



# Optimasi proses penjadwalan mata kuliah menggunakan algoritme genetika dan pencarian tabu

## Course scheduling optimization using genetic algorithm and tabu search

Arif Amrulloh<sup>\*)</sup>, Enny Itje Sela

Program Magister Teknologi Informasi, Program Pascasarjana, Universitas Teknologi Yogyakarta  
Jl. Siliwangi (Ringroad Utara), Jombor, Sleman, D.I. Yogyakarta, Indonesia 55285

**Cara sitasi:** A. Amrulloh and E. I. Sela, "Optimasi proses penjadwalan mata kuliah menggunakan algoritme genetika dan pencarian tabu," *Jurnal Teknologi dan Sistem Komputer*, vol. 9, no. 3, pp. 157-166, 2021. doi: [10.14710/jtsiskom.2021.14137](https://doi.org/10.14710/jtsiskom.2021.14137), [Online].

**Abstract - Scheduling courses in higher education often face problems, such as the clashes of teachers' schedules, rooms, and students' schedules. This study proposes course scheduling optimization using genetic algorithms and taboo search. The genetic algorithm produces the best generation of chromosomes composed of lecturer, day, and hour genes. The Tabu search method is used for the lecture rooms division. Scheduling is carried out for the Informatics faculty with four study programs, 65 lecturers, 93 courses, 265 lecturer assignments, and 65 classes. The process of generating 265 schedules took 561 seconds without any scheduling clashes. The genetic algorithms and taboo searches can process quite many course schedules faster than the manual method.**

**Keywords** – optimization; courses scheduling; genetic algorithms; tabu search

**Abstrak - Penjadwalan mata kuliah merupakan permasalahan yang sering terjadi pada perguruan tinggi, di antaranya adalah bentrok waktu mengajar dosen, ruangan dan kelas mahasiswa. Kajian ini mengusulkan optimasi penjadwalan mata kuliah menggunakan algoritme genetika dan pencarian tabu. Algoritme genetika berfungsi untuk menghasilkan generasi terbaik kromosom yang tersusun atas gen dosen, hari, dan jam. Pencarian tabu digunakan untuk pembagian ruang perkuliahan. Penjadwalan dilakukan di fakultas Informatika yang mempunyai empat program studi dengan 65 dosen, 93 mata kuliah, 265 penugasan dosen, dan 65 kelas. Proses pembangkitan 265 jadwal membutuhkan waktu selama 561 detik dan tidak ada bentrok yang terjadi. Kombinasi algoritme genetika dan pencarian tabu dapat memproses jadwal mata kuliah yang cukup banyak dengan lebih cepat daripada cara manual.**

**Kata kunci** – optimasi; penjadwalan mata kuliah; algoritme genetika; pencarian tabu

## I. PENDAHULUAN

Penjadwalan mata kuliah merupakan salah satu masalah penelitian yang menantang yang melibatkan banyak komponen. Karena sifatnya yang menantang, tujuan utama penjadwalan sering kali hanya untuk menemukan solusi yang layak daripada mencari solusi yang optimal [1]. Tujuan utama dari penjadwalan adalah untuk mengatur sejumlah komponen, seperti dosen, mahasiswa, mata kuliah, waktu, dan ruang perkuliahan dalam satu waktu tertentu [2], [3].

Secara umum, penjadwalan mata kuliah merupakan masalah berskala besar karena terdiri dari banyak komponen yang melibatkan ratusan hingga ribuan mahasiswa. Banyaknya komponen yang terlibat mempengaruhi lamanya waktu proses dan risiko terhadap bentrok atau konflik jadwal cukup tinggi [3]. Konflik yang sering terjadi dalam proses penyusunan jadwal mata kuliah di antaranya adalah dosen, mata kuliah, ruang kuliah, dan selot waktu [4]. Masalah yang paling rumit dalam penyusunan jadwal adalah menghindari bentrok, apalagi jika masih menggunakan cara manual menggunakan Microsoft Excel, seperti dalam [5]. Untuk mengatasi permasalahan tersebut, sebuah mekanisme penjadwalan perlu dikembangkan untuk mendapatkan solusi optimal [6].

Algoritme genetika (AG) dan pencarian tabu (TS) dapat dikombinasikan untuk optimasi pada proses penjadwalan mata kuliah. Algoritme genetika melakukan pencarian yang meniru mekanisme dari genetika alam [7]. Dengan meniru teori evolusi, AG dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam dunia nyata yang optimal dan efektif, salah satunya adalah untuk menghasilkan penjadwalan yang optimal [8], [9]. AG cukup baik jika digunakan dalam pembuatan jadwal mata kuliah untuk memecahkan masalah yang cukup besar, meskipun membutuhkan waktu yang lama jika di kerjakan secara manual [2].

Di sisi lain, algoritme TS dapat memecahkan solusi yang kompleks dengan cara menemukan solusi yang tepat dan tidak terdapat pelanggaran terhadap batasan yang ditentukan [10]. Kajian penjadwalan menggunakan AG-TS telah dilakukan dalam [11]-[13]. TS digunakan

<sup>\*)</sup> Penulis korespondensi (Arif Amrulloh)  
Email: [arif.amrulloh@student.uty.ac.id](mailto:arif.amrulloh@student.uty.ac.id)

sebagai filter di dalam proses AG agar kromosom yang telah mengalami *crossover* tidak diproses lagi pada iterasi selanjutnya. Kombinasi AG-TS terbukti dapat memberikan hasil yang lebih optimal dibandingkan menggunakan AG [14]. Kajian ini bertujuan untuk melakukan optimasi penjadwalan kuliah menggunakan AG-TS. Berbeda dengan kajian AG-TS sebelumnya, algoritme TS dalam kajian ini diimplementasikan setelah mendapat generasi terbaik hasil AG. Kajian ini juga menguji penerapan kombinasi AG-TS dalam memberikan hasil yang optimal, yaitu lamanya proses pembangkitan jadwal dan bentrok yang terjadi.

## II. METODE PENELITIAN

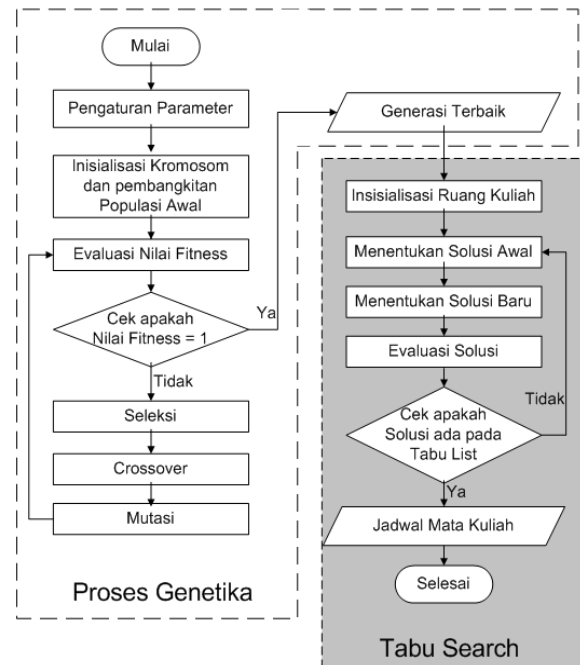
Data yang digunakan pada penelitian ini diambil dari basis data di salah satu perguruan tinggi yang berada di Purwokerto. Data-data tersebut adalah data fakultas, program studi (*prodi*), ruang perkuliahan, waktu perkuliahan, tahun akademik, mata kuliah, dan dosen. Basis data yang dibangun terdiri dari komponen utama penjadwalan mata kuliah dan komponen pengembangan dari beberapa komponen utama. Komponen utama terdiri dari tabel fakultas, *prodi*, ruang, hari, jam, mata kuliah, dosen, dan tahun akademik. Komponen pengembangan terdiri dari tabel *fitness* yang akan digunakan untuk menyimpan nilai *fitness* hasil perhitungan dari *crossover* dan mutasi, tabel kromosom digunakan untuk menyimpan kromosom, dan tabel *plotting* jadwal mengajar dosen yang di antaranya berisi komponen dosen, mata kuliah, kelas mata kuliah.

Proses pembuatan jadwal mata kuliah yang diusulkan menggunakan kombinasi AG dan TS. Algoritme genetika berfungsi untuk menghasilkan generasi terbaik kromosom yang tersusun dari gen dosen, hari, dan jam. Pencarian TS digunakan untuk pembagian ruang perkuliahan. Proses genetika akan berhenti pada generasi terbaik, sedangkan pencarian TS akan dimulai dari generasi terbaik yang dihasilkan oleh proses genetika. Metode yang diusulkan untuk penjadwalan ini dapat dilihat pada Gambar 1.

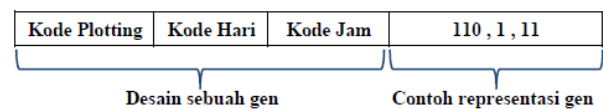
### Algoritme genetika

Parameter genetika meliputi jumlah generasi (*iterasi*) yang digunakan sebagai batasan untuk menghindari proses iterasi yang tak terhingga. Ukuran populasi menunjukkan banyaknya populasi awal yang akan dijadikan sebagai induk. *Crossover* dan mutasi merupakan proses pembentukan populasi baru. Nilai pada parameter genetika disesuaikan untuk mendapatkan hasil yang optimal ditunjukkan pada Tabel 1.

Inisialisasi kromosom dinyatakan dengan angka yang merupakan pengkodean (*encoding*) dari penyelesaian asli suatu masalah. Pengkodean tersebut meliputi penyediaan gen dengan satu gen mewakili satu variabel [15]. Variabel yang digunakan dalam representasi kromosom ini adalah *plotting* dosen, hari, dan jam. Desain dan representasi gen pembentuk kromosom dapat dilihat pada Gambar 2. Setiap gen yang terbentuk mewakili waktu mengajar dosen yang



Gambar 1. Metode yang diusulkan



Gambar 2. Desain dan representasi sebuah gen

Tabel 1. Parameter Genetika

Keterangan	Jumlah
Generasi ( <i>iterasi</i> )	250
Populasi	10
<i>Crossover</i> ( <i>kawin silang</i> )	0,7
Mutasi	0,3

Tabel 2. Contoh tabel *plotting* mengajar dosen

Kode Plotting	Kode Dosen	Kode Mata Kuliah	Kelas
110	IFA	IF6415403	S1IF-05-SC1
111	SDA	IS6211703	S1SI-03-D
112	AGZ	IF6415503	S1IF-05-SC1
113	PRD	IN0000703	S1SI-02-C
114	STS	IS6211802	S1IF-07-X2

meliputi *string* Kode Plotting (Tabel 2), Kode Hari (Tabel 3), dan Kode Jam (Tabel 4). Kode Hari menunjukkan hari-hari yang digunakan untuk kuliah, yaitu sebanyak 5 hari, sedangkan Kode Jam menunjukkan jam yang digunakan untuk kuliah sebanyak 19 sesi perkuliahan.

Kode Plotting mewakili data dosen beserta mata kuliah dan kelas yang diampu. Contoh *plotting* mengajar dosen

Allele (nilai gen) berisi jadwal mengajar dosen

110,1,10	111,2,11	112,3,12	113,3,12	114,5,14	.....n
Gen1	Gen2	Gen3	Gen4	Gen5	Gen...n

**Gambar 3.** Representasi kromosom

dapat dilihat pada [Tabel 2](#). Data berasal dari beberapa data pendukung. Pertama, data dosen yang berisi informasi id dosen, kode dosen, dan nama dosen. Kedua, data mata kuliah yang berisi informasi id mata kuliah, kode mata kuliah, nama mata kuliah, dan id prodi. Selain itu, data pendukung berisi informasi tambahan yang tidak berkaitan dengan data pendukung lain, yaitu kelas perkuliahan yang dimasukkan secara manual pada saat dilakukan proses *plotting* mengajar dosen. Kode *string* 110 pada [Gambar 2](#) adalah kode *plotting* untuk dosen IFA yang mengajar mata kuliah IF641540 pada kelas S1IF-06-TI.

Kode dosen IFA merupakan akronim dari dosen dengan nama Ipam Fuadina Adam. Kode mata kuliah IF6415403 memiliki arti, yaitu dua digit pertama mewakili prodi Informatika yang disingkat IF, satu digit yaitu 6 mewakili level KKNi, satu digit mewakili bahwa mata kuliah dilaksanakan di tingkat 4, satu digit mewakili waktu dilaksanakan perkuliahan, yaitu 1 untuk semester ganjil, dua digit mewakili mata kuliah ke-54 dari prodi Informatika, satu digit bernilai 0 merupakan kode untuk mata kuliah teori, dan satu digit terakhir mewakili jumlah SKS.

Setiap digit pada Kelas S1IF-05-SC1 memiliki arti bahwa dua digit pertama mewakili jenjang pendidikan S1, dua digit mewakili prodi Informatika yang disingkat IF, dua digit setelah tanda pemisah pertama, yaitu 05, mewakili angkatan masuk mahasiswa, dan digit terakhir setelah tanda pemisah kedua, yaitu SC1, adalah kode unik yang membedakan masing-masing kelas pada prodi tertentu untuk tahun angkatan yang sama. Selanjutnya, *string* 1 menunjukkan hari Senin berasal dari [Tabel 3](#), dan *string* 11 menunjukkan jam belajar 07:00 berasal dari [Tabel 4](#). Kode *Plotting* adalah nomor urut dari daftar mengajar dosen yang dibangkitkan secara otomatis oleh sistem pada saat proses dijalankan. Gabungan dari gen-gen yang membentuk nilai tertentu disebut dengan kromosom. Representasi kromosom dapat dilihat pada [Gambar 3](#).

Pembangkitkan populasi awal dilakukan secara acak. Panjang gen pembentuk kromosom adalah sebanyak jadwal yang tersedia. Proses pembangkitan populasi awal melakukan pengecekan terhadap jumlah SKS dari mata kuliah yang akan dijadwalkan dengan alokasi waktu untuk 1 SKS adalah 50 menit. Jika mata kuliah yang dijadwalkan berjumlah 3 SKS, maka waktu yang dialokasikan untuk 1 mata kuliah adalah 150 menit. Misalnya, jika perkuliahan dengan 3 SKS dimulai jam 07:00, maka perkuliahan tersebut akan berakhir pada pukul 09:30. Pada rentang pukul 07:00 sampai 09:30 tidak boleh ada perkuliahan lain yang berjalan di hari dan ruang yang sama. Pengecekan waktu akhir perkuliahan dilakukan berdasarkan jam istirahat dan jam akhir kegiatan perkuliahan. Jam istirahat pada hari Senin

**Tabel 3.** Hari perkuliahan

Kode Hari	Nama Hari
1	SENIN
2	SELASA
3	RABU
4	KAMIS
5	JUMAT

**Tabel 4.** Desain jam perkuliahan

Sesi	Kode Jam	Jam
Sesi 1	11	07:00
Sesi 2	12	07:30
Sesi 3	13	08:00
Sesi 4	14	08:30
Sesi 5	15	09:00
Sesi 6	16	09:30
Sesi 7	17	10:00
Sesi 8	18	10:30
Sesi 9	19	11:00
Sesi 10	20	11:30
...	...	...
Sesi 19	29	17:00

**Tabel 5.** Batasan pembuatan jadwal mata kuliah

Batasan	Ketentuan
Batasan Keras	Tidak boleh ada perkuliahan lain yang berjalan di hari dan jam yang sama pada saat satu perkuliahan sedang berlangsung Waktu istirahat dan batas akhir jam kerja yang boleh dilanggar dalam satu sesi perkuliahan tidak boleh lebih dari 30 menit Pada hari Jumat waktu istirahat tidak boleh dilanggar
Batasan Lunak	Waktu istirahat yang boleh dilanggar oleh batas berakhirnya perkuliahan adalah 30 menit untuk hari Senin sampai Kamis

sampai Kamis adalah pukul 12:00 sampai 13:00, jam istirahat pada hari Jumat adalah pukul 11:30 sampai 13:00, dan jam akhir kegiatan perkuliahan sesuai jam kerja adalah pukul 17:00.

Dalam proses pembuatan jadwal terdapat batasan-batasan. Pertama, batasan keras (*hard constraint*) menyatakan batasan yang wajib dipenuhi dan tidak boleh dilanggar. Kedua, batasan lunak (*soft constraint*) merupakan batasan yang boleh dilanggar tetapi seminimal mungkin dihindari. Batasan pada proses pembangkitan populasi awal dapat dilihat pada [Tabel 5](#).

Diagram alir proses pembangkitan populasi awal dapat dilihat pada [Gambar 4](#). Tujuan dibuatnya aturan pada proses pembangkitan populasi awal adalah untuk mengontrol alokasi waktu yang tersedia dan meminimalkan bentrok jam mengajar sehingga pada proses genetika selanjutnya tidak perlu lagi dilakukan evaluasi terhadap jumlah SKS yang berpengaruh terhadap alokasi waktu yang dibutuhkan.

**Tabel 6.** Batasan atau *constraint* nilai *fitness*

No.	Ketentuan yang tidak boleh dilanggar
1	Dosen tidak boleh mengajar lebih dari satu kelas pada hari dan jam yang sama
2	Satu kelas mahasiswa tidak boleh dijadwalkan lebih dari satu kali pada hari dan jam yang sama

Evaluasi nilai *fitness* bertujuan untuk mengetahui kualitas dari setiap kromosom dalam suatu populasi. Fungsi *fitness* digunakan untuk proses evaluasi kromosom agar memperoleh kromosom yang diinginkan dengan mengukur nilai kecocokannya. Fungsi ini membedakan kualitas dari kromosom untuk mengetahui seberapa baik kromosom yang dihasilkan [11]. Proses evaluasi nilai *fitness* ini akan terus berjalan sampai terpenuhinya kriteria berhenti. Evaluasi nilai *fitness* ditentukan dengan menghitung nilai dari setiap pelanggaran yang terjadi dengan menggunakan aturan yang sudah ditentukan di mana setiap aturan yang ditetapkan memiliki bobot nilai 1.

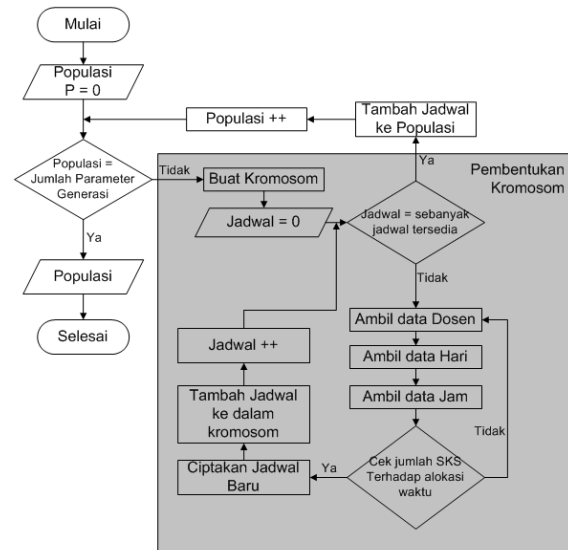
Persamaan yang digunakan untuk menghitung nilai *fitness* dinyatakan pada (1) [16]. Nilai *fitness* dihitung berdasarkan banyaknya jumlah pelanggaran atau penalti pada suatu kromosom [17]. Nilai *fitness* terbaik adalah 1, yaitu jika sudah tidak ada pelanggaran yang terjadi. Nilai yang dihasilkan dari evaluasi *fitness* merepresentasikan banyaknya jumlah pelanggaran. Semakin kecil pelanggaran yang dilakukan, maka solusi yang dihasilkan semakin baik. Bobot nilai yang diberikan untuk setiap pelanggaran adalah 1. Untuk menghindari kesalahan program yang diakibatkan oleh pembagian 0, maka jumlah total bobot pelanggaran ditambahkan nilai 1.

$$Fitness = \frac{1}{1 + penalti} \quad (1)$$

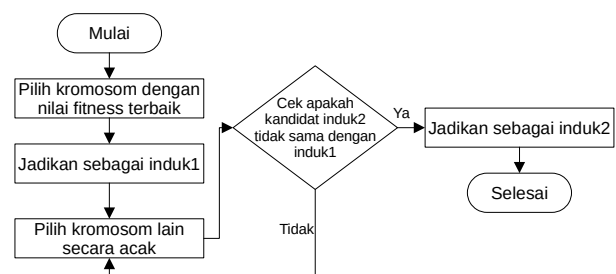
Batasan diperlukan untuk menghindari bentrok jadwal mengajar dosen dan jadwal belajar mahasiswa. Bentrok jadwal akan mempengaruhi proses belajar mengajar karena masing-masing dosen atau mahasiswa tidak bisa mengikuti kegiatan perkuliahan pada satu waktu yang sama. Batasan yang tidak boleh dilanggar dinyatakan pada Tabel 6.

Seleksi adalah proses pemilihan individu terbaik, setelah penghitungan nilai *fitness* terhadap setiap individu selesai maka proses selanjutnya adalah pemilihan induk. Seleksi yang digunakan adalah seleksi roda *roulette*. Pada seleksi roda *roulette*, semakin tinggi nilai *fitness*, maka semakin besar kemungkinan untuk dipilih menjadi induk [18], [19].

Proses seleksi kromosom yang akan dijadikan induk dapat dilihat pada Gambar 5. *Crossover* atau perkawinan silang menggunakan perkawinan silang satu titik, yaitu dengan cara memilih satu individu dengan nilai *fitness* terbaik yang akan dijadikan induk pertama. Selanjutnya, satu individu lagi dipilih secara acak sebagai induk kedua. Kedua induk yang sudah terpilih akan dilakukan kawin silang dengan cara menukar gen pada satu titik



**Gambar 4.** Diagram alir proses pembangkitan populasi awal



**Gambar 5.** Diagram alir proses seleksi

Induk 1	19,2,46	19,3,50	19,4,35	19,5,46	19,1,50	19,2,46
Induk 2	19,3,40	19,4,35	19,5,46	19,1,35	19,2,39	19,3,50
Keturunan 1	19,2,46	19,3,50	19,4,35	19,1,35	19,2,39	19,3,50
Keturunan 2	19,3,40	19,4,35	19,5,46	19,5,46	19,1,50	19,2,46

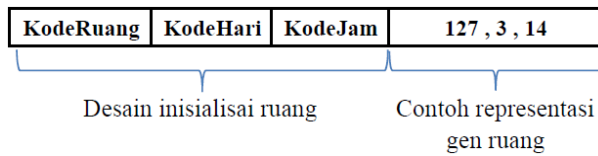
**Gambar 6.** Ilustrasi *crossover* satu titik

yang sudah ditentukan. Dari perkawinan silang tersebut, terbentuk generasi baru (*offspring*). Setiap generasi baru yang terbentuk akan dievaluasi nilai *fitness*-nya. Proses tersebut akan dilakukan terus menerus sampai kriteria yang ditentukan tercapai.

Pada *crossover* satu titik, posisi  $k$ , dengan  $k=1,2,\dots,N-1$  dan  $N$  menyatakan panjang kromosom, diseleksi secara acak. Ilustrasi kromosom satu titik dapat dilihat pada Gambar 6. Pada proses *crossover* akan dilakukan iterasi sampai batas kriteria terpenuhi sesuai parameter yang sudah ditetapkan, yaitu sudah tidak ada pelanggaran yang ditemukan atau nilai *fitness* sama dengan 1, batas maksimum proses *crossover* telah tercapai sesuai parameter yang telah ditetapkan, dan jumlah generasi atau iterasi maksimum tercapai.

Induk 1	28,5,46	28,1,35	28,2,46	28,3,50	28,4,35	28,3,50
Induk 2	28,3,50	28,4,46	28,5,38	28,1,50	28,2,46	28,3,40
Keturunan 1	28,5,46	28,1,35	28,2,46	28,1,50	28,4,35	28,3,50
Keturunan 2	28,3,50	28,4,46	28,5,38	28,3,50	28,2,46	28,3,40

Gambar 7. Ilustrasi proses mutasi



Gambar 8. Desain inialisasi dan representasi gen ruang



Gambar 9. Diagram alir proses penetapan solusi baru

Mutasi dilakukan dengan cara memilih gen yang mengalami pelanggaran dan kemudian dimutasi dengan gen lain dari kromosom lain yang diperoleh secara acak. Setiap individu baru yang terbentuk dari proses mutasi akan dievaluasi nilai *fitness*-nya. Iterasi akan terus dijalankan sampai batas kriteria terpenuhi. Ilustrasi dari proses mutasi dapat dilihat pada Gambar 7. Induk 1 mempunyai dua gen yang sama, yaitu pada kolom ke-4 dan kolom ke-6, sehingga salah satu dari dua gen tersebut harus ditukar dengan gen dari kromosom lain (induk 2) pada titik yang sama.

Generasi terbaik adalah kromosom dengan nilai *fitness* terbaik. Kromosom ini akan digunakan untuk proses pembagian ruang menggunakan algoritme TS. Pada penelitian ini, generasi terbaik ditemukan pada populasi ke-244 dengan nilai *fitness* 1 dan jumlah jadwal sebanyak 265. Ilustrasi solusi terbaik dapat dilihat pada Tabel 7.

### Algoritme pencarian tabu

Setelah generasi terbaik ditemukan, proses selanjutnya adalah pembagian ruangan. Proses pembagian ruangan menggunakan TS dengan empat langkah. Pertama adalah inialisasi ruang perkuliahan yang merupakan hasil kombinasi dari ruang, hari dan jam. Ruang perkuliahan diambil dari tabel ruang yang ada pada basis data, sedangkan hari dan jam berasal dari solusi terbaik jadwal mengajar dosen. Data ruang kuliah disajikan pada Tabel 8. Desain inialisasi ruang perkuliahan dan contoh representasi gen ruang yang mewakili jadwal mengajar dosen dapat dilihat pada Gambar 8.

Tabel 7. Solusi terbaik jadwal mengajar dosen

No	Kode Dosen	Kelas	Hari	Jam	Gen
1	AAB	S1IF-05-C	SENIN	7:30	39,1,35
2	AAB	S1SE-02-A	JUMAT	9:00	39,5,38
3	ABA	S1IF-08-D	SELASA	15:30	156,2,46
4	ACW	S1SI-02-B	KAMIS	10:00	113,5,51
...	...	...	...	...	...
265	YUS	S1SI-03-C	SELASA	9:30	79,5,51

Tabel 8. Tabel ruang perkuliahan

No.	Kode Ruang	Nama Ruang
1	100	IOT-101
2	101	IOT-102
3	102	IOT-103
...	...	.....
65	150	DC-204

Tabel 9. Solusi awal pembagian ruang

No	Dosen	Kelas	Mata kuliah	Hari	Jam	Ruang
1	SDA	S1SD-01-A	Bahasa Indonesia	SENIN	07: 00	IOT-103
2	IFA	S1IF-06-TI	Komputasi Awan	SENIN	09: 30	
3	AGZ	S1IF-07-N	Sistem Operasi	SENIN	13: 00	
4	PRD	S1IF-06-TI	Algoritme Pemrograman	SENIN	07: 30	
5	STS	S1SI-02-B	Kalkulus	SENIN	10: 00	

Kedua adalah menentukan solusi awal yang akan dipilih, yaitu solusi dengan nomor urut terkecil yang ada pada nomor urut 1. Setiap solusi memiliki nomor urut yang mewakili no urut jadwal mengajar dosen. Ilustrasi solusi awal dapat dilihat pada Tabel 9. Solusi awal terdapat pada nomor satu dengan ruang perkuliahan di IOT-103. Pembagian ruang dilakukan dengan cara mengambil ruang secara acak dari tabel ruang.

Ketiga adalah menentukan solusi baru. Tahap ini mencari solusi baru dari solusi yang telah diperoleh pada tahap sebelumnya dengan cara melakukan pencarian tetangga, yaitu memulai proses iterasi dari satu titik (solusi) ke solusi lain sampai kriteria terpenuhi (Gambar 9). Contoh penentuan solusi baru dapat dilihat pada Tabel 10. Setelah solusi awal ditemukan, yaitu pada nomor urut 1, dan kolom ruang sudah diperbarui, maka iterasi akan berjalan dengan mencari solusi terdekat berdasarkan nomor urut yang sudah digunakan pada proses sebelumnya. Solusi baru terdapat pada nomor dua dengan ruang perkuliahan di DC-204 yang diambil secara acak dari tabel ruang.

Proses keempat adalah evaluasi solusi-solusi alternatif dengan *tabu list*. Pada penelitian ini tabel yang digunakan sebagai *tabu list* adalah tabel *plotting* jadwal. Pada tabel *plotting* jadwal terdapat informasi berupa

**Tabel 10.** Solusi baru pembagian ruang

No	Dosen	Kelas	Mata kuliah	Hari	Jam	Ruang
1	SDA	S1SD-01-A	Bahasa Indonesia	SENIN	07: 00	IOT-103
2	IFA	S1IF-06-TI	Komputasi Awan	SENIN	09: 30	DC-204
3	AGZ	S1IF-07-N	Sistem Operasi	SENIN	13: 00	
4	PRD	S1IF-06-TI	Algoritme Pemrograman	SENIN	07: 30	
5	STS	S1SI-02-B	Kalkulus	SENIN	10: 00	

jadwal mengajar dosen, hari, dan jam yang diperoleh dari proses genetika sebagai generasi terbaik. Pada tabel jadwal ada tiga kolom kosong yaitu kolom gen ruang, ruang perkuliahan dan status. Ilustrasi tabel *plotting* jadwal dapat dilihat pada [Tabel 11](#).

Evaluasi dilakukan untuk melihat apakah kandidat solusi (solusi alternatif) tersebut sudah ada pada *tabu list*, yaitu dengan mengecek kolom *genruang*. Apabila solusi alternatif sudah ada pada *tabu list*, maka solusi tersebut tidak akan di evaluasi lagi. Pada proses iterasi ada parameter yang tidak boleh dilanggar, yaitu satu ruang perkuliahan tidak boleh digunakan untuk mengajar lebih dari satu mata kuliah pada hari dan jam yang sama. Apabila tidak ada pelanggaran, maka *genruang*, *idruang* dan *status* akan diperbarui dan kolom status akan diberi nilai 1, sehingga pada proses selanjutnya tidak akan dievaluasi lagi. Proses evaluasi solusi dapat dilihat pada [Gambar 10](#).

### III. HASIL DAN PEMBAHASAN

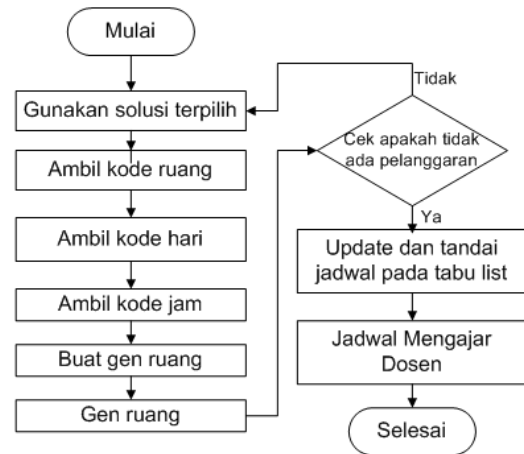
#### Hasil evaluasi GA-TS

Pada proses pembangkitan populasi awal, dilakukan pembuatan jadwal mengajar dosen menggunakan data yang berasal dari Fakultas Informatika dengan empat program studi. Masing-masing dosen memiliki kemungkinan ditugaskan mengajar lebih dari satu mata kuliah. Dosen yang sudah mendapatkan tugas mengajar akan ditempatkan di ruang kelas yang tersedia. Rincian jadwal mengajar dosen dapat dilihat pada [Tabel 12](#).

Pembangkitan populasi awal dengan menggunakan parameter pada [Tabel 1](#) memperoleh hasil berupa data populasi awal, yang terdiri dari kode dosen, kelas, dan kromosom seperti pada [Tabel 13](#). Parameter *krom1* sampai *krom10* adalah kromosom yang terbentuk sebagai

**Tabel 11.** *Tabu list* menggunakan tabel *plotting* jadwal

Kode Plotting	Kode Dosen	Id Mata Kuliah	Kode Prodi	Kelas	Kode Hari	Kode Jam	Gen Ruang	Kode Ruang	Status
204	STK	225	31	S1IF-05-S	3	50	56,3,50	56	1
205	YUS	4	32	S1SI-03-D	3	40	56,3,40	56	1
206	STK	263	31	S1IF-05-S	4	46			
207	CWA	49	33	S1SI-02-C	1	40			
208	ASI	257	32	S1IF-07-X	5	46			
209	AGZ	24	34	S1SI-03-C	2	41			



**Gambar 10.** Diagram alir evaluasi solusi alternatif

**Tabel 12.** Rincian jadwal mengajar dosen

Keterangan	Jumlah
Dosen	65
Mata Kuliah	93
Penugasan Dosen	265
Kelas	65

solusi awal dari proses genetika. Pada tiap kromosom terdapat angka-angka yang merepresentasikan jadwal mengajar dosen, seperti dinyatakan pada [Gambar 3](#). Waktu yang dibutuhkan pada proses pembangkitan populasi awal adalah 304 detik.

Hasil evaluasi nilai *fitness* dinyatakan pada [Tabel 14](#). Pada proses evaluasi nilai *fitness* awal masih ditemukan banyak pelanggaran dan belum ditemukan nilai *fitness* terbaik. Evaluasi nilai *fitness* dengan nilai tertinggi ada pada populasi ke-2 dan 10 yang disebabkan jumlah pelanggarannya paling sedikit. Pada evaluasi nilai *fitness*, semakin sedikit pelanggaran yang dilakukan, maka semakin tinggi nilai *fitness*-nya.

Pada proses *crossover*, proses berhenti pada iterasi atau populasi ke-190 seperti ditunjukkan pada [Tabel 15](#). Iterasi pada *crossover* berhenti pada generasi ke-190 dan masih ditemukan pelanggaran, baik pada *constraint* pertama maupun *constraint* kedua. Karena pada proses *crossover* masih ditemukan pelanggaran, maka proses mutasi otomatis dijalankan. Total waktu yang digunakan pada proses *crossover* dan mutasi adalah 243 detik.

Pada proses mutasi, iterasi berhenti pada generasi ke-244 karena sudah tidak ada pelanggaran yang

**Tabel 13.** Hasil populasi awal pada aplikasi penjadwalan mata kuliah

No	Kode Dosen	Kelas	Krom1	Krom2	Krom3	Krom4	Krom5	Krom6	Krom7	Krom8	Krom9	Krom10
1	ASI	S11F-06-X6	19,2,46	19,3,50	19,4,35	19,5,46	19,1,50	19,2,46	19,3,40	19,4,35	19,5,46	19,1,35
2	ASI	S11F-07-X2	19,3,40	19,4,35	19,5,46	19,1,35	19,2,39	19,3,50	19,4,35	19,5,38	19,1,35	19,2,46
3	ASI	S11F-07-X1	19,4,46	19,5,46	19,1,50	19,2,46	19,3,50	19,4,46	19,5,38	19,1,50	19,2,46	19,3,50
4	ASI	S11F-06-X7	19,5,38	19,1,35	19,2,46	19,3,40	19,4,46	19,5,38	19,1,35	19,2,39	19,3,50	19,4,46
5	IFA	S11F-06-TI	28,1,50	28,2,39	28,3,50	28,4,46	28,5,46	28,1,35	28,2,46	28,3,50	28,4,46	28,5,46
6	IFA	S11F-06-TI	28,2,39	28,3,50	28,4,46	28,5,46	28,1,35	28,2,46	28,3,50	28,4,35	28,5,46	28,1,50
7	IFA	S11F-06-TI	28,3,50	28,4,46	28,5,38	28,1,50	28,2,46	28,3,40	28,4,46	28,5,46	28,1,50	28,2,39
8	IFA	S11F-06-TI	28,4,46	28,5,38	28,1,35	28,2,39	28,3,40	28,4,46	28,5,46	28,1,35	28,2,46	28,3,50
9	FMW	S11F-07-O	33,5,34	33,1,46	33,2,51	33,3,46	33,4,40	33,5,51	33,1,46	33,2,51	33,3,36	33,4,40
10	FMW	S11F-07-M	33,1,46	33,2,51	33,3,36	33,4,40	33,5,51	33,1,46	33,2,51	33,3,36	33,4,51	33,5,34
...	...	...	...	...	...	...	...	...	...	...	...	...
265	NAF	S1DS-01-A	154,1,46	154,2,35	154,3,36	154,4,51	154,5,34	154,1,40	154,2,35	154,3,36	154,4,51	154,5,51

**Tabel 14.** Evaluasi nilai *fitness*

Populasi	Pelanggaran		Nilai <i>fitness</i>
	1	2	
10	8	40	0,020408
9	10	44	0,018182
8	12	50	0,015873
7	8	43	0,019231
6	12	52	0,015385
5	12	41	0,018519
4	12	44	0,017544
3	8	33	0,02381
2	8	40	0,020408
1	6	46	0,018868

**Tabel 16.** Evaluasi nilai *fitness* pada proses mutasi

Populasi	Induk		Pelanggaran		Nilai <i>fitness</i>
	1	2	1	2	
244	243	180	0	0	1
243	242	25	2	0	0,333333
242	241	179	2	2	0,2
241	230	134	2	2	0,2
240	239	8	4	2	0,142857
239	238	94	4	0	0,2
238	230	55	2	2	0,2
237	236	2	6	0	0,142857
236	235	137	4	0	0,2
...	...	...	...	...	...

**Tabel 15.** Evaluasi nilai *fitness* pada proses *crossover*

Populasi	Induk		Pelanggaran		Nilai <i>fitness</i>
	1	2	1	2	
190	3	188	10	39	0,02
189	3	187	10	44	0,018182
188	3	172	10	39	0,02
187	3	186	10	44	0,018182
186	3	177	10	44	0,018182
185	3	161	12	44	0,017544
184	3	173	12	35	0,020833
183	3	171	10	39	0,02
182	3	181	8	39	0,020833
181	3	178	8	39	0,020833
180	3	179	14	37	0,019231

**Tabel 17.** Generasi terbaik yang dihasilkan

No	Kode Dosen	Kelas	Hari	Jam	Gen
1	AAB	S11F-05-SC	SENIN	07:30	39,1,35
2	AAB	S1SE-02-A	JUMAT	09:00	39,5,38
3	AAB	S11F-08-E	SELASA	15:30	39,2,51
4	AAB	S11F-08-D	KAMIS	10:00	39,4,40
5	ABA	S11F-07-N	SELASA	13:00	156,2,46
6	ABA	S11F-07-O	RABU	10:00	156,3,40
7	ABA	S11F-07-P	KAMIS	07:30	156,4,35
8	ACW	S1SE-03-B	JUMAT	15:30	113,3,51
9	ACW	S1SE-02-A	RABU	15:00	113,3,50
10	ACW	S1SE-02-A	SENIN	13:00	113,1,46
...	...	...	...	...	...
265	YUS	S1SI-03-C	SELASA	09:30	79,2,39

ditemukan dan nilai *fitness* terbaik telah ditemukan. Hasil dari proses mutasi dapat dilihat pada [Tabel 16](#). Generasi terbaik ditemukan pada populasi ke-244 dengan nilai *fitness* 1 yang artinya tidak ada bentrok jadwal mengajar dosen, sedangkan solusi terburuk ada pada populasi ke-6 dengan nilai *fitness* 0,015385 yang artinya masih ditemukan bentrok jadwal mengajar dosen. Hasil populasi terbaik dinyatakan pada [Tabel 17](#).

Hasil pembagian ruang menggunakan algoritme TS telah memperoleh jadwal mengajar dosen dengan informasi lengkap, yang terdiri dari data Dosen, Kelas, Mata Kuliah, Hari, Ruang, Waktu Mulai, dan Waktu

Selesai perkuliahan. Pada proses pembagian ruang waktu yang dibutuhkan adalah 14 detik dan tidak ditemukan pelanggaran. Hasil akhir jadwal mengajar dosen dapat dilihat pada [Tabel 18](#).

### Pembahasan

Penelitian ini bertujuan untuk memberikan solusi terhadap pengembangan sistem informasi penjadwalan dalam [5] yang proses penjadwalannya masih manual menggunakan Microsoft Excel. Proses pembuatan jadwal menggunakan cara manual bisa membutuhkan

**Tabel 18.** Jadwal mengajar dosen

No	Dosen	Kelas	Mata Kuliah	Hari	Ruang	Mulai	Selesai
1	SDA - SHINTIA DA	S1SD-01-A	IN0001003 - BAHASA INDONESIA	SENIN	B407	7:30	10:00
2	IFA - IPAM FA	S1IF-06-TI3	IF6313503 - KOMPUTASI AWAN	SENIN	REK-205	7:30	10:00
3	AGZ - ANGGI ZFA	S1IF-07-N	IF6211403 - SISTEM OPERASI	SENIN	REK-206	15:00	17:30
4	PRD - PARADISE	S1IF-06-TI3	IF6312804 - DESAIN DAN ANALISIS ALGORITMA	SENIN	IOT-201	7:00	10:20
5	STS - S. THYA SFT	S1SI-02-B	FI6000302 - METODOLOGI PENELITIAN	SENIN	IOT-102	10:00	11:40
6	MYF - YOKA FTH	S1SI-04-A	IS6110302 - PENGANTAR SISTEM INFORMASI	SENIN	B304	10:00	11:40
7	MAM - M AFRIZAL AMR	S1IF-08-B	IF6110202 - ALGORITMA PEMROGRAMAN	SENIN	B503	10:00	11:40
8	HWU - HARI WUO	S1SI-03-B	IN0000902 - BAHASA INGGRIS UNTUK KOMUNIKASI DAN BISNIS	SENIN	IOT-103	10:00	11:40
9	GFA - GITA FFT	S1SE-01-A	SE6404322 - TUGAS AKHIR 1	SENIN	REK-205	10:00	11:40
10	ASI - AULIA SCH	S1IF-07-X1	IN0000703 - PANCASILA DAN KEWARGANEGARAAN	SENIN	B308	15:00	17:30
...	...	...	...	...	...	...	...
265	NA - NOVIAN APO	S1IF-06-TI2	IF6312703 - PEMROGRAMAN PERANGKAT BERGERAK	JUMAT	B408	9:00	11:30

**Tabel 19.** Hasil perbandingan metode AG dan AG-TS

Ujicoba	Jadwal	AG			AG-TS		
		Generasi	Waktu (detik)	Fitness	Generasi	Waktu (detik)	Fitness
1	132	200	190	0,02	164	123	1
2	132	200	209	0,018519	162	126	1
3	132	200	189	0,015873	162	121	1
4	132	200	196	0,018182	194	147	1
5	132	200	202	0,019231	167	132	1
6	132	200	200	0,018182	164	120	1
7	132	200	201	0,015152	161	122	1
8	132	200	192	0,023256	156	119	1
9	132	200	186	0,02	163	124	1
10	132	200	192	0,020833	176	136	1
	Rata-rata	200	196	0,0189228	167	127	1

waktu satu minggu bahkan sampai satu bulan karena banyak aspek, seperti jumlah mata kuliah yang diselenggarakan, jumlah ruangan yang terbatas, jumlah dosen dan jadwal dosen yang bersangkutan [7], dan mengandung risiko bentrok jadwal cukup tinggi [3].

Pada tahap uji coba dilakukan penyusunan jadwal menggunakan dua metode, yaitu hanya menggunakan AG, dan menggunakan AG-TS. Data yang digunakan pada tahap uji coba diambil dari program studi Teknik Informatika sejumlah 132 data, dengan parameter generasi (iterasi) sebanyak 200. Pengujian ini dilakukan untuk mengetahui metode yang lebih optimal. Dari sepuluh kali uji coba, kombinasi AG-TS menghasilkan solusi yang lebih baik dari hasil rata-rata jumlah generasi, nilai *fitness*, dan waktu proses (Tabel 19).

Lamanya waktu proses dan hasil nilai *fitness* AG disebabkan karena nilai gen pembentuk kromosom lebih bervariasi, yaitu (dosen, hari, jam, dan ruang). Nilai yang dievaluasi menjadi lebih banyak, yaitu keterkaitan dosen terhadap waktu (hari dan jam) dan keterkaitan ruang terhadap waktu. Hal ini sesuai dengan [20] yang menyatakan bahwa variasi nilai gen pembentuk

**Tabel 20.** Perbandingan AG-TS dengan [11]

Keterangan	Hasil kajian ini	Hasil dalam [11]
Generasi / iterasi	622	754
Waktu proses (detik)	336	70
Nilai fitness	1	1

kromosom mempengaruhi waktu proses, sedangkan pada metode AG-TS lebih sederhana, yaitu (dosen, hari, jam).

Perbandingan kinerja AG-TS dilakukan terhadap [11] yang juga menggunakan algoritme GA-TS untuk penjadwalan perkuliahan. Pengukuran kinerja kedua sistem dilakukan menggunakan data sebanyak 32 dengan maksimum iterasi 1000, mutasi 0,1, dan *crossover* 0,6. Hasil perbandingannya dinyatakan pada Tabel 20. Keduanya konsisten dengan [16] yang menunjukkan bahwa kombinasi AG-TS lebih optimal dengan *fitness* bernilai 1 jika dibandingkan dengan AG.

Namun, kajian [11] menunjukkan hasil yang lebih baik dari segi waktu proses, sedangkan pada kajian ini memberikan hasil lebih baik dalam jumlah iterasi. Lama



proses dalam kedua sistem ditentukan oleh jumlah komponen yang diproses. Jumlah dosen, mata kuliah, dan kelas dalam [11] lebih sedikit, yaitu 29 mata kuliah, 32 dosen, dan 24 kelas, dibandingkan dalam kajian ini sebanyak 93 mata kuliah, 65 dosen, dan 65 kelas. Perbedaan jumlah iterasi disebabkan karena metode yang diterapkan, yaitu pada kombinasi AG-TS. Dalam kajian [11], TS digunakan sebagai penapis di dalam proses AG, yaitu pada proses pembentukan kromosom, sedangkan pada kajian ini TS diimplementasikan setelah mendapat generasi terbaik hasil AG. Kajian ini menghasilkan jumlah iterasi yang lebih sedikit.

#### IV. KESIMPULAN

Kombinasi AG-TS mampu memproses data untuk penjadwalan mata kuliah dengan lebih optimal dengan rata-rata *fitness* bernilai 1 dibandingkan metode yang hanya menggunakan AG dengan *fitness* rata-rata sebesar 0,0189228. Implementasi TS setelah memperoleh generasi terbaik hasil AG membuat jumlah iterasi lebih sedikit. Pada sistem yang dikembangkan masih terdapat kelemahan, yaitu sistem tidak berjalan maksimal pada uji coba menggunakan data sebanyak 550 kelas. Penelitian lanjutan dapat dilakukan untuk meningkatkan performa sistem dengan menggunakan AG atau TS yang dikombinasikan dengan algoritme lain.

#### MATERI PENDUKUNG

Naskah ini mempunyai berkas pendukung, berupa data hasil analisis, yang dapat ditemukan versi daringnya di doi: [10.14710/jtsiskom.2021.14137](https://doi.org/10.14710/jtsiskom.2021.14137).

#### DAFTAR PUSTAKA

- [1] A. Akgunduz and Y. Zeng, "Innovative course scheduling and curriculum design," in *Canadian Engineering Education Association Conference*, Dalhousie, Canada, Jun. 2016, pp. 1-5.
- [2] F. N. Afandi and M. Yulianis, "Implementasi genetic algorithms untuk penjadwalan mata kuliah berbasis website," *Explore Jurnal Sistem Informasi dan Telematika*, vol. 9, no. 1, pp. 45-52, 2018. doi: [10.36448/jsit.v9i1.1031](https://doi.org/10.36448/jsit.v9i1.1031)
- [3] A. P. Rahadi, "Penjadwalan mata kuliah menggunakan pewarnaan graf algoritma largest first," *Jurnal Padagogik*, vol. 2, no. 1, pp. 1-13, 2019. doi: [10.35974/jpd.v2i1.1067](https://doi.org/10.35974/jpd.v2i1.1067)
- [4] T. Sunarni, "Optimasi penjadwalan mata kuliah menggunakan pewarnaan graf," in *Seminar Nasional Teknik Industri*, Malang, Indonesia, Oct. 2017, pp. 48-53.
- [5] I. A. Ramadhani, "Pengembangan sistem informasi penjadwalan mata kuliah berbasis web di fakultas Teknik Universitas Negeri Makassar," *Jurnal Pendidikan*, vol. 6, no. 2, pp. 1-15, 2018. doi: [10.36232/pendidikan.v6i2.36](https://doi.org/10.36232/pendidikan.v6i2.36)
- [6] N. M. H. Robbi and N. Nurochman, "Implementasi algoritma genetika untuk penjadwalan instruktur training ICT UIN Sunan Kalijaga," *Jurnal Informatika Sunan Kalijaga (JISKa)*, vol. 1, no. 3, p. 123-132, 2017. doi: [10.14421/jiska.2017.13-04](https://doi.org/10.14421/jiska.2017.13-04)
- [7] A. Josi, "Implementasi algoritma genetika pada aplikasi penjadwalan perkuliahan berbasis web dengan mengadopsi model waterfall (Studi kasus: STMIK Prabumulih)," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 2, no. 2, pp. 77-83, 2017.
- [8] R. M. Puspita, A. Arini, and S. U. Masruroh, "Pengembangan aplikasi penjadwalan kegiatan pelatihan teknologi informasi dan komunikasi dengan algoritma genetika (Studi kasus: BPRTIK)," *JOIN Jurnal Online Informatika*, vol. 1, no. 2, pp. 76-81, 2016. doi: [10.15575/join.v1i2.43](https://doi.org/10.15575/join.v1i2.43)
- [9] D. Kristiadi and R. Hartanto, "Genetic algorithm for lecturing schedule optimization (case study: University of Boyolali)," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 13, no. 1, pp. 83-94, 2019. doi: [10.22146/ijccs.43038](https://doi.org/10.22146/ijccs.43038)
- [10] A. S. Laswi, "Perbandingan algoritma fitness of spring dan algoritma tabu search pada kasus penjadwalan perkuliahan," *ILKOM Jurnal Ilmiah*, vol. 12, no. 1, pp. 39-46, 2020. doi: [10.33096/ilkom.v12i1.522.39-46](https://doi.org/10.33096/ilkom.v12i1.522.39-46)
- [11] R. Rusianah, M. A. Muslim, and S. H. Pramono, "Implementasi algoritma genetika-tabu search dalam optimasi penjadwalan perkuliahan," *Jurnal EECCIS*, vol. 10, no. 2, pp. 45-50, 2016.
- [12] Y. P. Sumihar and A. Musdholifah, "Kombinasi algoritme genetika dan tabu list pada kasus penjadwalan ujian," *Berkala MIPA*, vol. 25, no. 3, pp. 300-311, 2018.
- [13] V. Kinasya, "Application of (genetic – tabu search) algorithms for subsequent lease schedule," *IOP Conference Series: Materials Science and Engineering*, vol. 407, 012124, pp. 1-4, 2018. doi: [10.1088/1757-899X/407/1/012124](https://doi.org/10.1088/1757-899X/407/1/012124)
- [14] S. E. Ramadhania and S. Rani, "Implementasi kombinasi algoritma genetika dan tabu search untuk penyelesaian travelling salesman problem," *Automata*, vol. 2, no. 1, pp. 1-8, 2021.
- [15] N. H. Hari, F. P. E. Putra, and H. Hamdlani, "Optimasi penjadwalan menggunakan metode algoritma genetika di Sekolah Menengah Kejuruan Annuqayah - Sumenep," *Query: Jurnal Sistem Informasi*, vol. 2, no. 2, pp. 66-74, 2018.
- [16] N. I. Kurniati, A. Rahmatulloh, and D. Rahmawati, "Perbandingan performa algoritma koloni semut dengan algoritma genetika – tabu search dalam penjadwalan kuliah," *CESS (Journal of Computer Engineering System and Science)*, vol. 4, no. 1, pp. 17-23, 2019. doi: [10.24114/cess.v4i1.11387](https://doi.org/10.24114/cess.v4i1.11387)
- [17] E. Desiana, "Performance algoritma genetika (ga) pada penjadwalan mata pelajaran," *InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan)*, vol. 1, no. 1, pp. 56-60, 2016. doi: [10.30743/infotekjar.v1i1.42](https://doi.org/10.30743/infotekjar.v1i1.42)
- [18] A. Z. Alfaraby, A. M. Hilda, and M. Kamayani, "Penjadwalan hafalan Alquran dengan algoritma

- genetika," in *Seminar Nasional TEKNOKA*, Jakarta, Indonesia, Nov. 2018, pp. 35-41. doi: [10.22236/teknoka.v3i0.2898](https://doi.org/10.22236/teknoka.v3i0.2898)
- [19] I. Ivan, S. Raphael, and H. Agung, "Aplikasi penjadwalan mata pelajaran di SMAN 31 menggunakan algoritma genetika berbasis web," *Jurnal SIMETRIS*, vol. 9, no. 1, pp. 641-656, 2018. doi: [10.24176/simet.v9i1.2010](https://doi.org/10.24176/simet.v9i1.2010)
- [20] D. Setiawan, R. N. Putri, and R. Suryanita, "Implementasi algoritma genetika untuk prediksi penyakit autisme," *RABIT (Jurnal Teknologi dan Sistem Informasi Univrab)*, vol. 4, no. 1, pp. 8-16, 2019. doi: [10.36341/rabit.v4i1.595](https://doi.org/10.36341/rabit.v4i1.595)



©2021. This open-access article is distributed under the terms and conditions of the [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).