

## **Pengembangan Program Penjadwalan Matakuliah Menggunakan Algoritma Genetika Teroptimasi Mutasi Differential Evolution**

Ali Muhammad<sup>1</sup>, Angge Firizkiansah<sup>2</sup>, Dita Setiawan<sup>3</sup>  
<sup>1,2,3</sup>Universitas Sains Indonesia, Kabupaten Bekasi

E-mail:

ali.muhammad@lecturer.sains.ac.id<sup>1</sup>(korespondensi), ange.firizkiansah@lecturer.sains.ac.id<sup>2</sup>,  
dita.setiawan@lecturer.sains.ac.id<sup>3</sup>

### **Abstract**

*Course scheduling is a complex problem that requires an intelligent computational approach to produce optimal solutions in an efficient time. This problem becomes increasingly challenging with many constraints such as lecture availability, room capacity, and limited lecture time. The purpose of this study is to develop an application to solve the scheduling problem without any time and space collisions and to place the lecturer's schedule according to the availability of the lecturer's teaching time. The practical benefits of this study are to improve the quality of academic information systems, especially in the adaptive and flexible automatic scheduling process. In addition, the course scheduling and optimization process can also be adjusted for various similar cases. Such as, industrial work scheduling and cloud computing systems. This study uses a Genetic Algorithm optimized by Differential Evolution Treatment. Differential Evolution treatment optimization aims to improve the performance of finding optimal solutions. The results of the study show that the error rate of the resulting course scheduling is 0%. Computational time need for testing using 22 lecturers, 22 rooms, 24 classes data with the same number of iterations of 1000 times was completed in 24 minutes 25 seconds.*

**Keywords:** *course; genetic algorithm; mutation differential evolution; scheduling*

### **Abstrak**

Penjadwalan mata kuliah merupakan permasalahan kompleks yang memerlukan pendekatan komputasi cerdas untuk menghasilkan solusi optimal dalam waktu yang efisien. Permasalahan ini menjadi semakin menantang dengan banyaknya kendala seperti ketersediaan dosen, kapasitas ruang, serta waktu perkuliahan yang terbatas. Tujuan penelitian ini adalah untuk mengembangkan program untuk menyelesaikan masalah penjadwalan tanpa adanya bentrok waktu dan ruang serta menempatkan jadwal dosen sesuai dengan ketersediaan waktu dosen mengajar. Manfaat praktis penelitian ini adalah meningkatkan kualitas sistem informasi akademik, khususnya pada proses penjadwalan otomatis yang adaptif dan fleksibel. Selain itu, proses penjadwalan dan optimasi matakuliah juga dapat diadaptasi untuk berbagai kasus serupa seperti penjadwalan kerja industri dan sistem *cloud computing*. Penelitian ini menggunakan Algoritma Genetika teroptimasi mutasi *Differential Evolution*. Optimasi mutasi *Differential Evolution* bertujuan untuk meningkatkan performa pencarian solusi optimal. Hasil penelitian menunjukkan bahwa tingkat kesalahan penjadwalan matakuliah yang dihasilkan sebesar 0%. Pengujian waktu komputasi menggunakan data 22 dosen, 22 ruang, 24 kelas dengan jumlah iterasi yang sama sebanyak 1000X memerlukan waktu komputasi sebesar 24 menit 25 detik.

**Kata kunci:** algoritma genetika; mata kuliah; mutasi differential evolution; penjadwalan

## 1. PENDAHULUAN

Penjadwalan perkuliahan merupakan salah satu komponen penting dalam manajemen akademik di perguruan tinggi. Permasalahan penjadwalan mata kuliah juga menjadi tantangan klasik dalam dunia pendidikan tinggi yang hingga kini masih menjadi perhatian serius. Kompleksitas permasalahan ini melibatkan banyak variabel dan kendala, seperti keterbatasan ruang kelas, ketersediaan dosen, konflik waktu antar mata kuliah, serta kepentingan administratif institusi [1, 2, 3, 4, 5, 6]. Seiring perkembangan teknologi informasi dan meningkatnya kebutuhan efisiensi manajemen akademik, pendekatan komputasi cerdas melalui algoritma metaheuristik mulai banyak diterapkan. Algoritma Genetika menjadi salah satu metode yang paling populer karena kemampuannya dalam menjelajahi ruang solusi yang luas dan menghasilkan jadwal dengan tingkat konflik rendah [1, 4, 7, 8, 9].

Kompleksitas struktur kurikulum dan jumlah peserta didik, kebutuhan akan sistem penjadwalan otomatis yang efisien dan adaptif menjadi semakin mendesak. Fenomena saat ini menunjukkan bahwa institusi pendidikan mulai beralih ke sistem berbasis kecerdasan buatan, khususnya metode optimasi berbasis algoritma evolusioner seperti Algoritma Genetika, untuk menangani permasalahan penjadwalan secara lebih optimal. Namun demikian, sejumlah penelitian menunjukkan bahwa Algoritma Genetika memiliki keterbatasan signifikan, di antaranya adalah kecenderungan terhadap konvergensi prematur, efisiensi waktu yang bervariasi, serta kurangnya fleksibilitas dalam menangani permasalahan skala besar atau kompleks [2, 4, 10, 11, 12]. Untuk mengatasi kelemahan tersebut, telah dilakukan beberapa pendekatan seperti optimasi algoritma genetika dengan *Tabu Search* [2, 15], *Artificial Bee Colony* [14], dan metode lain seperti *Simulated Annealing* [11] maupun *Spider Monkey Optimization* [15,16], serta optimasi mutasi *Differential Evolution* secara independen [17]. Meskipun demikian, sebagian besar pendekatan tersebut masih terbatas pada integrasi parsial atau hanya melakukan perbandingan per-

forma tanpa melakukan optimasi mutasi *Differential Evolution* kedalam Algoritma Genetika secara struktural [12, 17].

Hasil penjadwalan diharapkan tidak hanya optimal dari segi hasil, tetapi juga efisien dari sisi waktu komputasi dan adaptif terhadap berbagai kendala akademik. Oleh karena itu, penelitian ini mengembangkan sebuah pendekatan optimasi yang mengintegrasikan mekanisme mutasi dari *Differential Evolution* ke dalam proses evolusi Algoritma Genetika. Integrasi ini bertujuan untuk meningkatkan keragaman populasi, menghindari stagnasi solusi, serta menghasilkan jadwal yang lebih optimal dan stabil dalam berbagai skenario [10, 12, 15, 17, 18].

Penelitian terstruktur yang secara eksplisit menggabungkan teknik mutasi *Differential Evolution* kedalam kerangka Algoritma Genetika bertujuan untuk menyelesaikan masalah penjadwalan mata kuliah, meskipun penggabungan Algoritma Genetika dengan teknik lain telah banyak dikaji dalam berbagai konteks, seperti *Job Shop Scheduling*, *Travelling Salesman Problem*, dan sistem *cloud computing* [16, 18, 19, 20]. Selain itu, sebagian besar penelitian sebelumnya lebih fokus pada implementasi aplikasi atau studi kasus tanpa eksplorasi algoritmik mendalam [3, 8, 9].

## 2. METODE

Penelitian ini menggunakan metode *Research and Development* dengan model pengembangan *ADDIE*. Adapun pengembangan perangkat lunak pada penelitian ini menggunakan metode *Devops*. Secara umum untuk menyelesaikan permasalahan penjadwalan matakuliah yang kompleks, penelitian ini menggunakan Algoritma Genetika teroptimasi mutasi *Differential Evolution*. Terakhir, penelitian ini menggunakan metode pengujian berdasarkan parameter genetika menggunakan data simulasi pada setiap tahapan proses pada metode Algoritma genetika.

### 2.1. Data Penelitian

Penelitian ini menggunakan data Kelas, data Ruang, data Dosen yang diperoleh dari Biro

Administrasi dan Akademik Universitas Sains Indonesia Tahun 2024 .

SESI	HARI/WAKTU	PROGRAM STUDI						KET
		KODE MK	MATA KULIAH	SKS	KODE KLS	KET	RUANG	
1	SENIN 17.00 - 18.40 19.00 - 21.30	USI101	Pendidikan Agama	2	11ECO	TH-SJHK,DG		et
		FIC101	Pengantar Teknologi Informasi	3	11FA	TH-SI	C-102 (25)	TM
2	SELASA 19.00 - 21.30	FIC102	Logika Informatika	3	12FA	TH-SI	C-102 (25)	TM
3	RABU 19.00 - 21.30	FIC103	Kalkulus 1	3	13FA	TH-SI	C-102 (25)	TM
4	KAMIS 19.00 - 21.30	USI104	Kewirausahaan	3	14FCO	TH-SJHK,DG		et
5	JUMAT 17.00 - 18.40 19.00 - 21.30	USI102	Pancasila	2	15ECO	TH-SJHK,DG		et
		USI103	Bahasa Inggris I	3	15FCO	TH-SJHK,DG		et

Gambar 1. Contoh Tabel Penjadwalan Kuliah

2.2. Metode Research and Development

Metode penelitian Research and Development telah digunakan pada penelitian sebelumnya [21]. Metode ini difokuskan pada model pengembangan penelitian berbasis ADDIE. Metode penelitian ini dipilih karena dapat menyelesaikan penelitian dengan cepat melalui pendekatan tersistematis. Adapun langkah penyelesaian masalah menggunakan metode ADDIE diantaranya adalah : Analisis, Desain, Pembuatan, Penerapan, dan Evaluasi. Secara umum alur metode ADDIE dijabarkan pada gambar 2 berikut.

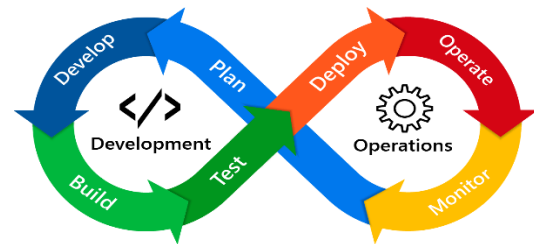


Gambar 2. Tahapan Penelitian Menggunakan Metode ADDIE

2.3. Metode DevOps

Metode devops telah digunakan pada penelitian pengembangan perangkat lunak sebelumnya [22]. Metode pengembangan DevOps digunakan agar proses pembuatan perangkat lunak dapat berjalan secara cepat dan

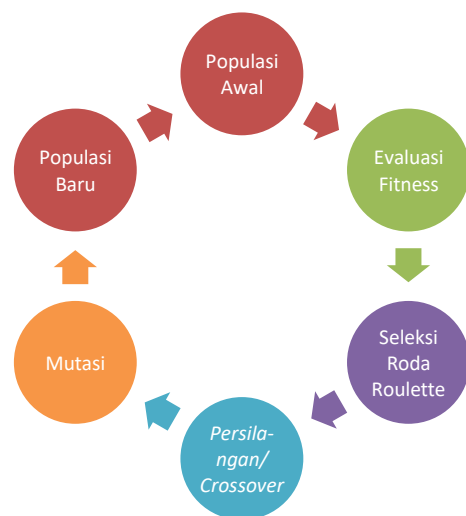
tepat. Adapun tahapan dari metode devops ini diantaranya adalah: Perencanaan, Pembuatan (*Develop*), Pengembangan (*Build*), Tahap Pengujian (*Test*), Penerapan (*Deploy*), Operasi (*Operate*), dan Pengawasan (*Monitor*). Keseluruhan proses pada metode *DevOps* dijabarkan pada gambar 3.



Gambar 3. Metode Pengembangan Perangkat Lunak Model DevOps

2.4. Metode Algoritma Teroptimasi Mutasi Differential Evolution

Penelitian sebelumnya telah menggunakan beberapa metode klasifikasi [23,24]. Penelitian ini menggunakan metode Algoritma Teroptimasi Mutasi *Differential Evolution* untuk menyelesaikan permasalahan penjadwalan yang kompleks. Dimana populasi dilakukan evaluasi *fitness*, seleksi roda *roulette*, persilangan/*crossover*, mutasi, dan menghasilkan populasi baru sampai target penelitian tercapai. Alur penyelesaian masalah metode algoritma genetika dijabarkan pada gambar 4.



Gambar 4. Proses Algoritma Genetika

$$f = \frac{1}{(h+a)^r} \tag{1}$$

$f$  pada formula *fitness* diatas merupakan fungsi *fitness*, sedangkan  $h$  adalah fungsi yang dimaksimasi atau diminimasi, sedangkan  $a$  adalah jumlah bilangan yang dianggap kecil dan disesuaikan dengan masalah yang akan diselesaikan.

Mutasi yang digunakan adalah mutasi *Differential Evolution*. Mutasi *Differential Evolution* yaitu membuat individu baru dari individu sebelumnya dengan menggunakan rumus yang berbeda. Proses mutasi pada Algoritma Genetika biasanya dilakukan secara acak. Sedangkan pada mutasi *Differential Evolution* digunakanlah persamaan  $V$  yang merupakan mutasi baru, sedangkan  $F$ , merupakan faktor skala sebesar 0,5 (nilai tersebut sudah dimasukkan pada proses pengkodean) dan  $X_{r1}$ ,  $X_{r2}$ ,  $X_{r3}$  merupakan kromosom yang sedang dimutasi. Secara sistematis rumus mutasi *Differential Evolution* dituliskan pada formula 2 berikut:

$$V = X_{r1} + F \times (X_{r2} - X_{r3}) \quad (1)$$

### 2.5. Pengujian Aplikasi

Metode pengujian perangkat lunak didasarkan pada keberfungsian metode Algoritma Genetika dalam membangun penjadwalan matakuliah tanpa adanya bentrok antara dosen, ruang maupun kelas. Secara umum metode pengujian ini terbagi menjadi Pengujian *Fitness* dengan Persyaratan Populasi, *Crossover*, Mutasi. Selain daripada itu, pengujian lainnya berupa pengujian *fitness* dengan kelompok, dosen, dan ruang yang berbeda.

Pengujian perangkat lunak lainnya meliputi pengujian banyaknya iterasi yang diperlukan dalam menyelesaikan masalah. Terakhir, pengujian waktu komputasi digunakan untuk mengetahui banyaknya waktu yang diperlukan komputer dalam menyelesaikan masalah.

### 3. HASIL DAN PEMBAHASAN

Pengembangan program penjadwalan matakuliah menggunakan algoritma genetika teroptimasi mutasi *Differential Evolution* diterapkan pada lingkungan bahasa pemrograman java.



**Gambar 5.** Tampilan Awal Program Penjadwalan Matakuliah

Pengujian penelitian ini didasarkan pada pengujian parameter genetika menggunakan data simulasi yang dijelaskan pada tabel 1 berikut.

**Tabel 1.** Data Simulasi Pengujian

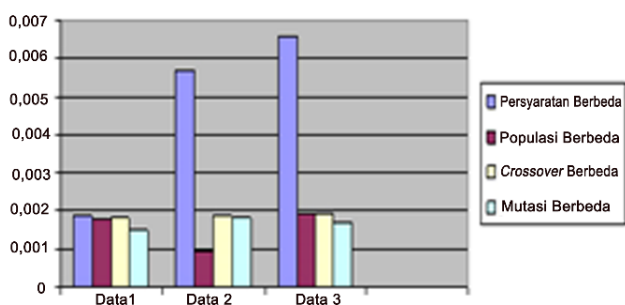
Pengujian	Jumlah Data 1	Jumlah Data 2	Jumlah Data 3
Persyaratan	2	4	9
Populasi	5	30	50
Crossover	0.001	0.01	1
Mutasi	0.001	0.01	1
Kelas	6	12	24
Dosen	14	18	22
Ruang	14	18	22

Data 1-3 merupakan data simulasi hasil pengujian perangkat lunak yang diambil dari rata-rata pengujian persyaratan, populasi, *crossover*, mutasi, kelas, dosen dan ruang.

Pengujian program penjadwalan matakuliah menggunakan algoritma genetika teroptimasi mutasi *differential evolution* meliputi :

#### 3.1. Pengujian Nilai *Fitness* terhadap Persyaratan Populasi, *Crossover* dan Mutasi Yang Berbeda

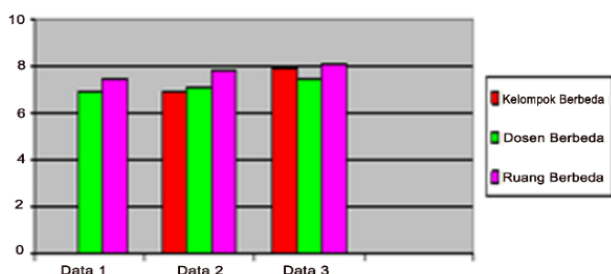
Pengujian nilai *fitness* dengan persyaratan populasi, *crossover*, dan mutasi yang berbeda menunjukkan hasil bahwa semakin besar persyaratan yang harus dipenuhi, semakin tinggi nilai *fitness* yang dihasilkan dan hampir mencapai 0,0065. Hasil pengujian nilai *fitness* dapat dilihat pada gambar 6.



**Gambar 6.** Hasil Pengujian Nilai *Fitness* dengan Persyaratan, Populasi, *Crossover*, dan Mutasi yang Berbeda

### 3.2. Pengujian Nilai *Fitness* terhadap Kelas, Dosen dan Ruang Yang Berbeda.

Hasil pengujian nilai *fitness* dengan kelas, dosen, dan ruang yang berbeda menunjukkan hasil bahwa nilai *fitness* naik dan mempunyai selisih yang sedikit diantara data 1, data 2, dan data 3. Hal ini menunjukkan bahwa bentrok antara kelompok/kelas yang berbeda, ruang yang berbeda dan dosen yang berbeda cenderung tidak pernah terjadi. Masil pengujian nilai *fitness* terhadap kelas, dosen dan ruang yang berbeda, dapat dilihat pada Gambar 7.

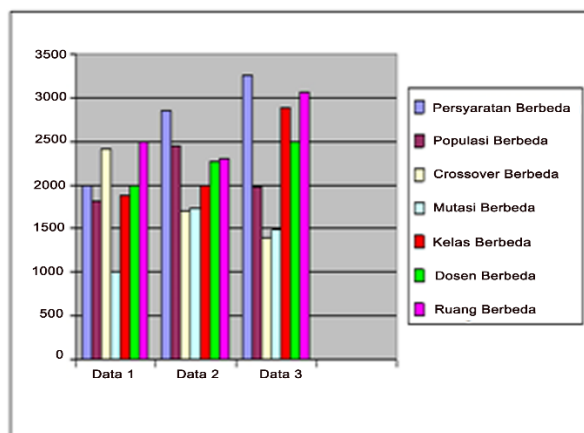


**Gambar 7.** Hasil Pengujian Nilai *Fitness* terhadap Kelas, Dosen dan Ruang Yang Berbeda

### 3.3. Pengujian Jumlah Generasi Dalam Menemukan Solusi

Hasil pengujian jumlah generasi dalam menemukan solusi terlihat bahwa semakin besar persyaratan yang berbeda, maka akan memerlukan jumlah generasi yang semakin banyak pula. Hal ini dikarenakan, semakin banyak persyaratan yang berbeda, algoritma genetika juga melakukan generasi dengan berbagai tingkat kemungkinan juga. Gambar 8 menunjukkan bahwa hasil generasi untuk persyaratan berbeda mencapai 3.500

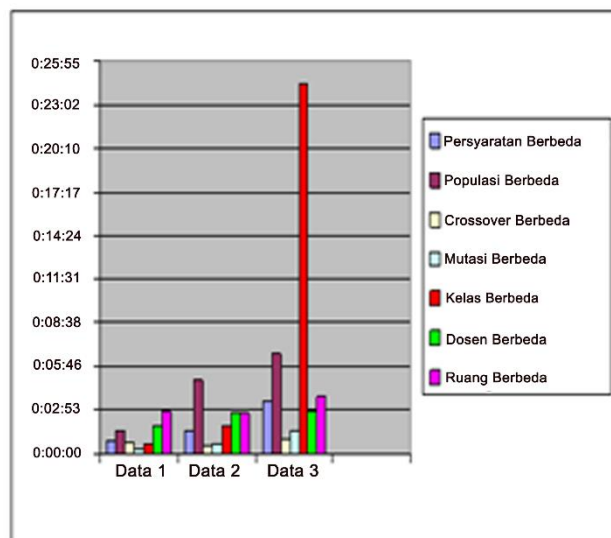
kemungkinan yang dilakukan oleh program untuk menemukan solusi optimal.



**Gambar 8.** Hasil Pengujian Jumlah Generasi Dalam Menemukan Solusi

### 3.4. Pengujian Waktu Komputasi

Hasil pengujian waktu komputasi menunjukkan bahwa, semakin banyak kelas yang berbeda, akan memerlukan waktu komputasi yang lama dibandingkan parameter persyaratan yang berbeda, populasi berbeda, *crossover* berbeda, mutasi berbeda, dosen berbeda, maupun ruang yang berbeda. Gambar 9 menunjukkan hasil pengujian menggunakan kelas berjumlah 24 kelas, memerlukan waktu komputasi sebesar 24 menit 25 detik.



**Gambar 9.** Hasil Pengujian Waktu Komputasi

#### 4. KESIMPULAN

Hasil penelitian pengembangan program penjadwalan menggunakan algoritma Genetika teroptimasi mutasi *Differential Evolution* dapat disimpulkan bahwa semakin banyak jumlah persyaratan yang harus dipenuhi maka akan semakin besar juga jumlah generasi dan semakin lama waktu yang dibutuhkan. Yang kedua, semakin banyak kelas berbeda, maka waktu komputasi akan semakin besar dan lama. Terakhir, didapat parameter genetika dengan populasi sebanyak 5, peluang *crossover* sebesar 0,01 dan mutasi sebesar 0.001 untuk menghasilkan waktu komputasi yang optimal.

#### 5. DAFTAR PUSTAKA

- [1] Nasien, D., & Andi, A. (2022). Optimization of Genetic Algorithm in Courses Scheduling. *IT Journal Research and Development*, 6(2), 151–161. <https://doi.org/10.25299/itjrd.2022.7896>
- [2] 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, Jun. 2021. <https://doi.org/10.14710/jtsiskom.2021.14137>
- [3] P. A. Rizki, Y. Hendriyani, D. Novaliendry, dan K. Budayawan, "Rancang Bangun Aplikasi Penjadwalan Mata Pelajaran SMK Muhammadiyah 1 Pekanbaru Berbasis Web Menggunakan Algoritma Genetika," *Fakultas Ilmu Pendidikan Universitas Pahlawan Tuanku Tambusai*, vol. 7, no. 3, hlm. 24787–24797, 2023, doi: <https://doi.org/10.31004/jptam.v7i3.10541>.
- [4] F. Mone dan J. E. Simarmata, "Aplikasi Algoritma Genetika Dalam Penjadwalan Mata Kuliah," *BAREKENG: J. Il. Mat. & Ter.*, vol. 15, no. 4, hlm. 615–628, Des 2021, doi: [10.30598/barekengvol15iss4pp615-628](https://doi.org/10.30598/barekengvol15iss4pp615-628)
- [5] Eka Yulia Sari, Dina Yulina H, dan Titik Rahmawati, "Pemodelan Sistem Optimasi Penjadwalan Mata Kuliah Dengan Algoritma Genetika," *teknimedia*, vol. 4, no. 1, hlm. 70–78, Jun 2023, doi: [10.46764/teknimedia.v4i1.97](https://doi.org/10.46764/teknimedia.v4i1.97).
- [6] L. A. Pangestu, S. H. Suryawan, dan A. J. Latipah, "Penerapan Algoritma Genetika Dalam Penjadwalan Mata Pelajaran," *Jurnal Penelitian Teknik Informatika, Manajemen Informatika dan Sistem Informasi*, vol. 10, no. 2, hlm. 194–205, Okt 2023, doi: [10.31294/inf.v10i2.16701](https://doi.org/10.31294/inf.v10i2.16701).
- [7] M. Furqan, R. Ananda, dan Armansyah, "Algoritma Genetika Untuk Perancangan Aplikasi Penjadwalan Mata Pelajaran," *JSAKTI (Jurnal Sains Komputer & Informatika)*, vol. 6, no. 2, hlm. 591–600, 2022, doi: <http://dx.doi.org/10.30645/jsakti.v6i2.476>.
- [8] A. Andriyadi, D. Yuliawati, dan S. Saleh, "Penerapan Algoritma Genetika Untuk Penjadwalan Sidang dan Seminar IIB Darmajaya," *Seminar Nasional Hasil Penelitian dan Pengabdian Masyarakat 2022*, vol. 1, hlm. 22–31, 2022
- [9] P. Puspitasari dan M. A. I. Pakereng, "Implementasi Algoritma Genetika Untuk Penjadwalan Sekolah (Studi Kasus: SMP Negeri 2 Wonosegoro)," *Jurnal Sains Komputer & Informatika (J-SAKTI)*, vol. 7, no. 1, hlm. 369–382, 2023, doi: <http://dx.doi.org/10.30645/jsakti.v7i1.600>.
- [10] Xiong Luo, Qian Qian, Yun Fa Fu, "Improved Genetic Algorithm for Solving Flexible Job Shop Scheduling Problem", *Proceedings of the 3rd International Conference on Mechatronics and Intelligent Robotics (ICMIR-2019)* vol. 166, pages 480-485, 2020, doi : <https://doi.org/10.1016/j.procs.2020.02.061>.
- [11] X. Liang and Z. Du, "Genetic Algorithm with Simulated Annealing for Resolving Job Shop Scheduling Problem," in *2020 IEEE 8th International Conference on Computer Science and Network Technology, ICCSNT 2020*, 2020, pp. 64–68, doi: [10.1109/ICCSNT50940.2020.9305010](https://doi.org/10.1109/ICCSNT50940.2020.9305010).
- [12] M. K. Fajarlestari dan I. B. Suban, "Kombinasi Crossover dan Mutasi Terbaik pada Algoritma Genetika dalam

- Penjadwalan Mata Kuliah,” *tc*, vol. 22, no. 4, hlm. 843–853, Nov 2023, doi: 10.33633/tc.v22i4.9298.
- [13] 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
- [14] S. N. Sari, R. Kaban, A. Khaliq, dan A. Andari, "Sistem Penjadwalan Mata Pelajaran Sekolah Menggunakan Metode Hybrid Artificial Bee Colony (HABC)," *JNASTEK*, vol. 2, no. 1, hlm. 20–32, Feb 2022, doi: 10.61306/jnastek.v2i1.21
- [15] M. A. H. Akhand, S. I. Ayon, S. A. Shahriyar, N. Siddique, and H. Adeli, "Discrete Spider Monkey Optimization for Travelling Salesman Problem," *Appl. Soft Comput. J.*, vol. 86, Jan. 2020, doi: 10.1016/j.asoc.2019.105887.
- [16] W. Firgiawan, A. Gunawan, S. Cokrowibowo, and A. Irianti, "Performance Comparison of Spider Monkey Optimization and Genetic Algorithm for Traveling Salesman Problem," in *3rd International Conference on Electronics Representation and Algorithm (ICERA)*, 2021, pp. 191–195, [Online]. Available: <http://elib.zib.de/pub/mp->
- [17] N. Lukman, M. Irfan, A. Nugraha, dan J. Jumadi, "Comparison Between Genetic Algorithm with Differential Evolution in Study Scheduling," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 1098, no. 3, hlm. 032082, Mar 2021, doi: 10.1088/1757-899X/1098/3/-032082
- [18] A. Yusron Mubarak dan U. Chotijah, "Penerapan Algoritma Genetika Untuk Mencari Optimasi Kombinasi Jalur Terpendek Dalam Kasus Travelling Salesman Problem," *j. teknologi terpadu*, vol. 7, no. 2, hlm. 77–82, 2021, doi: 10.54914/jtt.v7i2.424.
- [19] Peng, Zhihao, Pirozmand, Poria, Motevalli, Masoumeh, Esmaeili, Ali, Genetic Algorithm-Based Task Scheduling in Cloud Computing Using MapReduce Framework, *Mathematical Problems in Engineering*, 2022, 4290382, 11 pages, 2022.  
<https://doi.org/10.1155/2022/4290382>
- [20] S. Xinyi, W. Aimin, G. Yan, and Y. Jieran, "Job Shop Scheduling Problem with Job Sizes and Inventories," in *Proceedings of 2020 IEEE 11th International Conference on Mechanical and Intelligent Manufacturing Technologies, ICMIMT 2020*, 2020, pp. 202–206, doi: 10.1109/ICMIMT49010.-2020.9041174.
- [21] Winda N, Muhammad A, "Pengembangan Parsing PCPATR sebagai Preservasi Bahasa dan Sastra Banjar", *Jurnal Onoma: Pendidikan, Bahasa, dan Sastra*, vol. 9, no. 2, 2023.
- [22] Muhammad A, Widyastuti N, "Pengembangan Aplikasi Part-of-Speech Tagger Bahasa Banjar Menggunakan Metode Pengembangan DevOps", *JIKOMTI: Jurnal Ilmiah Ilmu Komputer dan Teknologi*, vol. 1, no.1, 2024.
- [23] Setiawan D, Muhammad A, Firizkiansah A, "Pengklasifikasian Dokumen Teks Bahasa Indonesia berbasis Vektor Space Model dengan menggunakan Metode k-Nearest Neighbor (k-NN) dan Euclidean Distance", *JIKOMTI: Jurnal Ilmiah Ilmu Komputer dan Teknologi*, vol. 1, no.1, 2024.
- [24] Firizkiansah A, Muhammad A, Setiawan D, "Implementasi Algoritma k-Nearest Neighbor (k-NN) pada Data Ulasan Pelaksanaan Pembelajaran Daring", *JIKOMTI: Jurnal Ilmiah Ilmu Komputer dan Teknologi*, vol. 1, no.1, 2024.