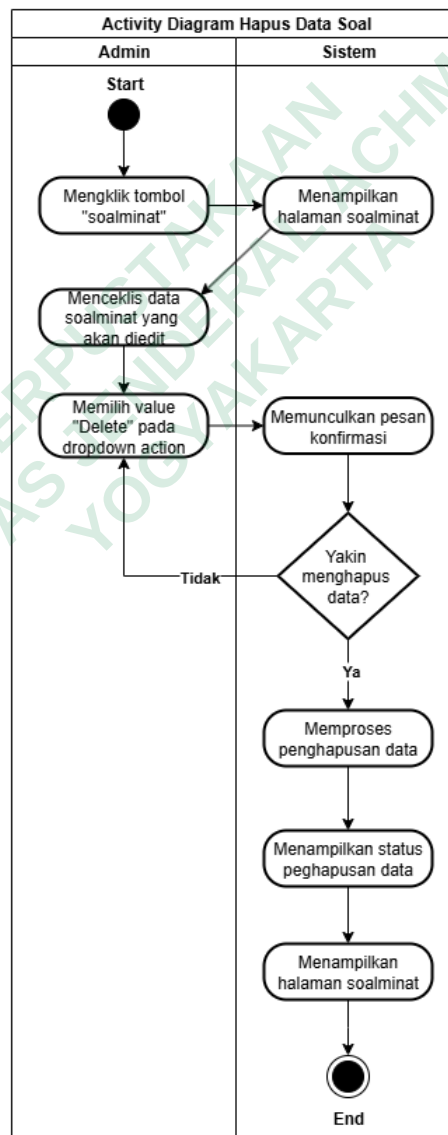
Activity diagram pada Gambar 3.9 menunjukkan alur aktivitas admin dalam melakukan input data soal tes minat RIASEC ke dalam SPK pemilihan jurusan kuliah berbasis web. Proses diawali ketika admin mengklik tombol "soalminat" pada halaman *dashboard* admin. Setelah itu, sistem akan menampilkan halaman soalminat yang berisi daftar soal yang sudah tersedia di dalam sistem. Untuk menambahkan soal baru, admin kemudian mengklik tombol "Tambah", dan sistem akan merespons dengan menampilkan *form* tambah soalminat. Kemudian admin mengisi *form* dengan konten soal baru, termasuk pertanyaan/ Pernyataan dan kategori RIASEC yang sesuai. Setelah *form* dikirimkan, sistem akan memberikan umpan balik berupa status penambahan data (berhasil atau gagal), dan kemudian mengembalikan tampilan ke halaman soalminat. *Activity* diagram edit data soal pada sistem ini dapat dilihat pada Gambar 3.10 berikut.



Gambar 3.10 *Activity* Diagram Edit Data Soal

Activity diagram pada Gambar 3.10 menggambarkan alur aktivitas admin dalam melakukan proses pengeditan atau pembaruan data soal tes minat RIASEC yang sudah tersedia di dalam sistem. Proses dimulai ketika admin mengklik tombol "soalminat" pada *dashboard* admin.

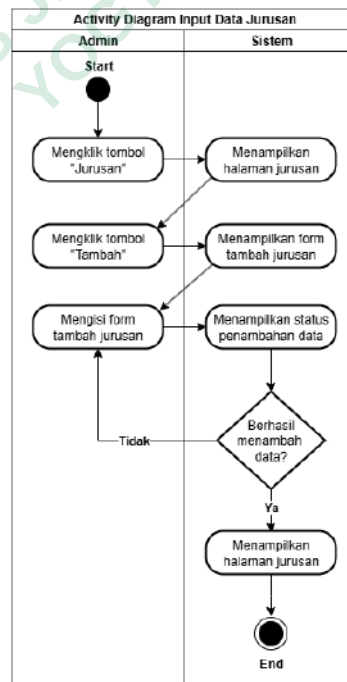
Sistem kemudian akan menampilkan halaman soalminat yang memuat daftar soal yang telah terdaftar di sistem. Selanjutnya, admin memilih data soalminat yang ingin diedit, kemudian sistem akan menampilkan *form* edit soalminat berisi data soal yang telah ada. Admin kemudian melakukan perubahan atau perbaikan terhadap data tersebut, baik dari sisi pertanyaan, maupun kategori RIASEC-nya. Setelah admin menyimpan perubahan, sistem akan memberikan notifikasi berupa status perubahan data (berhasil atau gagal), kemudian jika berhasil akan mengembalikan tampilan ke halaman soalminat yang telah diperbarui. *Activity* diagram hapus data soal pada sistem ini dapat dilihat pada Gambar 3.11 berikut.



Gambar 3.11 *Activity* Diagram Hapus Data Soal

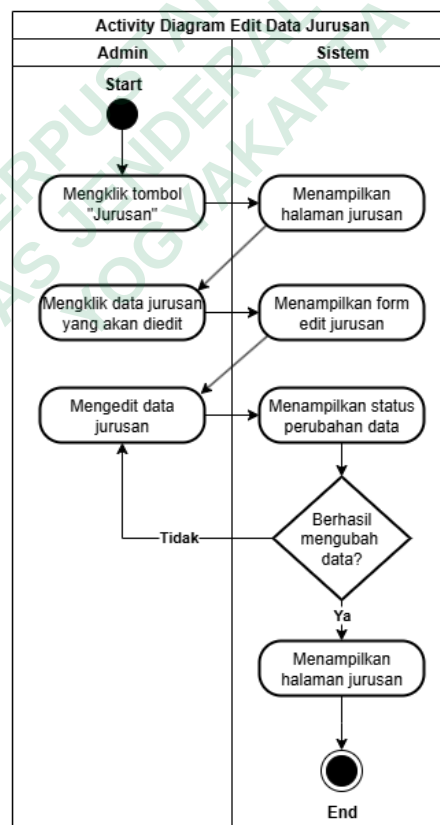
Activity diagram pada Gambar 3.11 menggambarkan proses yang dilakukan oleh admin untuk melakukan penghapusan data soal tes minat RIASEC melalui halaman manajemen soalminat di sistem. Proses dimulai ketika admin mengklik tombol "soalminat" untuk mengakses daftar soal yang tersedia. Sistem kemudian akan menampilkan halaman soalminat, yang berisi seluruh data soal yang sudah terdaftar. Selanjutnya, admin mencentang atau memilih soal yang akan dihapus, kemudian memilih opsi "*Delete*" pada *dropdown action* yang disediakan di antarmuka sistem. Setelah itu, sistem akan menampilkan pesan konfirmasi untuk memastikan tindakan penghapusan.

Jika admin memilih tidak (batal), maka proses dibatalkan dan kembali ke halaman soalminat tanpa ada perubahan. Namun, jika admin memilih ya (konfirmasi penghapusan), sistem akan melanjutkan untuk memproses penghapusan data dari database. Setelah data berhasil dihapus, sistem akan menampilkan status keberhasilan penghapusan, kemudian mengembalikan tampilan ke halaman soalminat yang telah diperbarui tanpa data soal yang telah dihapus. *Activity* diagram input data jurusan pada sistem ini dapat dilihat pada Gambar 3.12 berikut.



Gambar 3.12 *Activity* Diagram Input Data Jurusan

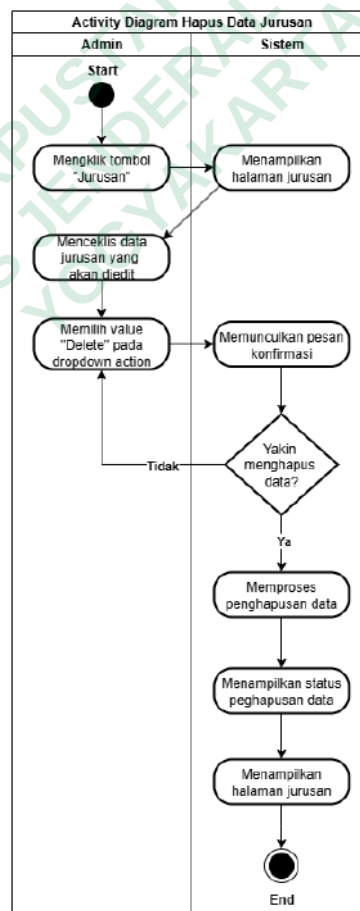
Activity diagram pada Gambar 3.12 menggambarkan alur aktivitas yang dilakukan oleh admin dan sistem dalam proses penambahan data jurusan ke dalam sistem. Proses ini diawali oleh admin yang mengakses halaman jurusan dengan mengklik tombol "Jurusan". Setelah itu, sistem akan menampilkan halaman daftar jurusan yang tersedia. Selanjutnya, admin melanjutkan dengan mengklik tombol "Tambah", yang kemudian memicu sistem untuk menampilkan *form* input data jurusan. Admin kemudian mengisi formulir tersebut dengan informasi yang diperlukan. Setelah formulir diisi dan dikirimkan, sistem akan memproses dan menampilkan status hasil dari penambahan data tersebut, apakah berhasil atau gagal. Jika penambahan berhasil, sistem akan mengarahkan kembali ke halaman daftar jurusan, di mana jurusan baru yang telah ditambahkan akan ditampilkan. *Activity* diagram edit data jurusan pada sistem ini dapat dilihat pada Gambar 3.13 berikut.



Gambar 3.13 *Activity* Diagram Edit Data Jurusan

Activity diagram pada Gambar 3.13 menjelaskan alur proses yang dilakukan oleh admin dan sistem dalam mengedit data jurusan yang sudah ada. Proses dimulai ketika admin mengakses halaman jurusan dengan mengklik tombol "Jurusan", kemudian sistem menampilkan daftar jurusan yang tersedia. Admin kemudian memilih data jurusan yang ingin diedit, dan sistem akan menampilkan *form* edit jurusan yang berisi data jurusan tersebut. Setelah itu, admin melakukan perubahan data sesuai kebutuhan dan mengirimkan perubahan tersebut ke sistem.

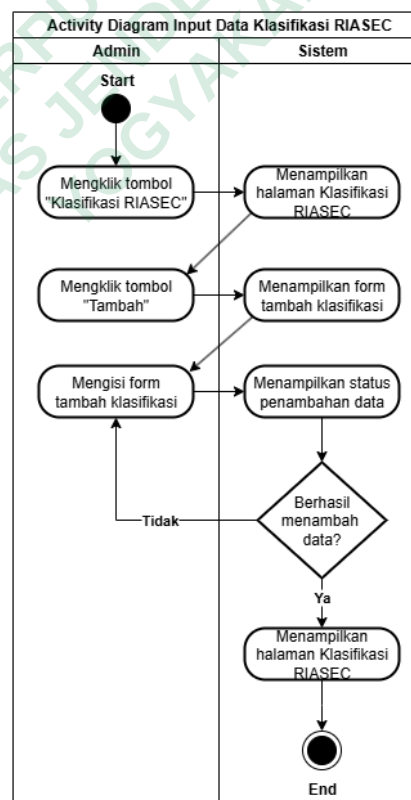
Sistem kemudian memproses perubahan data dan menampilkan status perubahan. Pada tahap ini terdapat percabangan keputusan, yaitu apakah proses perubahan data berhasil atau tidak. Jika perubahan gagal dilakukan, maka admin akan kembali ke langkah pengeditan data. Namun jika perubahan berhasil, sistem akan menampilkan kembali halaman jurusan yang sudah diperbarui. *Activity* diagram hapus data jurusan pada sistem ini dapat dilihat pada Gambar 3.14 berikut.



Gambar 3.14 *Activity* Diagram Hapus Data Jurusan

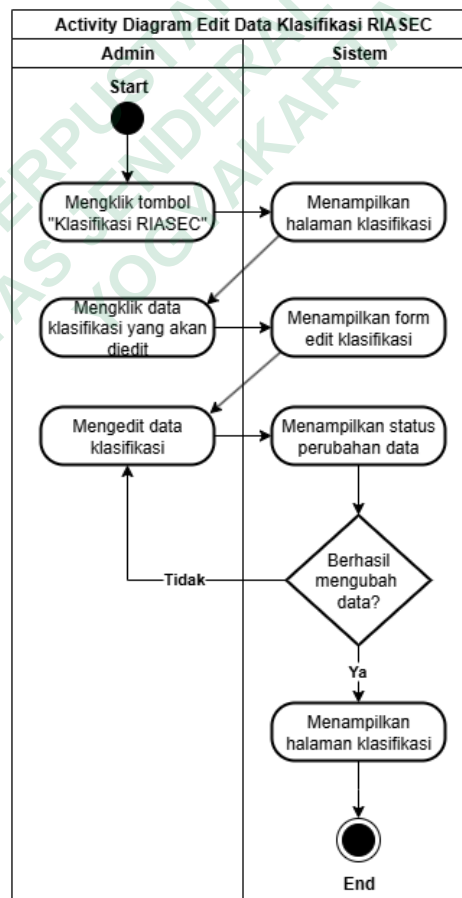
Activity diagram pada Gambar 3.14 menggambarkan alur proses penghapusan data jurusan oleh admin. Proses dimulai ketika admin mengklik tombol "Jurusan" untuk mengakses daftar jurusan yang tersedia. Setelah itu, sistem menampilkan halaman jurusan. Admin kemudian mencentang atau memilih data jurusan yang ingin dihapus, lalu memilih opsi "Delete" dari *dropdown action* yang tersedia. Setelah aksi tersebut dilakukan, sistem akan menampilkan pesan konfirmasi untuk memastikan apakah admin benar-benar ingin menghapus data tersebut.

Jika admin memilih tidak, maka proses dibatalkan dan sistem kembali ke tampilan sebelumnya. Namun jika admin memilih ya, maka sistem akan melanjutkan dengan memproses penghapusan data. Setelah proses penghapusan selesai, sistem akan menampilkan status hasil penghapusan dan kemudian mengarahkan kembali ke halaman jurusan yang telah diperbarui. *Activity* diagram input data klasifikasi RIASEC pada sistem ini dapat dilihat pada Gambar 3.15 berikut.



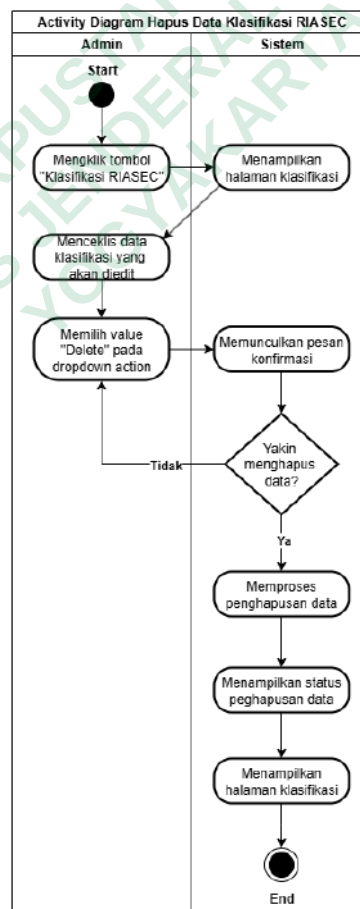
Gambar 3.15 *Activity* Diagram Input Data Klasifikasi RIASEC

Activity diagram pada Gambar 3.15 menunjukkan proses input data klasifikasi RIASEC yang dilakukan oleh admin melalui sistem. Proses diawali dengan admin mengklik tombol "Klasifikasi RIASEC", kemudian sistem menampilkan halaman yang berisi daftar klasifikasi RIASEC yang telah tersedia. Selanjutnya, admin menekan tombol "Tambah" untuk menambahkan data klasifikasi baru. Sistem akan merespons dengan menampilkan *form* input klasifikasi. Admin kemudian mengisi formulir tersebut dengan data yang diperlukan. Setelah *form* dikirim, sistem akan menampilkan status hasil penambahan data. Jika proses penambahan berhasil, sistem akan menampilkan kembali halaman klasifikasi RIASEC dengan data baru yang sudah ditambahkan. Jika tidak berhasil, maka admin diarahkan kembali untuk memperbaiki atau melengkapi data input. *Activity* diagram edit data klasifikasi RIASEC pada sistem ini dapat dilihat pada Gambar 3.16 berikut.



Gambar 3.16 Activity Diagram Edit Data Klasifikasi RIASEC

Activity diagram pada Gambar 3.16 menjelaskan proses yang dilakukan admin dalam melakukan pengeditan data klasifikasi RIASEC pada sistem. Proses dimulai saat admin mengklik tombol "Klasifikasi RIASEC" untuk membuka halaman daftar klasifikasi yang telah tersedia. Sistem kemudian menampilkan halaman klasifikasi tersebut. Admin kemudian memilih data klasifikasi yang akan diedit, dan sistem menampilkan *form* edit klasifikasi yang berisi data yang sudah ada. Setelah itu, admin melakukan pengeditan sesuai kebutuhan. Setelah data diedit dan dikirim, sistem akan menampilkan status perubahan data. Selanjutnya, terdapat percabangan proses berdasarkan keberhasilan perubahan data. Jika proses pengeditan gagal, sistem akan mengarahkan kembali admin untuk memperbaiki data. Namun jika berhasil, sistem akan menampilkan halaman klasifikasi yang telah diperbarui. *Activity* diagram hapus data klasifikasi RIASEC pada sistem ini dapat dilihat pada Gambar 3.17 berikut.



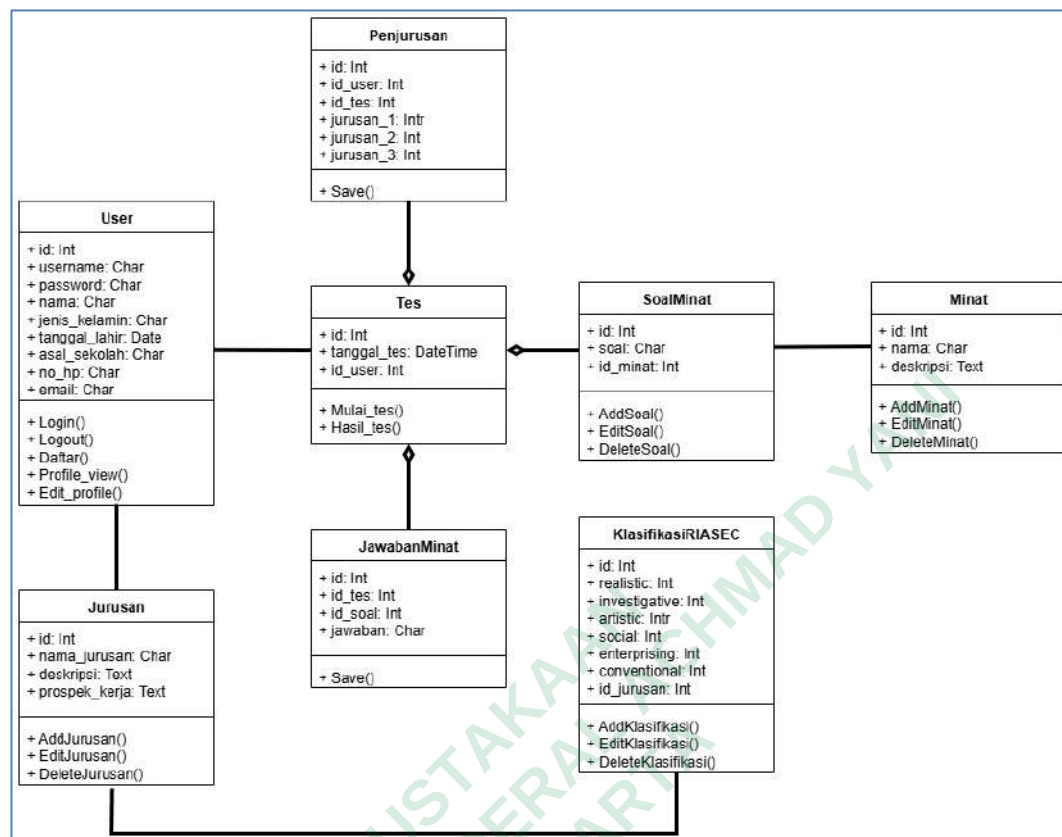
Gambar 3.17 Activity Diagram Hapus Data Klasifikasi RIASEC

Activity diagram pada Gambar 3.17 menggambarkan alur aktivitas yang dilakukan oleh admin dalam proses penghapusan data klasifikasi RIASEC pada sistem. Proses dimulai ketika admin mengklik tombol "Klasifikasi RIASEC" untuk membuka halaman daftar klasifikasi. Sistem kemudian menampilkan halaman klasifikasi yang tersedia. Admin kemudian mencentang data klasifikasi yang ingin dihapus dan memilih opsi "*Delete*" pada menu *dropdown* aksi. Setelah itu, sistem akan menampilkan pesan konfirmasi untuk memastikan bahwa penghapusan dilakukan secara sadar dan tidak disengaja.

Jika admin membatalkan proses, maka proses dihentikan dan sistem kembali menampilkan halaman klasifikasi. Jika admin mengonfirmasi penghapusan, maka sistem akan memproses penghapusan data, menampilkan status hasil penghapusan, dan akhirnya menampilkan kembali halaman klasifikasi yang telah diperbarui.

3.3.1.3 *Class Diagram*

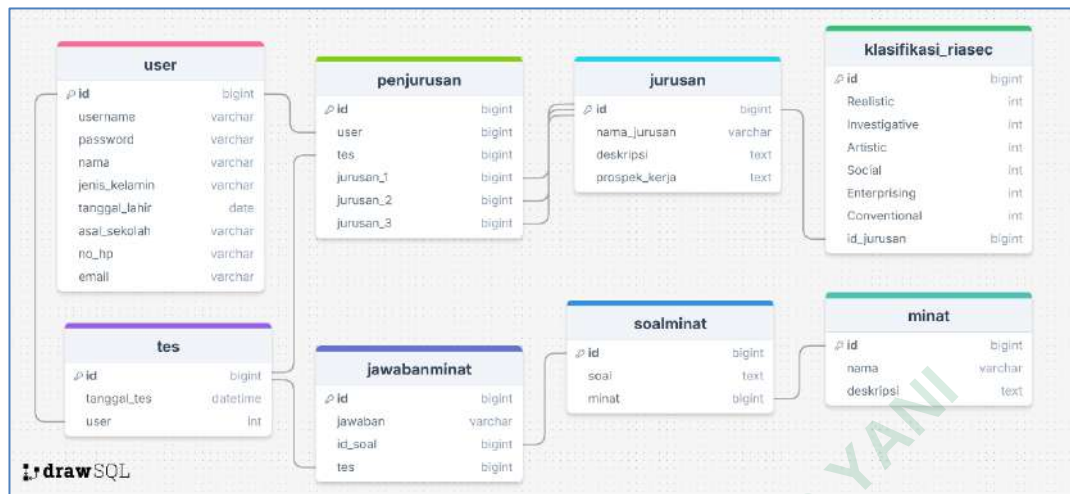
Class diagram merupakan diagram yang menampilkan kumpulan *class* dalam suatu sistem beserta hubungan logis di antara *class-class* tersebut. Diagram ini digunakan untuk merepresentasikan struktur statis dari sistem yang sedang dikembangkan (Cahyani et al., 2020). *Class* diagram pada sistem ini dapat dilihat pada Gambar 3.18 berikut.



Gambar 3.18 Class Diagram

3.3.2 Perancangan Basis Data

Perancangan basis data sistem dilakukan dengan menggunakan model relasional yang disesuaikan dengan kebutuhan sistem. *Database* relasional adalah sebuah model penyimpanan data yang menyajikan informasi kepada pengguna dalam format tabel dua dimensi, di mana data diorganisasikan dalam bentuk baris dan kolom untuk merepresentasikan suatu kumpulan data atau file (Kristiyanto et al., 2023). Dalam basis data relasional, terdapat beberapa jenis kunci yang umum digunakan, antara lain *primary key* (kunci primer) dan *foreign key* (kunci asing). *Primary key* berfungsi untuk menjamin keunikan setiap baris data dalam sebuah tabel, sementara *foreign key* digunakan untuk membangun relasi antara dua tabel yang berbeda (Purbasari et al., 2024). Rancangan basis data pada sistem ini ditunjukkan pada Gambar 3.19 berikut.



Gambar 3.19 Rancangan Basis Data

Berikut penjelasan struktur basis data yang ditunjukkan oleh Gambar 3.19:

1. Tabel *User*

Merupakan tabel yang digunakan untuk menyimpan data pengguna seperti nama, jenis kelamin, asal sekolah, tanggal lahir, dan lain-lain. Struktur tabel *user* dapat dilihat pada Tabel 3.1 berikut.

Tabel 3.1 Tabel *User*

<i>Field</i>	Type Data	Keterangan
<i>id</i>	<i>big integer</i>	<i>primary key</i>
<i>username</i>	<i>varchar</i>	
<i>password</i>	<i>varchar</i>	
<i>nama</i>	<i>varchar</i>	
<i>jenis_kelamin</i>	<i>varchar</i>	
<i>tanggal_lahir</i>	<i>date</i>	
<i>asal_sekolah</i>	<i>varchar</i>	
<i>no_hp</i>	<i>varchar</i>	
<i>email</i>	<i>varchar</i>	

2. Tabel Tes

Merupakan tabel yang digunakan untuk menyimpan data setiap pelaksanaan tes oleh pengguna. Struktur tabel tes dapat dilihat pada Tabel 3.2 berikut.

Tabel 3.2 Tabel Tes

<i>Field</i>	<i>Tipe Data</i>	<i>Keterangan</i>
id	<i>big integer</i>	<i>primary key</i>
tanggal_tes	<i>datetime</i>	
user	<i>big integer</i>	<i>foreign key</i>

3. Tabel SoalMinat

Merupakan tabel yang digunakan untuk menyimpan soal-soal RIASEC yang dikaitkan dengan salah satu aspek minat. Struktur tabel soalminat dapat dilihat pada Tabel 3.3 berikut.

Tabel 3.3 Tabel SoalMinat

<i>Field</i>	<i>Tipe Data</i>	<i>Keterangan</i>
id	<i>big integer</i>	<i>primary key</i>
soal	<i>text</i>	
minat	<i>big integer</i>	<i>foreign key</i>

4. Tabel JawabanMinat

Merupakan tabel yang digunakan untuk menyimpan jawaban “Ya” atau “Tidak” terhadap masing-masing soal. Struktur tabel jawabanminat dapat dilihat pada Tabel 3.4 berikut.

Tabel 3.4 Tabel JawabanMinat

<i>Field</i>	<i>Tipe Data</i>	<i>Keterangan</i>
id	<i>big integer</i>	<i>primary key</i>
jawaban	<i>varchar</i>	
id_soal	<i>big integer</i>	<i>foreign key</i>
tes	<i>big integer</i>	<i>foreign key</i>

5. Tabel Minat

Merupakan tabel yang berisi deskripsi enam aspek minat sesuai teori kepribadian RIASEC. Struktur tabel minat dapat dilihat pada Tabel 3.5 berikut.

Tabel 3.5 Tabel Minat

<i>Field</i>	Tipe Data	Keterangan
id	<i>big integer</i>	<i>primary key</i>
nama	<i>varchar</i>	
deskripsi	<i>text</i>	

6. Tabel Jurusan

Merupakan tabel yang digunakan untuk menyimpan data jurusan, deskripsi, dan prospek kerja. Struktur tabel jurusan dapat dilihat pada Tabel 3.6 berikut.

Tabel 3.6 Tabel Jurusan

<i>Field</i>	Tipe Data	Keterangan
id	<i>big integer</i>	<i>primary key</i>
nama_jurusan	<i>varchar</i>	
deskripsi	<i>text</i>	
prospek_kerja	<i>text</i>	

7. Tabel KlasifikasiRIASEC

Merupakan tabel yang digunakan untuk menyimpan data vektor skor RIASEC untuk setiap jurusan, yang digunakan dalam proses klasifikasi menggunakan KNN. Struktur tabel ini dapat dilihat pada Tabel 3.7 berikut.

Tabel 3.7 Tabel KlasifikasiRIASEC

<i>Field</i>	<i>Tipe Data</i>	<i>Keterangan</i>
id	<i>big integer</i>	<i>primary key</i>
<i>realistic</i>	<i>integer</i>	
<i>investigative</i>	<i>integer</i>	
<i>artistic</i>	<i>integer</i>	
<i>social</i>	<i>integer</i>	
<i>enterprising</i>	<i>integer</i>	
<i>conventional</i>	<i>integer</i>	
id_jurusan	<i>big integer</i>	<i>foreign key</i>

8. Tabel Penjurusan

Merupakan tabel yang digunakan untuk menyimpan hasil rekomendasi jurusan berdasarkan tes yang dilakukan pengguna. Struktur tabel ini dapat dilihat pada Tabel 3.8 berikut.

Tabel 3.8 Tabel Penjurusan

<i>Field</i>	<i>Tipe Data</i>	<i>Keterangan</i>
id	<i>big integer</i>	<i>primary key</i>
<i>user</i>	<i>big integer</i>	<i>foreign key</i>
tes	<i>big integer</i>	<i>foreign key</i>
jurusan_1	<i>big integer</i>	<i>foreign key</i>
jurusan_2	<i>big integer</i>	<i>foreign key</i>
jurusan_3	<i>big integer</i>	<i>foreign key</i>

3.3.3 Perancangan Antarmuka Pengguna

Perancangan Antarmuka Pengguna (*User Interface Design/UI Design*) merupakan proses merancang tampilan visual dan elemen interaktif dari suatu sistem atau aplikasi dengan tujuan mempermudah pengguna dalam mengoperasikannya (Tajrin et al., 2025). Perancangan antarmuka dimaksudkan untuk membangun interaksi yang efektif, efisien, dan nyaman antara pengguna dan sistem komputer.

3.3.3.1 Desain *Landing Page*

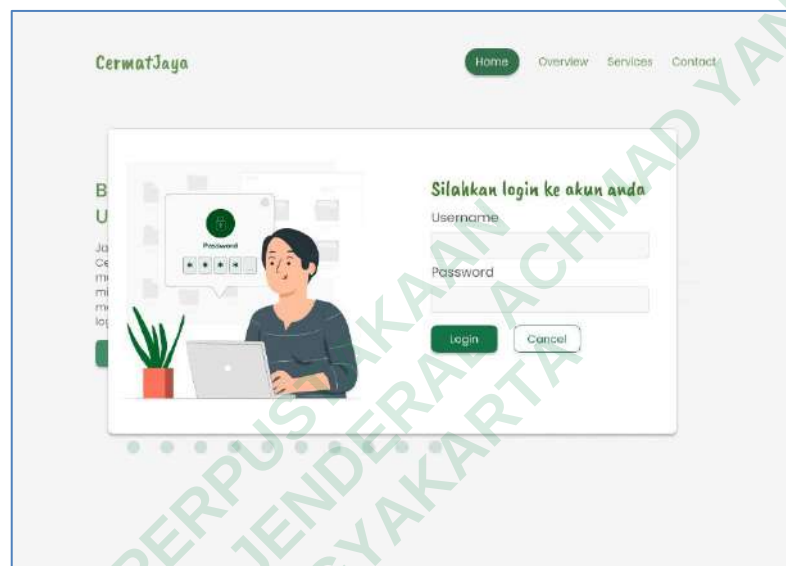
Halaman ini menampilkan informasi umum dan akses menuju *pop-up login* serta halaman pendaftaran pengguna baru. Rancangan desain *landing page* pada sistem ini ditunjukkan pada Gambar 3.20 berikut.



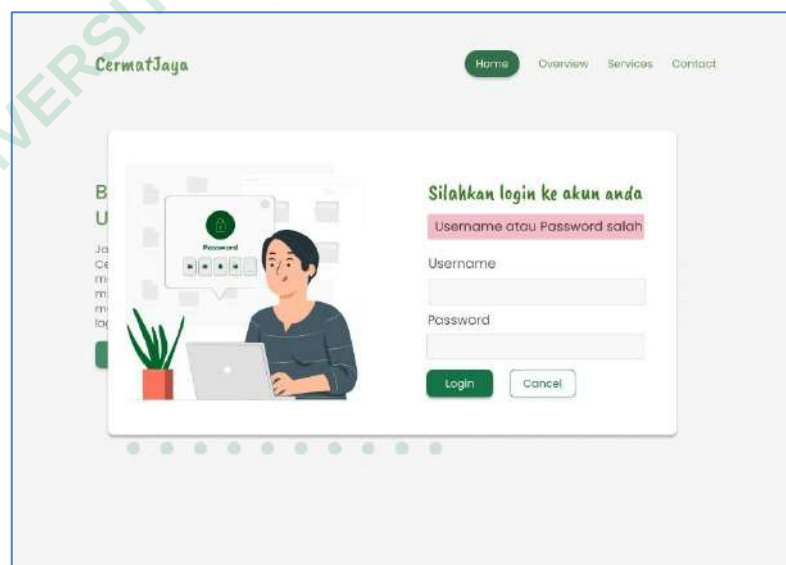
Gambar 3.20 Desain Antarmuka *Landing Page*

3.3.3.2 Desain Pop-Up Login

Pop-up login dirancang untuk memudahkan pengguna dalam mengakses sistem secara cepat dan langsung dari *landing page* tanpa perlu berpindah halaman. *Pop-up* ini dirancang untuk menampilkan *form* autentikasi yang terdiri dari input *username* dan *password*, tombol aksi (*Login* dan *Cancel*), serta ilustrasi pendukung untuk memperkuat konteks visual. Gambar 3.21 menampilkan antarmuka *pop-up login* yang muncul pada halaman utama ketika pengguna menekan tombol "Login".



Gambar 3.21 Desain Antarmuka *Pop-Up Login*



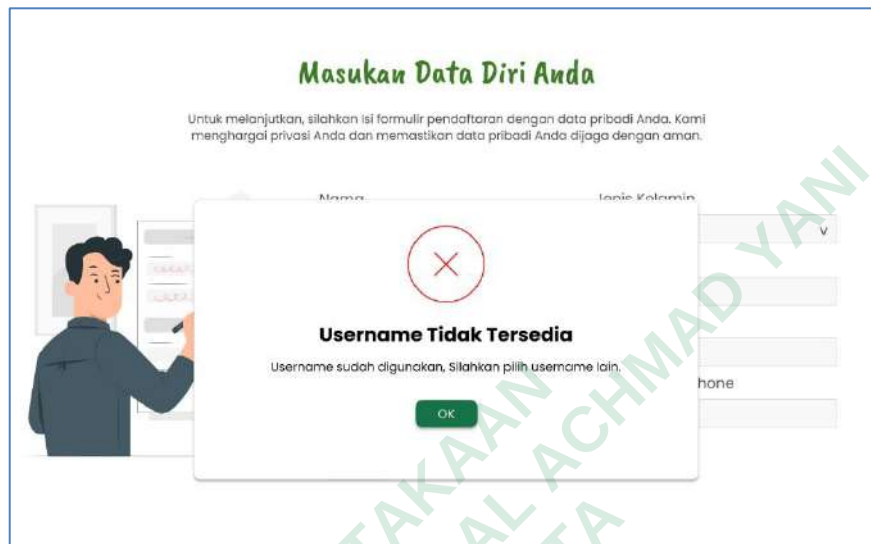
Gambar 3.22 Desain Antarmuka *Pop-Up Login* (Kesalahan Autentikasi)

Halaman pendaftaran dirancang agar pengguna dapat membuat akun dengan mengisi data pribadi yang diperlukan. Formulir pendaftaran ini terdiri dari beberapa komponen input, yaitu: nama, jenis kelamin, tanggal lahir, asal sekolah, email, nomor handphone, *username*, dan *password*. Selain itu, terdapat dua tombol aksi utama yaitu tombol “Daftar” untuk mengirimkan data, dan “Cancel” untuk membatalkan proses pendaftaran. Rancangan desain halaman pendaftaran pada sistem ini ditunjukkan pada Gambar 3.23 berikut.

Gambar 3.23 Desain Antarmuka Halaman Pendaftaran

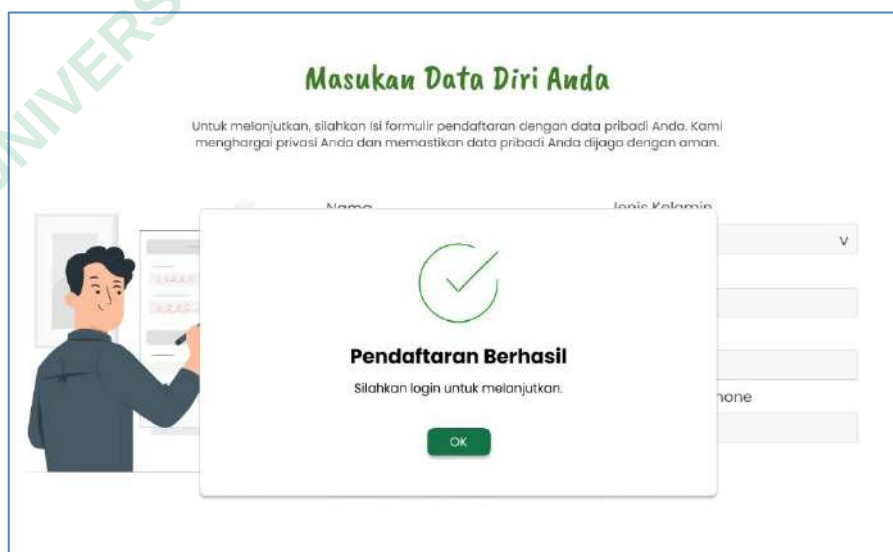
Gambar 3.23 Desain Antarmuka Halaman Pendaftaran

Pada Gambar 3.24 ditunjukkan proses pendaftaran yang gagal akibat *username* yang sudah digunakan oleh pengguna lain. Sistem secara otomatis menampilkan pesan peringatan berupa *pop-up* "*Username Tidak Tersedia*" sehingga pengguna dapat segera mengganti *username* tersebut dengan yang lain.



Gambar 3.24 Desain Antarmuka *Pop-Up* Pendaftaran Gagal

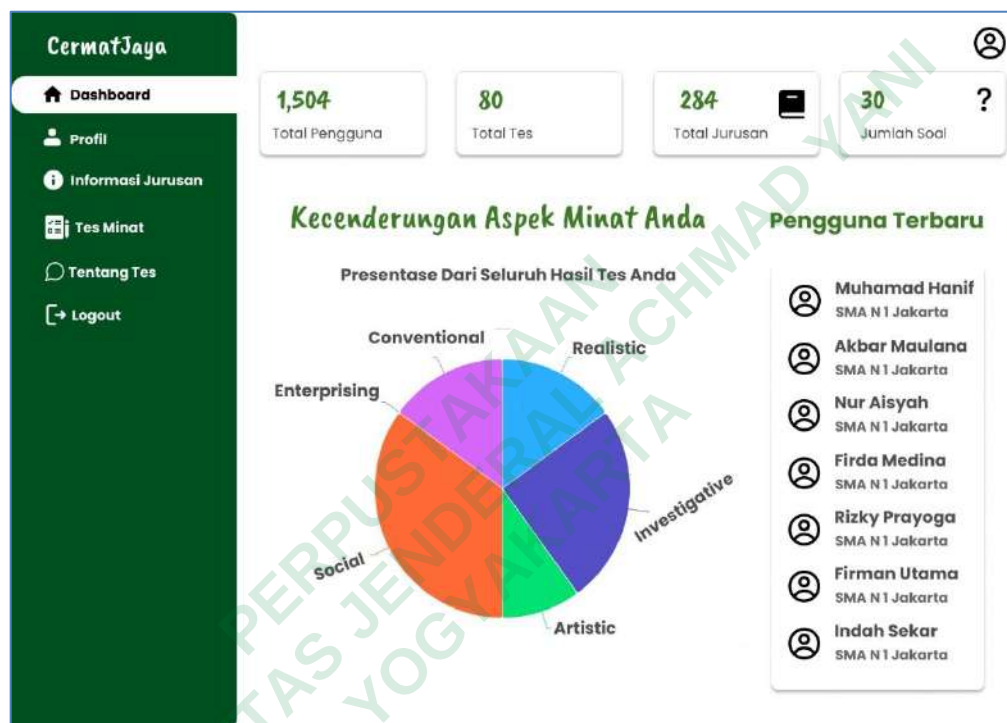
Selanjutnya, pada Gambar 3.25 diperlihatkan proses pendaftaran yang berhasil. Setelah seluruh data diisi dengan benar dan *username* tersedia, sistem menampilkan *pop-up* "Pendaftaran Berhasil" dan mengarahkan pengguna untuk *login* ke dalam sistem.



Gambar 3.25 Desain Antarmuka *Pop-Up* Pendaftaran Berhasil

3.3.3.4 Desain Halaman *Dashboard*

Halaman *dashboard* merupakan tampilan utama yang ditampilkan setelah pengguna berhasil *login* ke dalam sistem. *Dashboard* dirancang untuk menyajikan ringkasan informasi secara cepat, menarik, dan mudah dipahami oleh pengguna. Rancangan desain halaman *dashboard* pada sistem ini ditunjukkan pada Gambar 3.26 berikut.



Gambar 3.26 Desain Halaman *Dashboard*

Pada Gambar 3.26 ditampilkan desain halaman *dashboard* yang memuat beberapa informasi, seperti total pengguna dalam sistem, total tes dalam sistem, total informasi jurusan yang ada di sistem, serta jumlah soal pada tes minat. Selain itu, ditampilkan pula grafik *pie* yang menunjukkan kecenderungan aspek minat berdasarkan hasil tes minat pengguna. Grafik ini mencakup enam aspek minat utama yaitu *realistic*, *investigative*, *artistic*, *social*, *enterprising*, dan *conventional*, lengkap dengan persentasenya. Di sisi kanan layar, ditampilkan daftar pengguna terbaru yang telah melakukan pendaftaran, termasuk nama dan asal sekolah masing-masing pengguna. Hal ini memberikan gambaran bahwa sistem terus berkembang dan digunakan oleh berbagai kalangan.

3.3.3.5 Desain Halaman Profil

Halaman ini memungkinkan pengguna untuk melihat dan mengedit informasi pribadi mereka. Rancangan desain halaman profil pada sistem ini ditunjukkan pada Gambar 3.27 berikut.

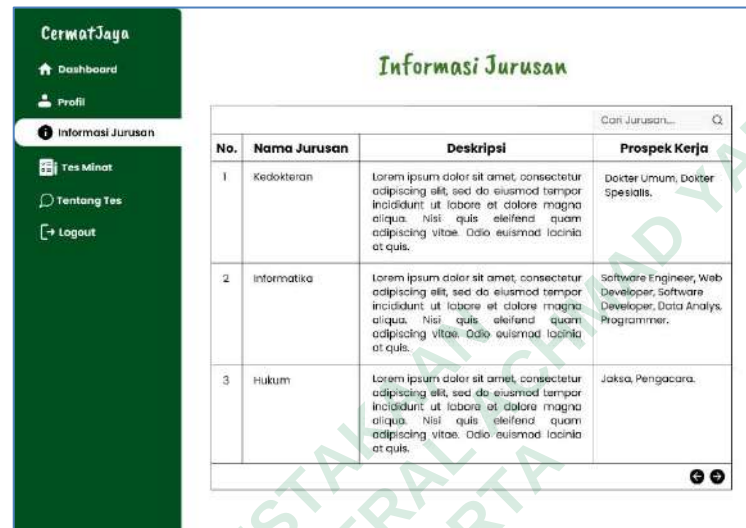
Gambar 3.27 Desain Halaman Profil

Pengguna dapat melakukan perubahan dengan menekan tombol “Edit”, yang kemudian akan memunculkan dialog konfirmasi seperti yang ditunjukkan pada Gambar 3.28. Dialog tersebut menanyakan apakah pengguna yakin untuk menyimpan perubahan dengan dua pilihan tombol yaitu "Ya, Simpan!" dan "Cancel".

Gambar 3.28 Desain Dialog Konfirmasi Halaman Profil

3.3.3.6 Desain Halaman Informasi Jurusan

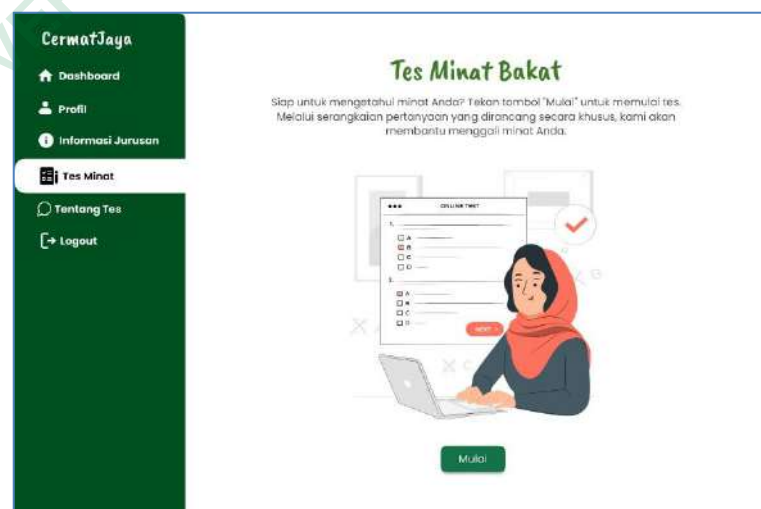
Halaman ini menyajikan daftar jurusan beserta deskripsi dan prospek kerja dari masing-masing jurusan. Pengguna juga dapat melakukan pencarian jurusan menggunakan kolom pencarian di pojok kanan atas tabel. Rancangan desain halaman informasi jurusan pada sistem ini ditunjukkan pada Gambar 3.29 berikut.



Gambar 3.29 Desain Halaman Informasi Jurusan

3.3.3.7 Desain Halaman Tes Minat

Halaman tes minat merupakan fitur utama dalam sistem yang dirancang untuk membantu pengguna menggali potensi dan minat mereka. Rancangan desain halaman tes minat pada sistem ini ditunjukkan pada Gambar 3.30 berikut.



Gambar 3.30 Desain Halaman Tes Minat

Pada Gambar 3.30 ditampilkan desain halaman awal tes minat yang memuat informasi pengantar mengenai tujuan tes. Teks pengantar menjelaskan bahwa pengguna akan mengikuti serangkaian pertanyaan untuk mengetahui minat mereka. Ilustrasi visual di bagian tengah menampilkan seseorang yang sedang mengikuti tes secara *online*, menciptakan kesan yang bersahabat dan mendorong pengguna untuk mulai mengikuti tes.

Di bawah ilustrasi, terdapat tombol “Mulai” yang berfungsi untuk mengarahkan pengguna ke halaman pertanyaan. Setelah tombol “Mulai” ditekan, pengguna diarahkan ke tampilan halaman pertanyaan seperti yang ditampilkan pada Gambar 3.31 berikut.

The image shows a web application interface for 'CeramatJaya'. On the left is a dark green sidebar with white text and icons for navigation: 'Dashboard', 'Profil', 'Informasi Jurusan', 'Tes Minat' (which is highlighted with a white background), 'Tentang Tes', and 'Logout'. The main area of the page is white. In the center, there is a form titled 'Tes Minat' with three questions, each followed by two radio button options: 'Ya' (Yes) and 'Tidak' (No). The questions are: 'Saya suka membantu orang.' (I like helping people), 'Saya ingin memulai bisnis.' (I want to start a business), and 'Saya suka menyanyi.' (I like singing). Below the questions is a large green button with the word 'Submit' in white text. A large, faint watermark 'UNIVERSITAS PETER LACHMAD' is visible diagonally across the page.

Gambar 3.31 Desain Halaman Tes Minat Dimulai

3.3.3.8 Desain Halaman Hasil Tes

Halaman hasil tes merupakan tampilan akhir setelah pengguna menyelesaikan tes minat. Halaman ini dirancang untuk menampilkan hasil analisis tes secara ringkas, informatif, dan mudah dipahami, guna membantu pengguna dalam mengenali potensi minat mereka. Rancangan desain halaman hasil tes pada sistem ini ditunjukkan pada Gambar 3.32 berikut.

CermatJaya

- Dashboard
- Profil
- Informasi Jurusan
- Tes Minat**
- Tentang Tes
- Logout

Hasil Tes

Tanggal Tes: Dec. 8, 2023, 7:55 pm.

Aspek RIASEC	Skor
Realistic	2
Investigative	3
Artistic	1
Social	5
Enterprising	3
Conventional	7

Jurusan Terpilih: SI-Informatika, D3-Kebidanan, dan SI-Hukum

Deskripsi Aspek Minat: Aspek minat anda adalah conventional (Tipe Administratif). Anda cenderung suka dengan pekerjaan yang melibatkan kegiatan administratif dan pengaturan. Menyukai rutinitas, mengordinir data, dan bekerja dengan informasi terstruktur mungkin merupakan keahlian anda.

[Cetak PDF](#)

Gambar 3.32 Desain Halaman Hasil Tes

Pada Gambar 3.32 ditampilkan halaman hasil tes yang mencakup beberapa elemen penting, yaitu tanggal pelaksanaan tes, tabel skor RIASEC, jurusan terpilih, deskripsi kecenderungan aspek minat, serta tombol untuk mencetak hasil tes dalam format pdf. Rancangan desain hasil cetak pdf dari data hasil tes pada sistem ini ditunjukkan pada Gambar 3.33 berikut.

Hasil Tes

Tanggal Tes: Dec. 8, 2023, 7:55 pm.

Aspek RIASEC	Skor
Realistic	2
Investigative	3
Artistic	1
Social	5
Enterprising	3
Conventional	7

Jurusan Terpilih: SI-Informatika, D3-Kebidanan, dan SI-Hukum

Deskripsi Aspek Minat: Aspek minat anda adalah conventional (Tipe Administratif). Anda cenderung suka dengan pekerjaan yang melibatkan kegiatan administratif dan pengaturan. Menyukai rutinitas, mengordinir data, dan bekerja dengan informasi terstruktur mungkin merupakan keahlian anda.

@username-CermatJaya.com

Gambar 3.33 Desain Hasil Cetak PDF

3.3.3.9 Desain Halaman Tentang Tes

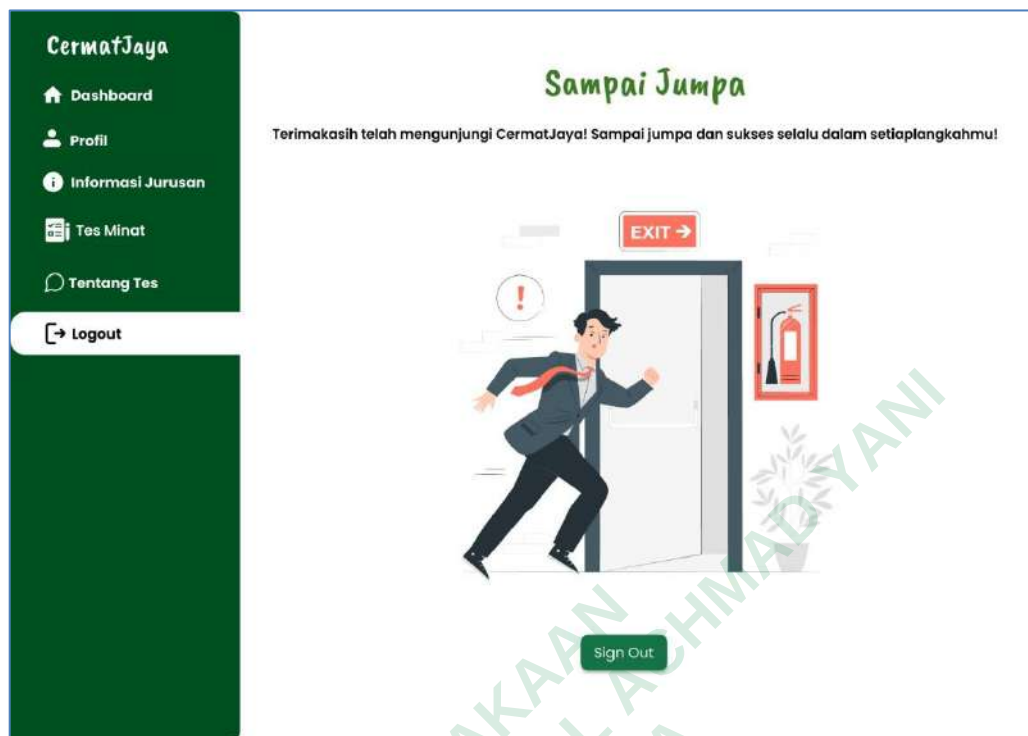
Halaman ini dirancang untuk memberikan penjelasan komprehensif mengenai dasar teori yang digunakan dalam pelaksanaan tes minat pengguna. Isi utama halaman menjelaskan bahwa teori yang digunakan dalam tes ini adalah teori kepribadian RIASEC atau Holland Codes, yang dikembangkan oleh psikolog asal Amerika Serikat, John Holland. Rancangan desain halaman tentang tes pada sistem ini ditunjukkan pada Gambar 3.34 berikut.



Gambar 3.34 Desain Halaman Tentang Tes

3.3.3.10 Desain Halaman Logout

Halaman ini dirancang dengan tampilan yang sederhana namun tetap komunikatif dan menyampaikan pesan dengan jelas kepada pengguna. Visual utama pada halaman ini adalah ilustrasi seorang pria yang tampak berlari ke arah pintu keluar dengan tulisan "EXIT" di atasnya. Ilustrasi ini merepresentasikan aksi keluar dari sistem. Rancangan desain halaman *logout* pada sistem ini ditunjukkan pada Gambar 3.35 berikut.



Gambar 3.35 Desain Halaman *Logout*

Di bagian bawah terdapat tombol *sign out* berwarna hijau yang mudah dikenali dan menjadi elemen interaktif utama pada halaman ini. Tombol ini menegaskan fungsionalitas halaman, yaitu sebagai titik akhir atau keluar dari sesi penggunaan sistem.