

BAB 2

LANDASAN TEORI

2.1 Penelitian Terdahulu

Analisis terhadap penelitian terdahulu dilakukan untuk mengidentifikasi celah penelitian (*research gap*) dan menegaskan kontribusi dari studi ini. Tinjauan ini mencakup penelitian yang berfokus pada pendekatan heuristik untuk pembuatan jadwal awal, serta kajian teoretis mengenai mekanisme pertukaran yang digunakan dalam penelitian ini.

Sebagian besar riset mengenai pembuatan jadwal awal untuk masalah yang kompleks seperti *University Course Timetabling Problem* (UCTP) memanfaatkan pendekatan algoritma heuristik. Algoritma heuristik merupakan sebuah metode yang dirancang untuk menemukan solusi yang cukup baik dalam waktu yang wajar, terutama untuk masalah optimasi yang terlalu sulit untuk ditemukan solusi optimalnya secara pasti [6]. Alih-alih menjamin solusi terbaik, pendekatan ini menggunakan jalan pintas komputasi untuk mencapai solusi yang mendekati optimal. Berbagai ulasan dan survei terbaru terus menegaskan bahwa UCTP tetap menjadi area riset yang aktif, dengan banyak pendekatan baru dikembangkan untuk menyeimbangkan batasan-batasan kompleks [6][7][8][9].

Pendekatan lain dalam literatur berfokus pada mekanisme alokasi dan pertukaran yang terjadi setelah proses penjadwalan awal, yang lebih relevan dengan konteks penelitian ini. Contohnya, dalam konteks alokasi mata kuliah atau sumber daya di institusi pendidikan, beberapa studi membahas mekanisme pertukaran berbasis preferensi. Studi oleh Josue Ortega dan Klein (2023) menunjukkan bahwa mekanisme pertukaran yang dirancang dengan baik dapat meningkatkan efisiensi alokasi dan keadilan (*fairness*) di antara partisipan, dengan memperhatikan preferensi mahasiswa serta struktur *slot* jadwal yang tersedia [10][11]. Dalam kajian tersebut, mekanisme seperti *bidding* ataupun variasi pertukaran dinamis juga telah dieksplorasi untuk mengakomodasi preferensi dan prioritas secara adaptif [12].

Sementara itu, penelitian oleh Erik Budish et al. (2021) dalam konteks pasar pertukaran multi-unit secara spesifik mengkaji mekanisme pertukaran mata kuliah (*course exchanges*), khususnya dalam konteks pertukaran *many-to-many*. Mereka mengusulkan sebuah mekanisme berbasis pasar (*Approximate Competitive*

Equilibrium from Equal Incomes – A-CEEI) yang bertujuan untuk menemukan harga dan alokasi yang mendekati keseimbangan kompetitif. Metode tersebut menggunakan optimasi untuk menangani kompleksitas pertukaran banyak objek sekaligus, dan hasil penelitian menunjukkan bahwa mekanisme yang diusulkan mampu diproses dalam waktu komputasi yang wajar serta tetap stabil pada skala data yang besar [13].

Di sisi lain, terdapat kajian teoretis mendalam mengenai mekanisme pertukaran itu sendiri, yang menggunakan pendekatan berbeda dari heuristik. Dur dan Paiement (2024) membahas algoritma TTC dalam konteks pemilihan sekolah. Studi ini merupakan analisis teoretis yang mengukuhkan bahwa TTC adalah mekanisme yang sangat andal. Hasilnya membuktikan bahwa alokasi yang dihasilkan TTC tidak dapat diubah lagi untuk menguntungkan satu pihak tanpa merugikan pihak lain. Selain itu, mekanismenya memastikan bahwa strategi terbaik bagi setiap peserta adalah menyatakan preferensi mereka yang sebenarnya [5]. Untuk memvisualisasikan posisi penelitian ini terhadap literatur yang ada, disajikan matriks perbandingan pada Tabel 2.1.

Tabel 2.1. Matriks Perbandingan Fitur Penelitian

No	Aspek / Fitur Utama	Ortega & Klein (2023) [10]	Budish et al. (2021) [13]	Dur & Paiement (2024) [5]	Penelitian Ini
1.	Top Trading Cycles (TTC)	✗	✗	✓	✓
2.	Pertukaran Jadwal Mahasiswa	✓	✓	✗	✓
3.	Berbasis Preferensi Mahasiswa	✓	✓	✓	✓
4.	Menggunakan Data Jadwal Riil	✗	✗	✗	✓
5.	Implementasi Sistem Aplikasi	✗	✗	✗	✓

Dari tinjauan di atas, terlihat sebuah pola yang jelas. Riset dalam penjadwalan akademik mayoritas terkonsentrasi pada masalah pertukaran jadwal. Sementara itu, kajian mengenai algoritma *Top Trading Cycles* (TTC) lebih bersifat teoretis dan diterapkan pada domain di luar perkuliahan. Tabel 2.1 secara visual menegaskan bahwa belum ada penelitian yang menggabungkan kelima aspek kunci tersebut. Hal ini menunjukkan adanya celah penelitian (*research gap*), yaitu kurangnya implementasi sistem aplikasi yang menerapkan algoritma TTC berbasis preferensi untuk menangani masalah pertukaran jadwal mahasiswa menggunakan data riil. Penelitian mengenai implementasi praktis dan studi kasus TTC dalam konteks spesifik seperti ini masih terus dikembangkan [14]. Penelitian ini bertujuan untuk mengisi celah tersebut.

2.2 Sistem Penjadwalan Perkuliahan

Sistem penjadwalan perkuliahan di perguruan tinggi merupakan inti dari permasalahan yang dikenal dalam literatur ilmu komputer sebagai UCTP. Secara definitif, UCTP adalah sebuah masalah optimasi dan pemenuhan batasan (*constraint satisfaction*) yang bertujuan untuk mengalokasikan serangkaian acara atau mata kuliah ke dalam slot waktu dan ruang yang terbatas, dengan tunduk pada berbagai batasan yang ada [8][9]. Komponen utama yang terlibat dalam sistem ini meliputi mahasiswa sebagai peserta, mata kuliah sebagai materi ajar, dosen sebagai pengajar, serta ruang kelas dan waktu sebagai sumber daya yang terbatas. Batasan-batasan (*constraints*) dalam UCTP dapat dikategorikan menjadi dua jenis [9].

Jenis yang pertama adalah batasan keras (*hard constraints*), yaitu aturan yang tidak boleh dilanggar sama sekali, misalnya satu mahasiswa tidak bisa mengambil kelas pada waktu yang sama dan jenis yang kedua adalah batasan lunak (*soft constraints*), yaitu kondisi yang sebaiknya dipenuhi untuk meningkatkan kualitas jadwal, contohnya seperti menghindari jadwal kosong yang terlalu lama bagi mahasiswa [6][15]. Kompleksitas dari interaksi antar komponen dan batasan ini menjadikan UCTP sebagai masalah NP-hard, di mana menemukan satu solusi optimal secara komputasi sangatlah sulit dan memakan waktu [6]. Status NP-hard ini telah dikonfirmasi berulang kali dalam literatur ilmu komputer *modern*, yang menegaskan perlunya algoritma heuristik atau mekanisme spesifik untuk menemukan solusi yang dapat digunakan [16].

NP-hard (*Non-deterministic Polynomial-time hard*) merujuk pada kelas masalah komputasi yang setidaknya sesulit masalah tersulit di kelas NP [16]. Masalah NP adalah masalah di mana solusi yang diberikan dapat diverifikasi kebenarannya dalam waktu polinomial [16]. Implikasi penting dari suatu masalah diklasifikasikan sebagai NP-hard, seperti masalah penjadwalan, adalah ketika tidak ada algoritma waktu polinomial yang diketahui yang dapat menjamin penemuan solusi optimal untuk semua kemungkinan *input* masalah tersebut [16].

2.3 Algoritma Top Trading Cycles

Algoritma Top Trading Cycles ini adalah mekanisme yang digunakan dalam penelitian ini untuk memfasilitasi pertukaran jadwal. Algoritma ini pertama kali dikonseptualisasikan dari sebuah model ekonomi yang dikenal sebagai *housing market problem*, di mana terdapat sekelompok agen yang masing-masing memiliki

sebuah rumah dan ingin menukarinya dengan rumah milik agen lain berdasarkan preferensi mereka [17]. Model dasar ini telah menjadi subjek dari berbagai generalisasi dan analisis untuk menangani preferensi yang lebih kompleks atau batasan tambahan [18]. Dalam konteks penelitian ini, agen adalah mahasiswa, dan objek atau rumah adalah jadwal kelas yang mereka miliki. Setiap mahasiswa mendaftarkan daftar preferensi, yaitu urutan jadwal kelas lain yang mereka inginkan, dari yang paling diminati hingga yang kurang diminati.

Mekanisme algoritma TTC bekerja melalui serangkaian langkah iteratif yang sistematis. Sebelum menjabarkan langkah-langkahnya, berikut adalah notasi formal yang digunakan untuk mendefinisikan model:

- $N = \{a_1, a_2, \dots, a_n\}$ adalah himpunan dari semua agen (mahasiswa) yang berpartisipasi.
- $H = \{h_1, h_2, \dots, h_n\}$ adalah himpunan dari semua objek (jadwal kelas) yang tersedia untuk dipertukarkan.
- $\pi(a_i)$ adalah fungsi alokasi awal yang menunjukkan kepemilikan, yaitu objek h_j yang dimiliki oleh agen a_i .
- P_i adalah daftar urutan preferensi dari agen a_i terhadap objek-objek di H .
- $G = (V, E)$ adalah graf preferensi. Ini merupakan representasi visual dari hubungan preferensi pada setiap langkah algoritma. Graf ini terdiri dari simpul (V) yang merepresentasikan agen aktif, dan sisi berarah (E) yang menunjukkan preferensi teratas setiap agen. Graf ini dinamis dan diperbarui pada setiap iterasi.
- V adalah himpunan simpul (*vertices*) dalam graf G . Himpunan ini berisi semua agen (mahasiswa) yang masih aktif dalam proses pertukaran pada iterasi saat itu. Awalnya, V sama dengan N , namun akan menyusut seiring agen dikeluarkan setelah berhasil bertukar.
- E adalah singkatan dari *edges*, yang merupakan himpunan atau daftar dari semua sisi berarah (panah) yang ada di dalam graf tersebut pada satu iterasi.
- C adalah sebuah siklus perdagangan (*cycle*). Ini adalah sebuah sub-himpunan dari V yang membentuk jalur tertutup dalam graf G . Contohnya, $C = (a_1, a_2, \dots, a_k)$ di mana a_1 menunjuk a_2 , a_2 menunjuk a_3 , ..., hingga a_k

menunjuk kembali ke a_1 . Penemuan siklus C inilah yang memicu eksekusi pertukaran di antara anggota siklus tersebut.

Dengan menggunakan notasi di atas, berikut adalah langkah-langkah kerja algoritma TTC:

1. Inisiasi Graf Preferensi: Pada setiap iterasi, sebuah graf berarah $G = (V, E)$ dibentuk. Himpunan simpul V adalah himpunan agen yang masih aktif (N_{aktif}). Setiap agen $a_i \in V$ akan membentuk sebuah sisi berarah (panah) $(a_i, a_j) \in E$ yang menunjuk ke agen a_j , di mana jadwal yang dimiliki oleh a_j (yaitu $\pi(a_j)$) merupakan preferensi teratas bagi a_i di antara semua jadwal yang dimiliki oleh agen lain yang masih aktif di V .
2. Identifikasi Siklus: Karena jumlah agen di V terbatas, pembentukan graf pada langkah sebelumnya dijamin akan menghasilkan setidaknya satu siklus perdagangan (*cycle*). Sebuah siklus C adalah sebuah urutan agen (a_1, a_2, \dots, a_k) di mana $(a_1, a_2) \in E, (a_2, a_3) \in E, \dots, (a_k, a_1) \in E$. Ini berarti agen a_1 menginginkan jadwal milik a_2 , a_2 menginginkan jadwal milik a_3 , dan seterusnya hingga a_k menginginkan jadwal milik a_1 .
3. Eksekusi Pertukaran: Semua agen yang berada di dalam siklus C yang teridentifikasi akan melakukan pertukaran jadwal secara simultan. Alokasi baru π' untuk setiap agen $a_i \in C$ akan menjadi jadwal yang dimiliki oleh agen yang ditunjuknya. Sebagai contoh, untuk siklus $C = (a_1, \dots, a_k)$, alokasi baru menjadi $\pi'(a_1) = \pi(a_2), \pi'(a_2) = \pi(a_3),$ dan seterusnya hingga $\pi'(a_k) = \pi(a_1)$.
4. Penghapusan Agen: Mahasiswa yang telah berhasil bertukar akan mendapatkan jadwal barunya dan dikeluarkan dari proses selanjutnya. Secara formal, himpunan agen aktif diperbarui menjadi $N_{aktif} = N_{aktif} \setminus C$, di mana semua agen yang berada di siklus C dihilangkan.
5. Iterasi: Proses ini diulangi kembali dari langkah pertama dengan himpunan agen aktif yang sudah diperbarui, hingga tidak ada lagi agen yang tersisa ($N_{aktif} = \emptyset$) atau tidak ada siklus yang bisa dibentuk.

Algoritma Top Trading Cycles (TTC) memiliki beberapa jaminan teoretis yang kuat dalam konteks mekanisme alokasi dan pertukaran. Salah satu sifat utama yang dijamin adalah *Pareto Efficiency*. Secara intuitif, sebuah alokasi

dikatakan Pareto-optimal apabila tidak terdapat alokasi alternatif lain yang dapat meningkatkan kesejahteraan seorang mahasiswa tanpa memperburuk kondisi mahasiswa lainnya [19].

Secara formal, misalkan N adalah himpunan mahasiswa dan A adalah suatu alokasi, di mana $A(i)$ menyatakan kelas yang diterima oleh mahasiswa i . Alokasi A disebut Pareto-optimal jika dan hanya jika tidak ada alokasi lain A' sehingga:

$$\forall j \in N, A'(j) \succeq_j A(j) \quad \text{dan} \quad \exists i \in N, A'(i) \succ_i A(i) \quad (2.1)$$

dan secara bersamaan

$$f(P_i, P_{-i}) \succeq_i f(P'_i, P_{-i}) \quad (2.2)$$

di mana simbol \succ_i menyatakan bahwa mahasiswa i lebih menyukai alokasi $A'(i)$ dibandingkan $A(i)$, dan \succeq_j menyatakan bahwa mahasiswa j tidak lebih buruk pada alokasi baru. Dengan kata lain, tidak ada perubahan alokasi yang dapat menguntungkan setidaknya satu mahasiswa tanpa merugikan mahasiswa lain, sehingga solusi TTC bersifat efisien secara Pareto.

Selain itu, TTC juga memenuhi sifat *Strategy-Proofness*, yaitu kondisi di mana strategi terbaik bagi setiap mahasiswa adalah melaporkan preferensi mereka yang sebenarnya. Artinya, tidak ada mahasiswa yang dapat memperoleh hasil yang lebih baik dengan memanipulasi atau memalsukan urutan preferensinya [20].

Secara matematis, suatu mekanisme alokasi f dikatakan *strategy-proof* jika untuk setiap mahasiswa $i \in N$, setiap profil preferensi asli P_i , setiap alternatif preferensi palsu P'_i , dan preferensi mahasiswa lain P_{-i} , berlaku:

$$A(i) \succeq_i A_0(i), \forall i \in N \quad (2.3)$$

yang berarti bahwa hasil yang diperoleh dengan melaporkan preferensi sebenarnya selalu setidaknya sama baiknya dibandingkan hasil yang diperoleh dengan melaporkan preferensi yang dimanipulasi.

Sifat *strategy-proofness* ini menjadikan TTC sangat sesuai untuk sistem pertukaran kelas akademik, karena menjamin bahwa mahasiswa tidak memiliki insentif untuk berbohong, sehingga mekanisme menjadi lebih adil, transparan, dan

stabil. Penelitian terkini juga terus memperkuat bukti formal mengenai sifat ini serta mengeksplorasi batasan-batasannya dalam model preferensi yang lebih umum [21].

2.4 Metrik Evaluasi Pertukaran Jadwal

Untuk mengukur keberhasilan dan kinerja sistem yang diusulkan, diperlukan serangkaian metrik evaluasi yang objektif. Metrik ini akan digunakan pada tahap pengujian di Bab 4 untuk memvalidasi fungsionalitas sistem. Metrik yang akan digunakan dapat dikelompokkan menjadi dua kategori, yaitu metrik keberhasilan alokasi dan metrik kinerja komputasi.

Metrik keberhasilan alokasi bertujuan untuk mengukur sejauh mana sistem berhasil memenuhi permintaan pengguna. Metrik ini dapat diukur secara kuantitatif sebagai berikut:

1. Tingkat Keberhasilan Pertukaran

Mengukur persentase mahasiswa yang berhasil mendapatkan jadwal baru dari total partisipan. Metrik ini dihitung dengan rumus:

$$\text{Tingkat Keberhasilan} = \frac{M_{\text{sukses}}}{M_{\text{total}}} \times 100\% \quad (2.4)$$

di mana:

- M_{sukses} = Jumlah Mahasiswa yang Berhasil Tukar
- M_{total} = Total Mahasiswa yang Mengajukan Permintaan

2. Tingkat Pemenuhan Preferensi Top-k

Mengukur kualitas pertukaran bagi mahasiswa yang berhasil. Metrik ini menghitung persentase mahasiswa yang mendapatkan jadwal yang berada di peringkat k teratas dalam daftar preferensi mereka.

$$\text{Tingkat Pemenuhan Top-k} = \frac{M_{\text{top-k}}}{M_{\text{sukses}}} \times 100\% \quad (2.5)$$

di mana:

- $M_{\text{top-k}}$ = Jumlah Mahasiswa yang Mendapat Jadwal di Peringkat k Teratas
- M_{sukses} = Jumlah Mahasiswa yang Berhasil Tukar
- k = Batas peringkat yang ditentukan

3. Distribusi Ukuran Siklus

Metrik ini merupakan analisis statistik untuk memberikan wawasan mengenai efisiensi pasar. Metrik ini tidak berupa rumus tunggal, melainkan himpunan dari jumlah total siklus yang ditemukan untuk setiap ukuran k :

$$\{|C_2|, |C_3|, \dots, |C_m|\} \quad (2.6)$$

di mana $|C_k|$ adalah jumlah total siklus yang melibatkan k mahasiswa. Hasil ini menunjukkan kemampuan sistem dalam menemukan pertukaran sederhana (siklus 2-arah) hingga pertukaran yang kompleks (siklus dengan $k > 2$).

Metrik kinerja komputasi bertujuan untuk mengukur efisiensi teknis dari implementasi algoritma. Analisis ini tidak mengukur waktu dalam detik, melainkan memprediksi bagaimana laju pertumbuhan waktu eksekusi akan berskala seiring bertambahnya ukuran *input* (n) [22].

1. Notasi *Big Omega* (Ω , Best-Case)

Notasi *Big Omega* digunakan untuk menggambarkan batas bawah (*lower bound*) atau skenario terbaik dari algoritma. Untuk algoritma TTC, skenario kasus terbaik secara teoretis dapat terjadi jika semua n partisipan membentuk siklus 2-arah yang sederhana. Dalam kasus ini, algoritma hanya perlu memproses setiap agen kira-kira satu kali untuk menemukan dan mengeksekusi semua siklus pada iterasi pertama, sehingga kompleksitas waktunya adalah:

$$\Omega(n) \quad (2.7)$$

di mana n adalah jumlah agen yang berpartisipasi.

2. Notasi *Big O* (Worst-Case)

Efisiensi algoritma TTC dapat dianalisis secara teoretis menggunakan notasi *Big O* untuk menentukan batas atas (*upper bound*) atau skenario terburuk. Untuk kasus di mana preferensi bersifat tegas (*strict preferences*), seperti yang diimplementasikan dalam penelitian ini di mana mahasiswa memberikan urutan preferensi yang pasti, kompleksitas waktu total di skenario terburuk adalah:

$$O(n^2) \quad (2.8)$$

di mana n adalah jumlah agen (mahasiswa) yang berpartisipasi. Ini mengindikasikan bahwa waktu eksekusi dapat tumbuh sejalan dengan pangkat dua dari jumlah partisipan.

2.5 Justifikasi Pemilihan Algoritma TTC

Pemilihan algoritma *Top Trading Cycles* (TTC) sebagai inti dari sistem ini didasarkan pada keunggulannya yang signifikan dibandingkan pendekatan alternatif dalam menangani masalah pertukaran pasca-alokasi. Tabel 2.2 menyajikan perbandingan komprehensif antara TTC dengan mekanisme alternatif seperti algoritma heuristik dan sistem *First-Come, First-Served* (FCFS).

Tabel 2.2. Perbandingan Terbukti: Algoritma TTC dengan Mekanisme Alternatif

Karakteristik	Top Trading Cycles (TTC)	Algoritma Heuristik	First-Come, First-Served (FCFS)
Sifat Hasil	Menghasilkan alokasi unik yang konsisten untuk setiap set preferensi tetap [19].	Hasil eksekusi dapat bervariasi meskipun menggunakan data <i>input</i> yang identik [22][6].	Hasil hanya ditentukan oleh stempel waktu tanpa mempertimbangkan nilai preferensi.
Optimalitas	Terbukti secara matematis mencapai alokasi optimal di mana tidak ada agen yang bisa untung tanpa merugikan orang lain [2].	Berisiko terjebak pada solusi sub-optimal karena gagal mendeteksi siklus pertukaran kompleks [6].	Mengabaikan potensi pertukaran yang saling menguntungkan [10].
Ketahanan Strategi	Melaporkan preferensi secara jujur merupakan strategi dominan bagi pengguna [23].	Partisipan cenderung memodifikasi urutan pilihan untuk mengantisipasi ketidakpastian luaran algoritma [6].	Mendorong penggunaan bot atau serangan <i>spamming</i> demi memenangkan urutan waktu.
Keamanan Partisipan	Menjamin mahasiswa tidak mendapatkan jadwal yang lebih buruk dari alokasi awal mereka [24].	Tidak memiliki jaminan matematis untuk mempertahankan nilai alokasi awal partisipan [6].	Partisipan yang terlambat kehilangan hak pertukaran sepenuhnya tanpa kompensasi preferensi [10].
Fokus Operasional	Mengoptimalkan kepuasan seluruh partisipan secara simultan [14].	Sering kali mengorbankan akurasi dan keadilan demi solusi cepat pada skala besar.	Fokus tunggal pada penyelesaian antrian berdasarkan urutan kedatangan.

Algoritma heuristik yang sering diimplementasikan dalam optimasi pertukaran jadwal merupakan kategori algoritma *non-deterministic*. Algoritma *non-deterministik* adalah algoritma yang dapat menghasilkan *output* yang berbeda atau menempuh jalur eksekusi yang berbeda pada setiap pengujian, meskipun

diberikan set data *input* yang sama [22]. TTC menawarkan hasil yang sepenuhnya *deterministic* dan terbukti optimal secara *Pareto* [19]. Sebaliknya, algoritma heuristik yang bersifat probabilistik dirancang untuk mencari solusi yang cukup baik tanpa jaminan teoretis yang pasti terkait stabilitas hasil [6]. Keunggulan fundamental TTC terletak pada kemampuannya memberikan jaminan *Pareto Optimality*, *Strategy-Proofness*, dan *Individual Rationality*. Melalui karakteristik tersebut, TTC memastikan bahwa proses pertukaran jadwal perkuliahan berlangsung secara adil dan konsisten, serta mampu memaksimalkan kepuasan kolektif seluruh mahasiswa tanpa risiko kerugian bagi pihak manapun.

