

BAB 3

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Penelitian ini menerapkan pendekatan pengembangan sistem yang terstruktur dan sistematis. Metodologi ini dipilih untuk memastikan alur kerja yang jelas dan berurutan, di mana setiap fase pengembangan dilakukan secara tuntas dan terdokumentasi sebelum melanjutkan ke fase berikutnya. Pendekatan ini menyediakan kerangka kerja yang kokoh untuk analisis, perancangan, implementasi, dan pengujian sistem pertukaran jadwal

1. Tinjauan Pustaka

Tahapan ini merupakan fondasi dari penelitian, di mana dilakukan identifikasi dan analisis terhadap penelitian terdahulu, artikel ilmiah, dan literatur lain yang relevan dengan topik yang diangkat. Kegiatan utama pada tahap ini adalah untuk membangun pemahaman yang mendalam mengenai lanskap penelitian yang ada, mengidentifikasi celah penelitian, serta membangun landasan teoretis yang kokoh untuk perancangan dan implementasi sistem.

2. Analisis Kebutuhan

Pada tahap ini, dilakukan analisis mendalam terhadap proses sistem yang sedang berjalan untuk memahami alur kerja dan mengidentifikasi kelemahannya. Berdasarkan analisis tersebut, dirumuskan serangkaian kebutuhan yang akan dikembangkan.

3. Perancangan Sistem

Setelah dilakukannya analisis terhadap kebutuhan dan sistem yang ada, pada tahap ini dibangun perancangan untuk gambaran dasar dari sistem yang akan dibuat. Antarmuka, fungsionalitas dan rancangan media penyimpanan (*database*) juga dilakukan pada tahap ini.

4. Implementasi Sistem

Berdasarkan dokumen perancangan yang telah dibuat, tahap implementasi adalah proses realisasi fisik dari sistem melalui penulisan kode program. Kegiatan ini meliputi pengembangan sisi server (*backend*) termasuk implementasi algoritma inti, pengembangan sisi klien (*frontend*) untuk interaksi dengan pengguna, serta pembuatan dan konfigurasi basis data.

5. Pengujian dan Evaluasi Sistem

Setelah sistem selesai dibangun, dilakukan serangkaian pengujian dan evaluasi untuk memastikan kualitas dan kesesuaian sistem dengan kebutuhan. Pengujian mencakup verifikasi fungsionalitas untuk memvalidasi bahwa setiap fitur berjalan sesuai harapan, serta evaluasi kinerja sistem berdasarkan metrik yang telah ditetapkan pada landasan teori untuk mengukur performa luaran algoritma.

6. Dokumentasi

Tahap dokumentasi merupakan proses yang berjalan secara paralel di sepanjang penelitian. Seluruh kegiatan, keputusan desain, hasil analisis, dan temuan pengujian dicatat dan disusun secara sistematis. Tujuannya adalah untuk menghasilkan laporan akhir yang komprehensif, dapat direplikasi, dan secara jelas mengkomunikasikan seluruh rangkaian proses dan hasil penelitian yang telah dicapai.

3.2 Analisis Sistem

Hasil analisis kebutuhan untuk sistem pertukaran jadwal perkuliahan ini dapat dibagi menjadi dua area utama. Pertama, adalah kebutuhan *input* dan *output* sistem. Kebutuhan *input* esensial dari sistem ini berupa data autentikasi pengguna (NIM dan *password*), data kelas yang dimiliki pengguna dan hendak ditukarkan, serta satu set data berisi daftar kelas lain yang diinginkan sebagai preferensi. Adapun kebutuhan *output* atau luaran utama dari sistem adalah informasi hasil alokasi akhir bagi setiap pengguna, yang berisi data kelas baru yang berhasil didapatkan atau notifikasi bahwa pertukaran tidak ditemukan.

Kedua, adalah analisis dari segi fungsionalitas dan peran pengguna yang berinteraksi dengan sistem. Berdasarkan observasi terhadap sistem informasi akademik pada umumnya, dibutuhkan minimal dua jenis hak akses, yaitu hak sebagai admin dan mahasiswa. Fungsionalitas sistem yang diharapkan harus dapat mengakomodasi proses autentikasi pengguna. Pengguna dengan peran mahasiswa harus dapat melihat daftar kelas yang tersedia, mengajukan kelas miliknya untuk ditukarkan, menyusun dan mengubah daftar peringkat preferensi, serta melihat hasil akhir dari proses pertukaran. Di sisi lain, pengguna dengan hak akses admin memiliki kewenangan untuk memicu eksekusi algoritma pertukaran dan meninjau statistik hasil pertukaran secara keseluruhan. Berdasarkan pembagian

fungsionalitas tersebut, maka dibutuhkan sebuah sistem terpusat berbasis web yang dapat diakses oleh kedua jenis pengguna melalui peramban.

Salah satu kendala dalam sistem pertukaran adalah validitas data. Oleh karena itu, dibutuhkan sebuah mekanisme verifikasi untuk memastikan setiap mahasiswa yang mengajukan pertukaran benar-benar terdaftar sebagai pemilik sah dari kelas yang ditawarkan. Proses pembuatan sistem ini diputuskan untuk berpusat pada algoritma *Top Trading Cycles* (TTC) sebagai mesin utama dalam pemrosesan pertukaran. Eksekusi algoritma akan dilakukan sepenuhnya pada sisi server (*back-end*) untuk menjaga integritas data dan proses. *Application Programming Interface* (API) pada sistem ini akan dibangun menggunakan bahasa pemrograman Python dengan framework Flask, serta memanfaatkan basis data MySQL untuk menyimpan data pengguna, data kelas, permintaan tukar, dan preferensi. Sistem yang dibuat untuk antarmuka pengguna (*front-end*) akan menggunakan teknologi web standar seperti *HyperText Markup Language* (HTML) dan Bootstrap.

3.3 Perancangan Sistem

Berdasarkan analisis kebutuhan yang telah dilakukan pada sub-bab sebelumnya, pada tahap ini dirancang sebuah sistem pertukaran jadwal perkuliahan berbasis web. Perancangan ini bertujuan untuk menerjemahkan kebutuhan fungsional menjadi sebuah arsitektur dan alur kerja sistem yang konkret. Fungsionalitas utama yang dirancang untuk sistem ini dijabarkan sebagai berikut:

A. Proses Autentikasi Pengguna

Fungsionalitas ini menangani proses masuk (*login*) bagi pengguna (mahasiswa dan admin) untuk memastikan hanya pengguna terotorisasi yang dapat mengakses sistem.

B. Proses Pengelolaan Permintaan Tukar

Fungsionalitas ini memungkinkan mahasiswa untuk mengajukan kelas yang ingin mereka tukarkan serta menyusun daftar peringkat kelas-kelas lain yang mereka inginkan sebagai preferensi.

C. Proses Eksekusi Algoritma Pertukaran

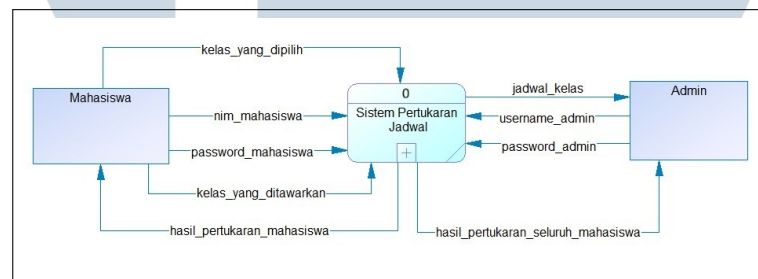
Proses ini hanya dapat diakses oleh admin, di mana sistem akan menjalankan algoritma TTC untuk memproses semua permintaan yang aktif dan menentukan hasilnya.

D. Proses Peninjauan Hasil Pertukaran

Setelah algoritma dieksekusi, mahasiswa dapat melihat hasil pertukaran pada halaman dasbor mereka, apakah mereka berhasil mendapatkan jadwal baru atau tidak.

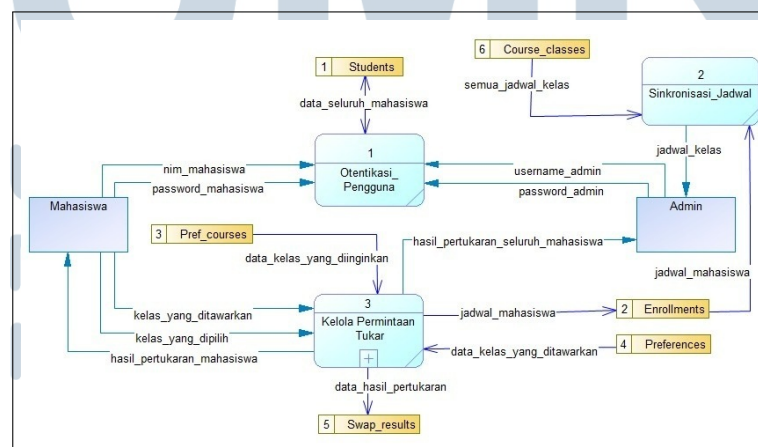
3.3.1 Data Flow Diagram

Rancangan sistem yang dibangun pada penelitian ini akan dijabarkan lebih lanjut dalam bentuk *Data Flow Diagram* (DFD) untuk menggambarkan alur data. Pada Gambar 3.1 merupakan DFD Konteks (Diagram Konteks) untuk Sistem Pertukaran Jadwal yang menggambarkan aliran data secara keseluruhan antara sistem dengan entitas eksternal. Terdapat dua entitas eksternal, yaitu Mahasiswa dan Admin.



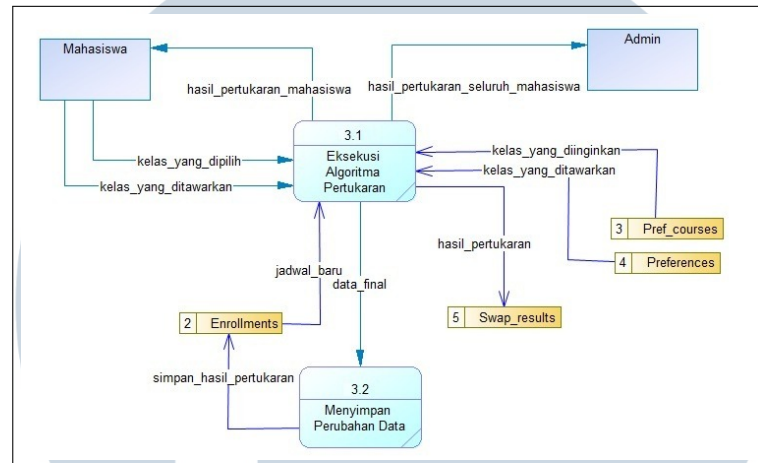
Gambar 3.1. DFD Context Diagram Sistem Pertukaran Jadwal

Diagram Konteks kemudian diuraikan lebih lanjut menjadi DFD Level 1 yang menjabarkan proses-proses utama di dalam sistem. Proses-proses tersebut dapat dilihat pada penjabaran pada Gambar 3.2



Gambar 3.2. DFD Level 1 Sistem Pertukaran Jadwal

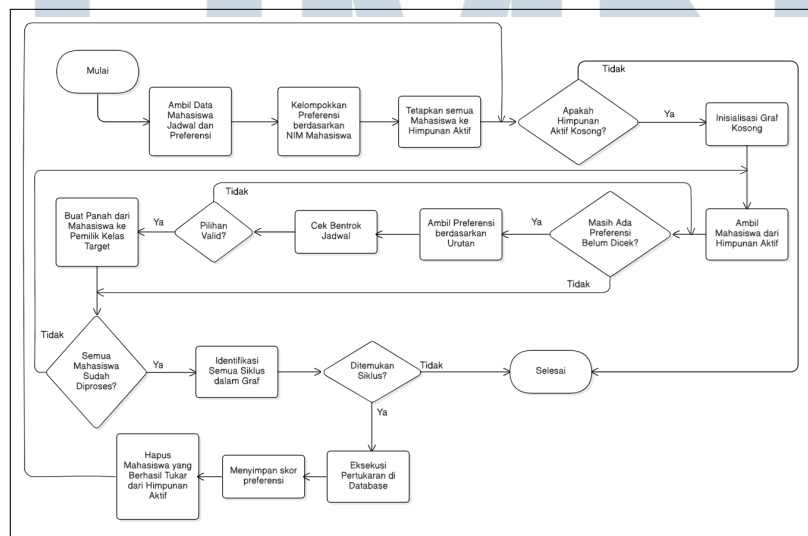
Selanjutnya pada Proses Kelola Permintaan Tukar dijelaskan lebih detail menjadi Proses Eksekusi Algoritma Pertukaran dan Proses Menyimpan Perubahan Data pada Gambar 3.3.



Gambar 3.3. DFD Level 2 Sistem Pertukaran Jadwal

3.3.2 Flowchart Sistem

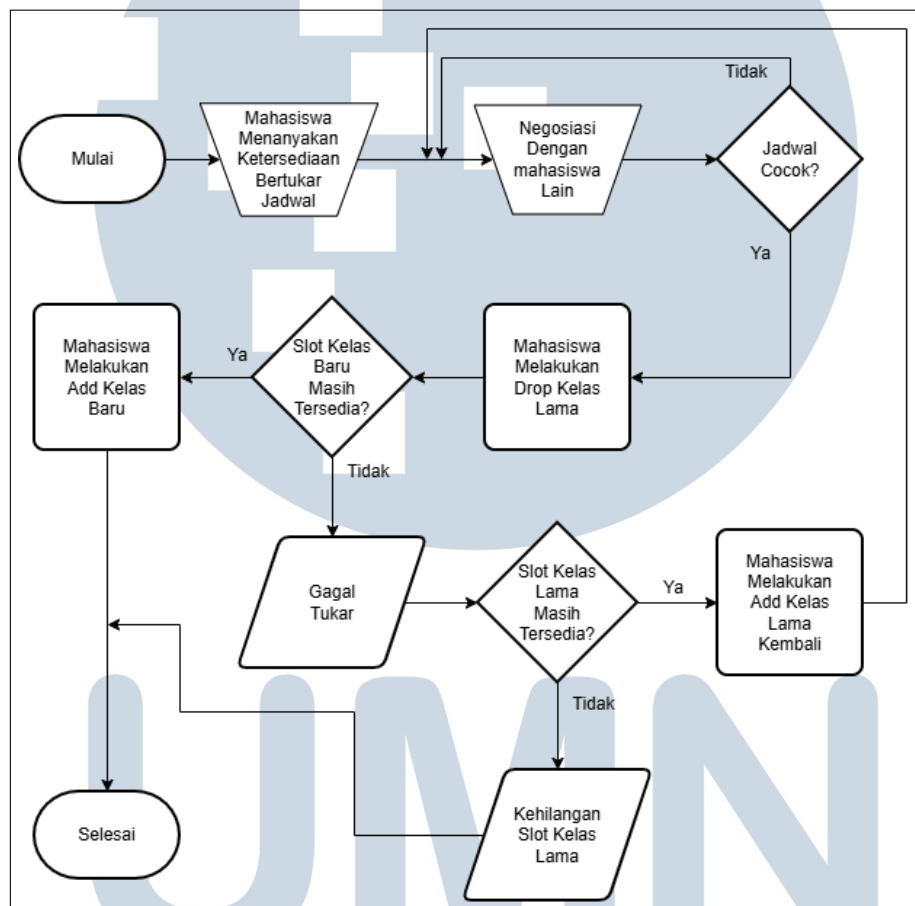
Rancangan sistem selanjutnya yang dibangun pada penelitian ini adalah *Flowchart* untuk menjelaskan alur kerja sistem dan logika proses. Alur ini dimulai dari alur kerja sistem menggunakan implementasi dari TTC dalam proses pertukaran jadwal mahasiswa pada Gambar 3.4.



Gambar 3.4. Flowchart Alur TTC

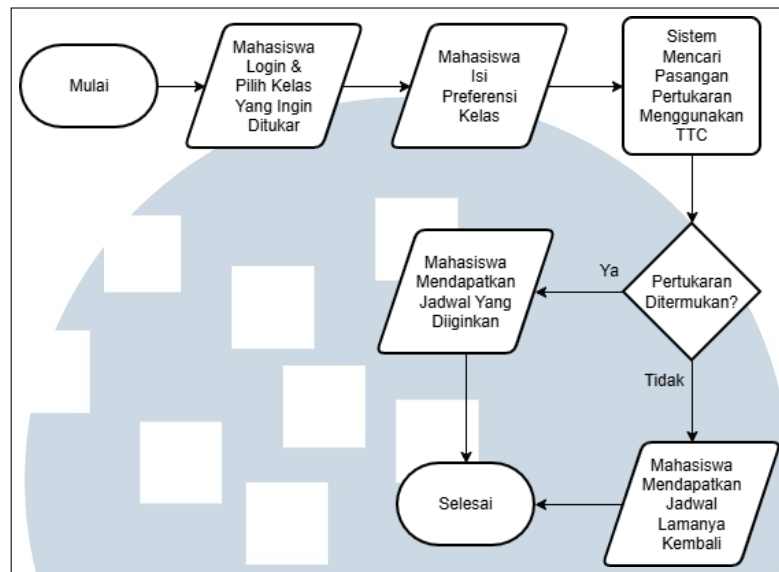
Alur proses pertukaran jadwal secara manual pada Gambar 3.5

merepresentasikan ketergantungan mahasiswa pada koordinasi tidak terstruktur melalui media sosial. Proses ini memiliki hambatan utama berupa kesulitan dalam mengidentifikasi siklus pertukaran yang melibatkan lebih dari dua pihak, sehingga sering kali berakhir tanpa hasil yang optimal bagi seluruh partisipan. Selain itu, terdapat risiko kehilangan slot jadwal awal saat mahasiswa melakukan proses *drop* kelas secara manual sebelum berhasil mengambil kelas yang baru.



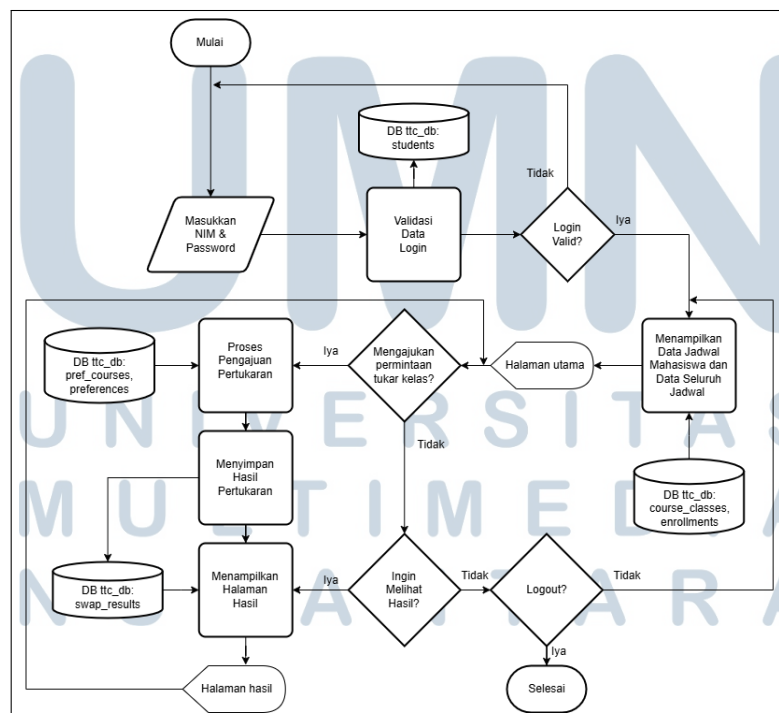
Gambar 3.5. Flowchart Sebelum Sistem TTC

Alur proses yang diusulkan pada Gambar 3.6 mengintegrasikan algoritma *Top Trading Cycles* untuk mengotomatisasi pencarian siklus pertukaran berdasarkan urutan preferensi yang telah dimasukkan oleh mahasiswa. Dengan mekanisme ini, sistem dapat menjamin tercapainya kondisi *Pareto Optimal* di mana setiap mahasiswa mendapatkan jadwal yang lebih baik atau setidaknya sama tanpa risiko teknis seperti kehilangan slot jadwal awal. Seluruh proses validasi bentrok jadwal dan pembaruan data dilakukan secara atomik oleh sistem untuk memastikan konsistensi data pada basis data universitas.



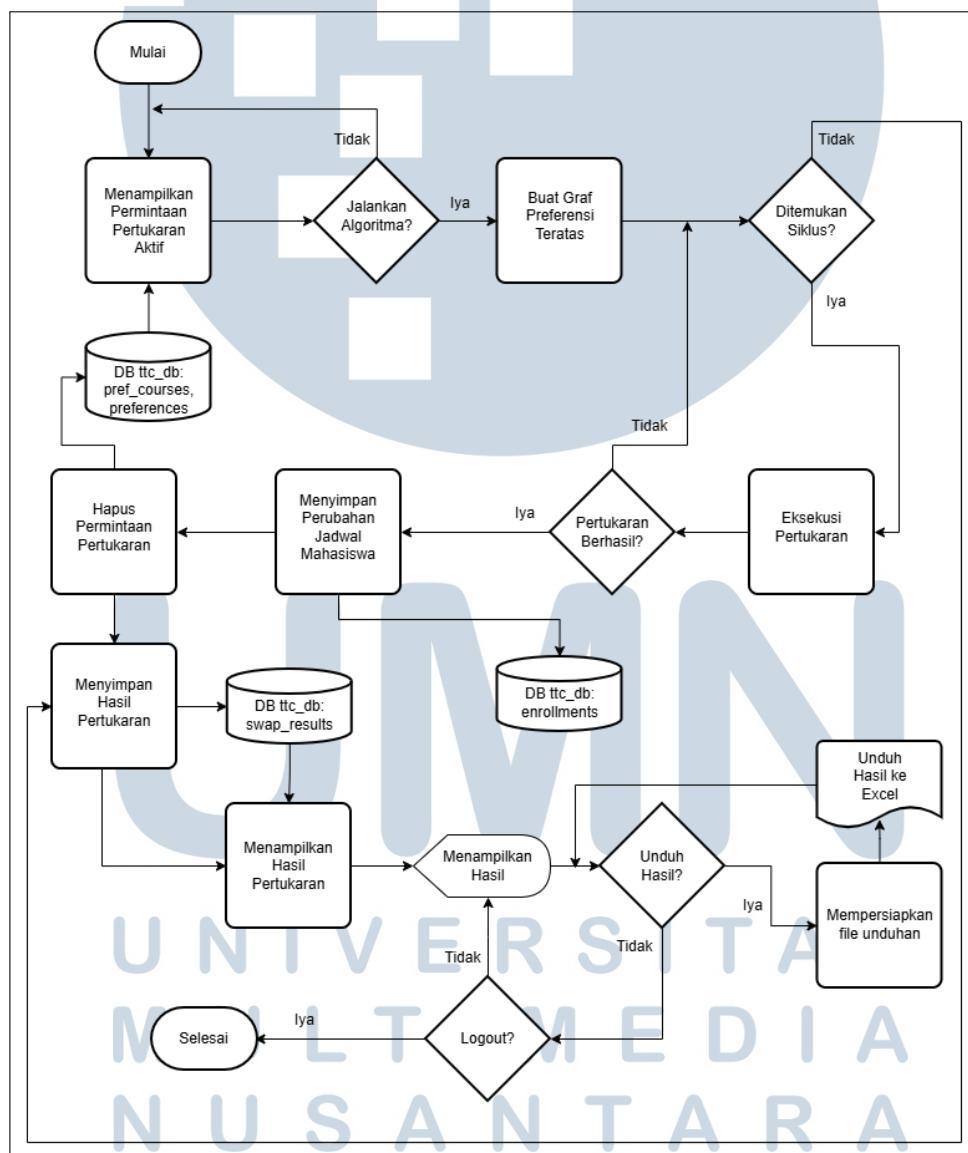
Gambar 3.6. Flowchart Sesudah Sistem TTC

Selanjutnya adalah *flowchart* alur kerja mahasiswa. Pada alur kerja mahasiswa dimulai dari proses autentikasi hingga mahasiswa keluar dari sistem. Flowchart pada Gambar 3.7 mengilustrasikan urutan langkah-langkah yang dapat dilakukan oleh mahasiswa, seperti mengajukan permintaan pertukaran, melihat status, hingga melihat hasil akhir dari proses pertukaran jadwal.



Gambar 3.7. Flowchart Alur Kerja Mahasiswa

Selanjutnya adalah alur kerja untuk pengguna dengan peran sebagai admin. Berbeda dengan mahasiswa, alur kerja admin jauh lebih sederhana dan berfokus pada fungsi pengelolaan sistem. Fungsi utamanya adalah untuk memicu eksekusi algoritma pertukaran jadwal. Flowchart pada Gambar 3.8 dimulai dengan autentikasi admin. Setelah berhasil masuk, admin memiliki opsi untuk langsung memicu proses eksekusi algoritma TTC. Setelah pemicu diaktifkan, sistem akan mengambil alih dan menjalankan proses pertukaran di latar belakang. Setelah proses selesai, alur kerja admin pada sesi tersebut dianggap tuntas.

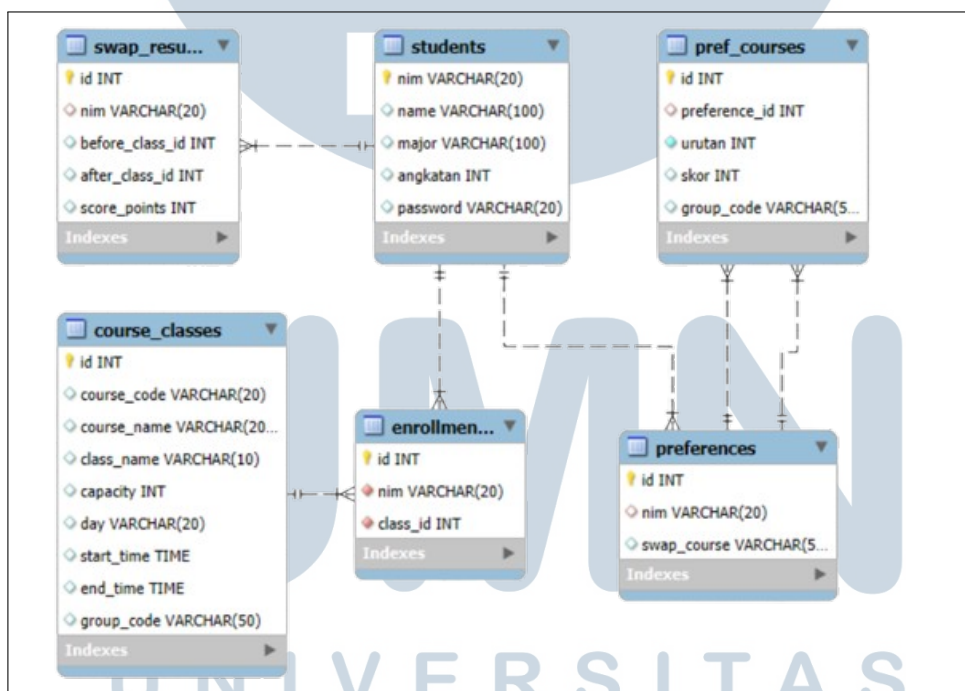


Gambar 3.8. Flowchart Alur Kerja Admin

3.3.3 Entity Relationship Diagram

Untuk memodelkan struktur basis data, digunakan *Entity Relationship Diagram* (ERD). ERD merupakan sebuah model konseptual yang merepresentasikan struktur logis dari basis data dengan menggambarkan entitas-entitas data utama beserta relasi atau hubungan antar entitas tersebut. Perancangan ERD ini menjadi landasan untuk pembuatan tabel-tabel fisik dalam basis data.

Rancangan ERD untuk Sistem Pertukaran Jadwal Perkuliahan ini berpusat pada entitas *students* sebagai pengguna utama dan *course_classes* sebagai objek yang dipertukarkan. Hubungan antara keduanya dicatat melalui entitas *enrollment*. Ketika seorang mahasiswa mengajukan permintaan, data akan disimpan pada entitas *preferences* dan detail pilihan kelasnya dicatat pada *course_classes*. Hasil akhir dari eksekusi algoritma akan disimpan dalam entitas *swap_results*. Keseluruhan model relasional ini dapat dilihat pada Gambar 3.9.



Gambar 3.9. ERD Sistem Pertukaran Jadwal

Data jadwal dalam format CSV yang diperoleh dari Biro Administrasi Akademik (BIA) ditampilkan pada Gambar 3.10 sebagai titik awal penelitian sebanyak 10.650 baris. Data ini berisi informasi krusial seperti NIM mahasiswa, kode mata kuliah, nama kelas, serta waktu pelaksanaan yang menjadi dasar penentuan kepemilikan awal bagi setiap partisipan sebelum proses pertukaran dimulai.

	A	B	C	D	E	F	G	H	I
1	id	PRODI	ANGKATAN	KODE_MK	NAMA_MK	KELAS	HARI	JAM_MULAI	JAM_SELESAI
2	88567154526	Sistem Informasi	2022	MSC3507	Sales Promotion Communication	A-EN	SENIN	11:00	14:00
3	88567154526	Sistem Informasi	2022	EM 716	Sustainable Marketing	A	SENIN	14:00	17:00
4	88567154526	Sistem Informasi	2022	JR 662	Digital Media Business	A	SELASA	11:00	14:00
5	88567154526	Sistem Informasi	2022	JR 114	Entertainment Content Producti	A-EN	SELASA	14:00	17:00
6	88567154526	Sistem Informasi	2022	EM 608	Digital Marketing	B	RABU	08:00	11:00
7	88567154526	Sistem Informasi	2022	EM 717	Sustainable Human Capital	D	RABU	14:00	17:00
8	88567154526	Sistem Informasi	2022	JR 349	Media and Politics	A-EN	KAMIS	08:00	11:00
9	07663854896	Sistem Informasi	2022	MSC3507	Sales Promotion Communication	A-EN	SENIN	11:00	14:00
10	07663854896	Sistem Informasi	2022	EM 716	Sustainable Marketing	A	SENIN	14:00	17:00
11	07663854896	Sistem Informasi	2022	JR 662	Digital Media Business	A	SELASA	11:00	14:00
12	07663854896	Sistem Informasi	2022	JR 114	Entertainment Content Producti	A-EN	SELASA	14:00	17:00
13	07663854896	Sistem Informasi	2022	EM 608	Digital Marketing	B	RABU	08:00	11:00
14	07663854896	Sistem Informasi	2022	EM 717	Sustainable Human Capital	D	RABU	14:00	17:00
15	07663854896	Sistem Informasi	2022	JR 349	Media and Politics	A-EN	KAMIS	08:00	11:00
16	46850239362	Informatika	2022	EM 526	Social Entrepreneurship	D	SENIN	11:00	14:00
17	46850239362	Informatika	2022	EM 715	Sustainable Finance	D	SENIN	14:00	17:00
18	46850239362	Informatika	2022	IS 411	Data Modelling	A	SELASA	08:00	09:00

Gambar 3.10. Tampilan *Dataset* Awal Jadwal Kuliah dari BIA

Setelah data mentah melalui tahap pra-pemrosesan, informasi tersebut ditransformasi menjadi matriks pasangan potensial (*potential pairs*) sebagaimana ditunjukkan pada Gambar 3.11. Visualisasi tersebut memaparkan pemetaan sistematis antara mahasiswa yang memiliki kelas tertentu dengan kandidat kelas target yang secara teknis memungkinkan untuk dipertukarkan berdasarkan kesamaan kode mata kuliah sebanyak 747.989 baris.

	A	B	C	D	E	F
1	id,GROUP_CODE,COURSE_ID,id.1,GROUP_CODE.1,COURSE_ID.1					
2	46850239362,EM 526-D,8,12759222660,EM 105-H,28					
3	46850239362,EM 526-D,8,43436728501,EM 105-F,116					
4	46850239362,EM 526-D,8,30991203120,MSC3507-A,1					
5	46850239362,EM 526-D,8,34467421830,EM 105-F,116					
6	46850239362,EM 526-D,8,98137266198,EM 105-F,116					
7	46850239362,EM 526-D,8,27866715053,EM 105-F,116					
8	46850239362,EM 526-D,8,84531395919,MSC3507-A,1					
9	46850239362,EM 526-D,8,73822290273,MSC7233-E,164					
10	46850239362,EM 526-D,8,33126828450,EM 105-F,116					
11	46850239362,EM 526-D,8,56081239744,MSC3201-A,328					
12	46850239362,EM 526-D,8,75785702630,MSC7233-E,164					
13	6449003244,EM 526-D,8,12759222660,EM 105-H,28					
14	6449003244,EM 526-D,8,43436728501,EM 105-F,116					
15	6449003244,EM 526-D,8,30991203120,MSC3507-A,1					
16	6449003244,EM 526-D,8,34467421830,EM 105-F,116					

Gambar 3.11. Matriks Pemetaan Pasangan Potensial (*Potential Pairs*)

Dalam mendukung jalannya simulasi algoritma TTC, struktur basis data relasional disiapkan dengan mendefinisikan tabel mahasiswa seperti terlihat pada Gambar 3.12. Tabel *students* ini berfungsi sebagai repositori utama untuk menyimpan identitas unik partisipan, di mana NIM digunakan sebagai kunci utama (*primary key*) untuk menjamin integritas data akademik.

```

13 -- Mengisi tabel students
14 • INSERT INTO students (nim, major, angkatan, password) VALUES ('46850239362', 'Informatika', 2022, '1234');
15 • INSERT INTO students (nim, major, angkatan, password) VALUES ('47573927613', 'Informatika', 2022, '1234');
16 • INSERT INTO students (nim, major, angkatan, password) VALUES ('42017585167', 'Informatika', 2022, '1234');
17 • INSERT INTO students (nim, major, angkatan, password) VALUES ('46080239870', 'Informatika', 2022, '1234');
18 • INSERT INTO students (nim, major, angkatan, password) VALUES ('6449003244', 'Informatika', 2022, '1234');
19 • INSERT INTO students (nim, major, angkatan, password) VALUES ('6523661321', 'Informatika', 2022, '1234');
20 • INSERT INTO students (nim, major, angkatan, password) VALUES ('9879548682', 'Informatika', 2022, '1234');
21 • INSERT INTO students (nim, major, angkatan, password) VALUES ('12759222660', 'Informatika', 2022, '1234');
22 • INSERT INTO students (nim, major, angkatan, password) VALUES ('35531007523', 'Informatika', 2022, '1234');
23 • INSERT INTO students (nim, major, angkatan, password) VALUES ('46850310311', 'Informatika', 2022, '1234');
24 • INSERT INTO students (nim, major, angkatan, password) VALUES ('50179661595', 'Informatika', 2022, '1234');
25 • INSERT INTO students (nim, major, angkatan, password) VALUES ('71853765271', 'Informatika', 2022, '1234');

```

Gambar 3.12. Struktur Tabel students dalam Basis Data

Struktur tabel berikutnya adalah `course_classes` yang dapat dilihat pada Gambar 3.13. Gambar tersebut mendefinisikan katalog seluruh kelas yang tersedia dalam sistem, mencakup atribut spesifik seperti kode mata kuliah, grup jadwal, dan waktu perkuliahan yang akan menjadi objek utama dalam mekanisme pertukaran.

```

639 -- Mengisi tabel course_classes
640 • INSERT INTO course_classes (id, course_code, course_name, class_name, day, start_time, end_time, group_code)
641 VALUES (8, 'EH 526', 'Social Entrepreneurship', 'D', 'SENIN', '11:00', '14:00', 'EH 526-D');
642 • INSERT INTO course_classes (id, course_code, course_name, class_name, day, start_time, end_time, group_code)
643 VALUES (9, 'EH 715', 'Sustainable Finance', 'D', 'SENIN', '14:00', '17:00', 'EH 715-D');
644 • INSERT INTO course_classes (id, course_code, course_name, class_name, day, start_time, end_time, group_code)
645 VALUES (10, 'IS 411', 'Data Modelling', 'A', 'SELASA', '08:00', '09:00', 'IS 411-A');
646 • INSERT INTO course_classes (id, course_code, course_name, class_name, day, start_time, end_time, group_code)
647 VALUES (11, 'IS 411', 'Data Modelling', 'AL', 'SELASA', '09:00', '11:00', 'IS 411-A');
648 • INSERT INTO course_classes (id, course_code, course_name, class_name, day, start_time, end_time, group_code)
649 VALUES (12, 'CE 542', 'System Administration', 'A-EN', 'SELASA', '13:00', '15:00', 'CE 542-A');
650 • INSERT INTO course_classes (id, course_code, course_name, class_name, day, start_time, end_time, group_code)
651 VALUES (5, 'EH 600', 'Digital Marketing', 'B', 'RABU', '08:00', '11:00', 'EH 600-B');
652 • INSERT INTO course_classes (id, course_code, course_name, class_name, day, start_time, end_time, group_code)
653 VALUES (13, 'IS 412', 'Data Visualization', 'A', 'RABU', '13:00', '14:00', 'IS 412-A');

```

Gambar 3.13. Struktur Tabel `course_classes` sebagai Katalog Mata Kuliah

Tabel `enrollments` yang ditunjukkan pada Gambar 3.14 merupakan tabel relasi yang sangat penting dalam merepresentasikan status kepemilikan kelas. Melalui struktur ini, sistem menghubungkan setiap NIM mahasiswa dengan ID kelas tertentu untuk mengunci posisi jadwal awal mahasiswa sebelum algoritma TTC melakukan pencarian siklus.

```

891 -- Mengisi tabel enrollments
892 • INSERT INTO enrollments (nim, class_id) VALUES ('46850239362', 8);
893 • INSERT INTO enrollments (nim, class_id) VALUES ('46850239362', 9);
894 • INSERT INTO enrollments (nim, class_id) VALUES ('46850239362', 10);
895 • INSERT INTO enrollments (nim, class_id) VALUES ('46850239362', 11);
896 • INSERT INTO enrollments (nim, class_id) VALUES ('46850239362', 12);
897 • INSERT INTO enrollments (nim, class_id) VALUES ('46850239362', 5);
898 • INSERT INTO enrollments (nim, class_id) VALUES ('46850239362', 13);
899 • INSERT INTO enrollments (nim, class_id) VALUES ('46850239362', 14);
900 • INSERT INTO enrollments (nim, class_id) VALUES ('46850239362', 15);
901 • INSERT INTO enrollments (nim, class_id) VALUES ('46850239362', 16);
902 • INSERT INTO enrollments (nim, class_id) VALUES ('47573927613', 17);
903 • INSERT INTO enrollments (nim, class_id) VALUES ('42017585167', 18);
904 • INSERT INTO enrollments (nim, class_id) VALUES ('42017585167', 19);
905 • INSERT INTO enrollments (nim, class_id) VALUES ('46080239870', 20);
906 • INSERT INTO enrollments (nim, class_id) VALUES ('6449003244', 8);

```

Gambar 3.14. Struktur Tabel `enrollments` untuk Relasi Kepemilikan Kelas

Proses injeksi data untuk preferensi pertama atau prioritas utama mahasiswa ditampilkan melalui potongan kueri SQL pada Gambar 3.15. Pada tahap ini, sistem secara simultan memasukkan data ke dalam tabel `preferences` untuk menentukan kelas yang ditawarkan dan tabel `pref_courses` untuk menetapkan target kelas yang paling diinginkan oleh mahasiswa.

```

>
4 • INSERT INTO preferences (nim, swap_course) SELECT '60623368362', e.id FROM enrollments e
5 JOIN course_classes cc ON e.class_id = cc.id WHERE e.nim = '60623368362' AND cc.group_code = 'MSC3401-F';
6 • INSERT INTO pref_courses (preference_id, urutan, skor, group_code) VALUES (LAST_INSERT_ID(), 1, 100, 'EM 105-E');
7
8 • INSERT INTO preferences (nim, swap_course) SELECT '55642788845', e.id FROM enrollments e
9 JOIN course_classes cc ON e.class_id = cc.id WHERE e.nim = '55642788845' AND cc.group_code = 'EM 105-E';
10 • INSERT INTO pref_courses (preference_id, urutan, skor, group_code) VALUES (LAST_INSERT_ID(), 1, 100, 'MSC3401-F');
11
12 • INSERT INTO preferences (nim, swap_course) SELECT '70890887117', e.id FROM enrollments e
13 JOIN course_classes cc ON e.class_id = cc.id WHERE e.nim = '70890887117' AND cc.group_code = 'IF 440-E';
14 • INSERT INTO pref_courses (preference_id, urutan, skor, group_code) VALUES (LAST_INSERT_ID(), 1, 100, 'IF 333-A');
15
16 • INSERT INTO preferences (nim, swap_course) SELECT '51668553132', e.id FROM enrollments e
17 JOIN course_classes cc ON e.class_id = cc.id WHERE e.nim = '51668553132' AND cc.group_code = 'IF 333-A';
18 • INSERT INTO pref_courses (preference_id, urutan, skor, group_code) VALUES (LAST_INSERT_ID(), 1, 100, 'IF 440-E');

```

Gambar 3.15. Kueri SQL untuk Injeksi Data Preferensi Pertama

Gambar 3.16 menunjukkan kueri SQL untuk pendaftaran preferensi kedua yang berfungsi sebagai pilihan alternatif bagi mahasiswa. Sama seperti proses sebelumnya, kueri ini mengisi tabel `preferences` dan `pref_courses` untuk mendokumentasikan opsi cadangan, sehingga algoritma memiliki fleksibilitas lebih tinggi dalam menemukan siklus jika preferensi utama tidak dapat terpenuhi.

```

4 • INSERT INTO preferences (nim, swap_course) SELECT '87786524252', e.id FROM enrollments e JOIN course_classes cc ON e.class_id = cc.id
5 WHERE e.nim = '87786524252' AND cc.group_code = 'IF 581-I';
6 • INSERT INTO pref_courses (preference_id, urutan, skor, group_code) VALUES (LAST_INSERT_ID(), 2, 50, 'IF 571-B');
7
8 • INSERT INTO preferences (nim, swap_course) SELECT '58813080923', e.id FROM enrollments e JOIN course_classes cc ON e.class_id = cc.id
9 WHERE e.nim = '58813080923' AND cc.group_code = 'IF 441-D';
10 • INSERT INTO pref_courses (preference_id, urutan, skor, group_code) VALUES (LAST_INSERT_ID(), 2, 50, 'IF 570-D');
11
12 • INSERT INTO preferences (nim, swap_course) SELECT '61440346668', e.id FROM enrollments e JOIN course_classes cc ON e.class_id = cc.id
13 WHERE e.nim = '61440346668' AND cc.group_code = 'CE 319-F';
14 • INSERT INTO pref_courses (preference_id, urutan, skor, group_code) VALUES (LAST_INSERT_ID(), 2, 50, 'EM 105-B');
15
16 • INSERT INTO preferences (nim, swap_course) SELECT '63878525586', e.id FROM enrollments e JOIN course_classes cc ON e.class_id = cc.id
17 WHERE e.nim = '63878525586' AND cc.group_code = 'IF 441-E';
18 • INSERT INTO pref_courses (preference_id, urutan, skor, group_code) VALUES (LAST_INSERT_ID(), 2, 50, 'IF 333-C');

```

Gambar 3.16. Kueri SQL untuk Injeksi Data Preferensi Kedua

Setelah algoritma TTC selesai mengeksekusi seluruh siklus pertukaran yang ditemukan, status akhir alokasi jadwal disimpan dalam tabel `swap_results`. Kueri SQL yang digunakan untuk mengekstraksi *dataset* hasil pemrosesan tersebut ditunjukkan pada kode di bawah ini. Kueri ini melakukan operasi *join* antara tabel hasil pertukaran dengan tabel katalog kelas (`course_classes`) sebanyak dua kali untuk membandingkan atribut kelas sebelum dan sesudah proses pemrosesan secara *head-to-head*.

```

1 • SELECT
2     sr.nim AS 'NIM',
3     cc_old.course_name AS 'Mata Kuliah',
4     cc_old.group_code AS 'Kelas Lama (Before)',
5     cc_new.group_code AS 'Kelas Baru (After)',
6     sr.score_points AS 'Skor Kepuasan'
7 FROM swap_results sr
8 JOIN course_classes cc_old ON sr.before_class_id = cc_old.id
9 JOIN course_classes cc_new ON sr.after_class_id = cc_new.id
10 ORDER BY sr.nim ASC;

```

NIM	Mata Kuliah	Kelas Lama (Before)	Kelas Baru (After)	Skor Kepuasan
10714770811	Game Development	IF 581-I	IF 571-F	50
11178556949	Probability and Statistics	CE 319-B	IF 351-C	50
11619784803	Information Technology Research	IF 590-G	IF 581-F	50
11721808342	English 3	UM 321-B	IF 441-F	50
11743977702	Technopreneurship	EM 105-L	EM 608-A	100
11822412054	English 3	UM 321-C	IF 590-G	50
12376775691	Advanced Web Programming	IF 451-A	IF 351-F	50
12389573937	Game Development	IF 581-E	IF 351-A	50

Hasil dari kueri tersebut memberikan gambaran komprehensif mengenai efektivitas algoritma, di mana setiap baris data merepresentasikan satu transaksi pertukaran yang berhasil. *Dataset* ini mencakup kolom *score_points* yang menunjukkan tingkat kepuasan yang dicapai mahasiswa berdasarkan bobot preferensi yang mereka ajukan. Melalui tampilan data setelah pemrosesan ini, dapat dibuktikan bahwa sistem telah berhasil melakukan pembaruan alokasi secara akurat sesuai dengan siklus yang terdeteksi oleh algoritma *Top Trading Cycles*.

3.3.4 Struktur Tabel

Bagian ini menjelaskan struktur basis data yang dirancang untuk mendukung operasional sistem pertukaran jadwal perkuliahan. Basis data ini akan menyimpan seluruh informasi terkait pengguna, kelas, preferensi, dan hasil pertukaran.

1. Tabel *students*

Tabel *students* berfungsi untuk menyimpan data profil dari setiap mahasiswa yang terdaftar di sistem. Data ini meliputi informasi dasar mahasiswa serta kredensial untuk autentikasi seperti NIM, nama, jurusan, angkatan, dan *password*.

Tabel 3.1. Struktur Tabel `students`

Nama Kolom	Tipe Data	Nullable	Identitas
nim	varchar(20)		PK
name	varchar(100)	ya	
major	varchar(100)	ya	
angkatan	int	ya	
password	varchar(255)		

2. Tabel `enrollment`

Tabel `enrollment` adalah tabel penghubung (*junction table*) yang mencatat kelas-kelas apa saja yang sedang diambil oleh setiap mahasiswa.

Tabel 3.2. Struktur Tabel `enrollment`

Nama Kolom	Tipe Data	Nullable	Identitas
id	int		PK
nim	varchar(20)		FK
class_id	int		FK

3. Tabel `course_classes`

Tabel `course_classes` pada gunanya untuk menyimpan data detail dari setiap kelas perkuliahan yang tersedia di sistem, termasuk informasi kode mata kuliah, nama, jadwal, dan ruang.

Tabel 3.3. Struktur Tabel `course_classes`

Nama Kolom	Tipe Data	Nullable	Identitas
id	int		PK
course_code	varchar(20)	ya	
course_name	varchar(200)	ya	
class_name	varchar(10)	ya	
capacity	int	ya	
day	varchar(20)	ya	
start_time	time	ya	
end_time	time	ya	
group_code	varchar(50)	ya	

4. Tabel `pref_courses`

Fungsi dari tabel `pref_courses` adalah untuk mencatat setiap pilihan kelas yang diinginkan oleh seorang mahasiswa untuk satu permintaan pertukaran spesifik, beserta urutannya.

Tabel 3.4. Struktur Tabel `pref_courses`

Nama Kolom	Tipe Data	Nullable	Identitas
<code>id</code>	<code>int</code>		PK
<code>preference_id</code>	<code>int</code>	ya	FK
<code>urutan</code>	<code>int</code>	ya	
<code>skor</code>	<code>int</code>	ya	
<code>group_code</code>	<code>varchar(50)</code>	ya	FK

5. Tabel `preferences`

Tabel `preferences` akan menyimpan data utama dari setiap permintaan pertukaran yang diajukan oleh mahasiswa, menautkan seorang mahasiswa dengan kelas yang ingin mereka lepaskan.

Tabel 3.5. Struktur Tabel `preferences`

Nama Kolom	Tipe Data	Nullable	Identitas
<code>id</code>	<code>int</code>		PK
<code>nim</code>	<code>varchar(20)</code>	ya	FK
<code>swap_course</code>	<code>varchar(50)</code>	ya	FK

6. Tabel `swap_results`

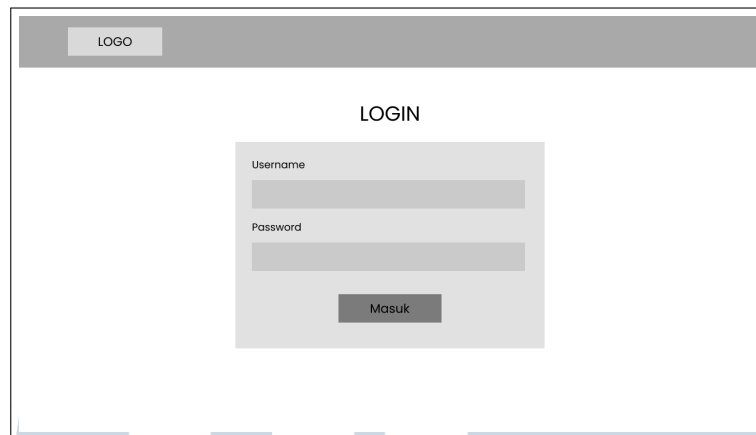
Tabel `swap_results` berfungsi untuk menyimpan catatan historis dari setiap pertukaran yang berhasil dieksekusi oleh sistem.

Tabel 3.6. Struktur Tabel `swap_results`

Nama Kolom	Tipe Data	Nullable	Identitas
<code>id</code>	<code>int</code>		PK
<code>nim</code>	<code>varchar(20)</code>	ya	FK
<code>before_class_id</code>	<code>int</code>	ya	FK
<code>after_class_id</code>	<code>int</code>	ya	FK
<code>score_points</code>	<code>int</code>	ya	

3.3.5 Perancangan Antarmuka Sistem

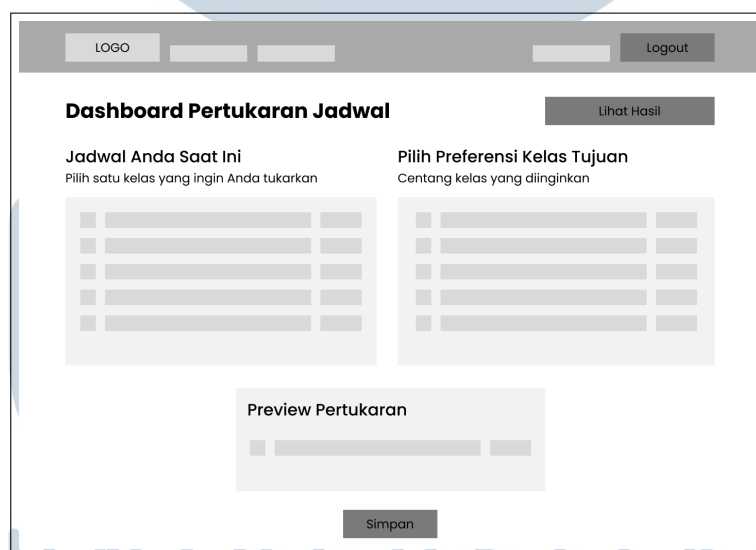
Perancangan halaman sistem berfokus pada pengalaman pengguna untuk memastikan setiap alur kerja dapat dilakukan dengan langkah seminimal mungkin yang dibuat dalam bentuk *mockup*. Gambar 3.17 merupakan halaman *login* untuk membedakan tampilan untuk mahasiswa atau admin sesuai dengan *username*.



Gambar 3.17. Halaman Login

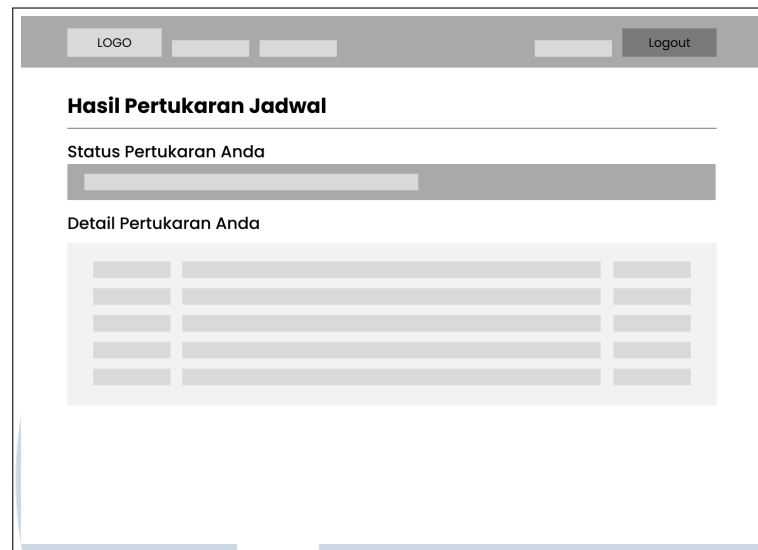
A. Halaman Mahasiswa

Halaman ini menampilkan jadwal yang dimiliki, formulir untuk mendaftarkan kelas yang ingin ditukar, serta sebuah daftar kelas yang sudah dipilih sebagai *preview* sebelum menyimpan pada Gambar 3.18



Gambar 3.18. Halaman Utama Mahasiswa

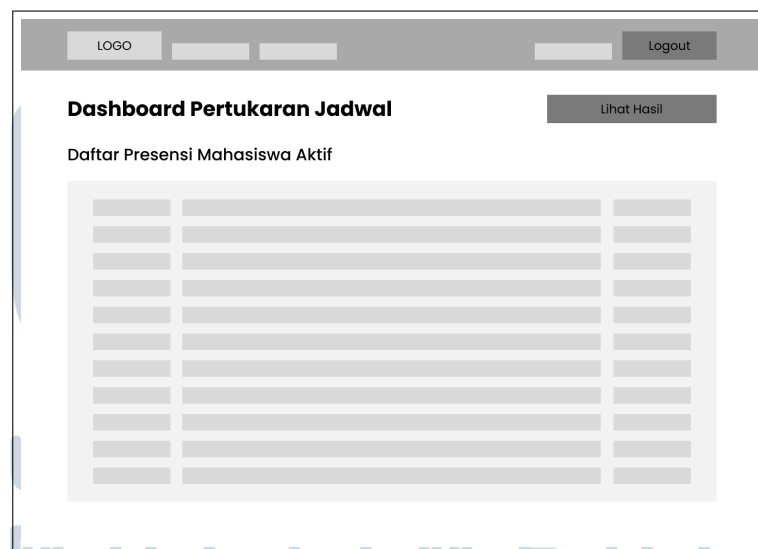
Tombol aksi yang jelas disediakan untuk menyimpan permintaan dan melihat hasil seperti pada Gambar 3.19. Tata letak ini dirancang untuk memastikan alur pengajuan permintaan dapat diselesaikan dalam satu halaman untuk kemudahan penggunaan.



Gambar 3.19. Halaman Hasil Mahasiswa

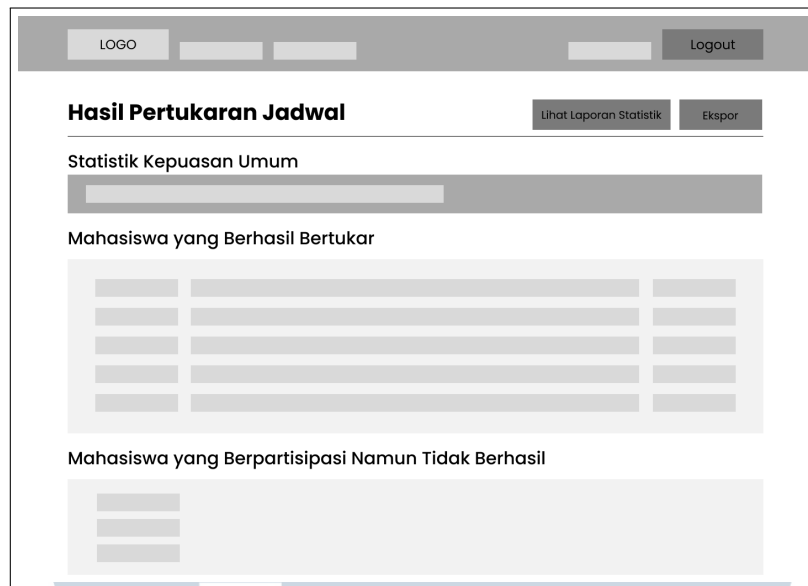
B. Halaman Admin

Antarmuka untuk admin dirancang dengan orientasi pada fungsionalitas dan kontrol untuk memberikan admin alat yang jelas dan efektif untuk mengelola siklus pertukaran pada Gambar 3.20.



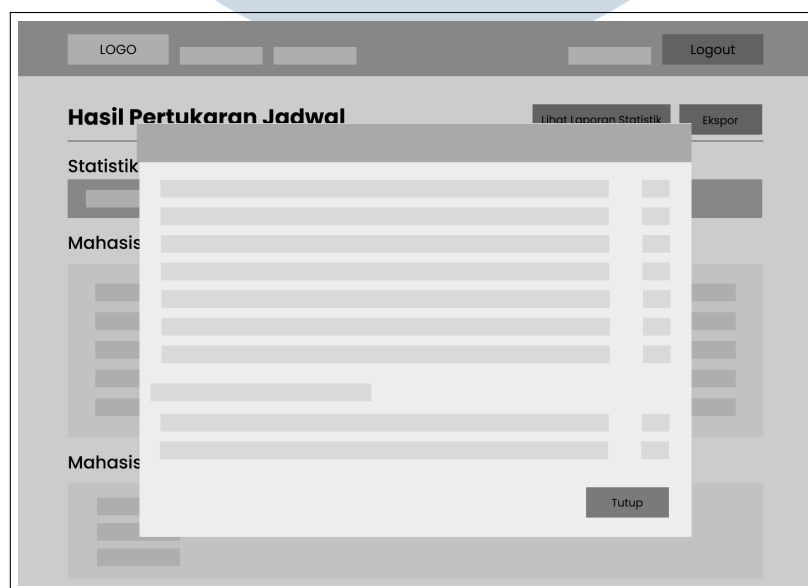
Gambar 3.20. Halaman Utama Admin

Gambar 3.21 yang merupakan halaman yang menampilkan statistik ringkas dari hasil eksekusi terakhir beserta detail mahasiswa yang berhasil menukar ataupun yang tidak berhasil.



Gambar 3.21. Halaman Hasil Admin

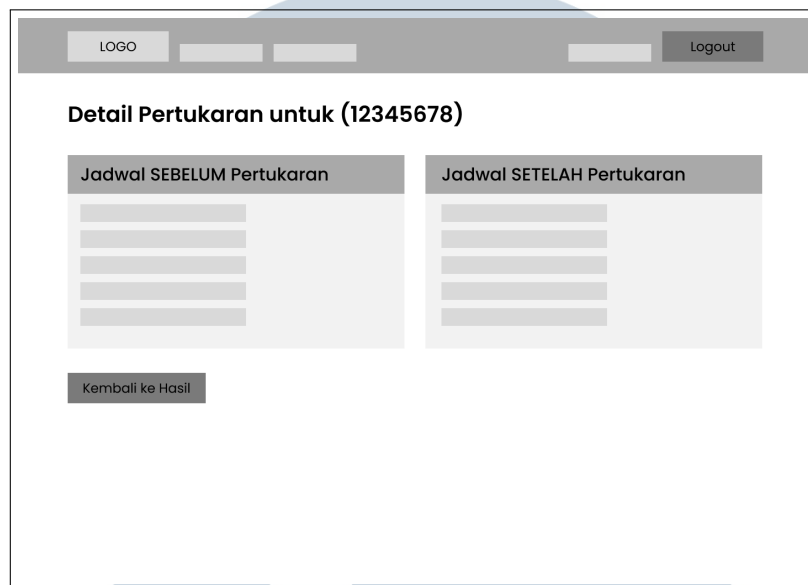
Saat tombol Lihat Laporan Statistik, maka akan menampilkan rangkuman hasil komputasi algoritma TTC seperti pada Gambar 3.22.



Gambar 3.22. Halaman Statistik Admin

Selain itu, ada juga detail pertukaran masing-masing kelas yang dapat dilihat dari tombol yang ada pada tabel seperti Gambar 3.23. Halaman ini memiliki tampilan yang sama antara halaman admin dan mahasiswa dimana menampilkan dua tabel, tabel jadwal sebelum pertukaran dan setelah pertukaran terjadi. Ini akan membantu mahasiswa ataupun admin untuk

mengecek apakah pertukaran yang terjadi sesuai atau tidak dengan preferensi mahasiswa per kelas.



Gambar 3.23. Halaman Statistik Admin

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA