

Disetujui Sempu  
Fili  
Andiansyah

**PENERAPAN *GREY WOLF OPTIMIZER* DALAM PENJADWALAN MATA KULIAH  
DENGAN FITUR KONSTRAIN DINAMIS PADA PROGRAM STUDI S1 INFORMATIKA  
UNIVERSITAS AHMAD DAHLAN**

**PROPOSAL SKRIPSI**

Disusun untuk memenuhi sebagian persyaratan mencapai derajat Sarjana



**Disusun Oleh:**

Bagas Tsiqoh Fiqyan Uwaidha

2000018287

**PROGRAM STUDI S1 INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS AHMAD DAHLAN**

**2024**

**PENERAPAN *GREY WOLF OPTIMIZER* DALAM PENJADWALAN MATA KULIAH  
DENGAN FITUR KONSTRAIN DINAMIS PADA PROGRAM STUDI S1 INFORMATIKA  
UNIVERSITAS AHMAD DAHLAN**

**PROPOSAL SKRIPSI**

**Disusun untuk memenuhi sebagian persyaratan mencapai derajat Sarjana**



**Disusun Oleh:**

Bagas Tsiqoh Fiqyan Uwaidha

2000018287

**PROGRAM STUDI S1 INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS AHMAD DAHLAN**

**2024**

**LEMBAR PERSETUJUAN PEMBIMBING  
PROPOSAL SKRIPSI**

**PENERAPAN *GREY WOLF OPTIMIZER* DALAM PENJADWALAN MATA KULIAH  
DENGAN FITUR KONSTRAIN DINAMIS PADA PROGRAM STUDI S1 INFORMATIKA  
UNIVERSITAS AHMAD DAHLAN**

Dipersiapkan dan disusun oleh:  
**BAGAS TSIQOH FIQYAN UWAIDHA  
2000018287**

**Program Studi S1 Informatika  
Fakultas Teknologi Industri  
Universitas Ahmad Dahlan**

**Telah disetujui oleh:  
Pembimbing**

**Dr. Ardiansyah, S.T., M.Cs.  
19790723 200309 111 0932301**

## LEMBAR PERNYATAAN KEASLIAN

### SURAT PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Bagas Tsiqoh Fiqyan Uwaidha

NIM : 2000018287

Prodi : Informatika

Judul TA/Skripsi : Penerapan *Grey Wolf Optimizer* dalam Penjadwalan Mata Kuliah dengan

Fitur Konstrain Dinamis pada Program Studi S1 Informatika Universitas Ahmad Dahlan

Dengan ini saya menyatakan bahwa Laporan Tugas Akhir ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar Ahli Madya/Kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 20 Juli 2024

Mengetahui,  
Dosen Pembimbing

Yang menyatakan,

Dr. Ardiansyah, S.T., M.Cs.  
19790723 200309 111 0932301

Bagas Tsiqoh Fiqyan Uwaidha  
2000018287

## DAFTAR ISI

HALAMAN JUDUL .....	i
LEMBAR PERSETUJUAN PEMBIMBING .....	ii
LEMBAR PERNYATAAN KEASLIAN .....	iii
DAFTAR ISI .....	iv
DAFTAR TABEL .....	vi
DAFTAR GAMBAR .....	vii
DAFTAR LAMPIRAN .....	viii
ABSTRAK .....	ix
BAB I. PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Tujuan Penelitian .....	3
1.4. Manfaat Penelitian .....	3
BAB II. TINJAUAN PUSTAKA .....	4
2.1. Kajian Penelitian Terdahulu .....	4
2.2. Landasan Teori .....	8
2.2.1. Penjadwalan .....	8
2.2.2. Optimasi .....	8
2.2.3. Algoritma <i>Grey Wolf Optimizer</i> (GWO) .....	9
2.2.4. Studi kasus <i>Grey Wolf Optimizer</i> (GWO) dalam penjadwalan mata kuliah .....	12
BAB III. METODE PENELITIAN .....	17
3.1. Tahapan Penelitian .....	17
3.2. Pengumpulan Data .....	18
3.3. Penentuan Alat .....	18
3.3.1. Hardware .....	18
3.3.2. Software .....	18
3.4. Analisis .....	19
3.5. Perancangan .....	19
3.5.1. Perancangan Model .....	20
3.5.2. Perancangan Proses .....	20
3.5.3. Perancangan Database .....	20

3.5.4. Perancangan Antarmuka .....	21
3.6. Implementasi .....	21
3.7. Pengujian .....	21
3.8. Jadwal Penelitian .....	22
DAFTAR PUSTAKA .....	23
LAMPIRAN .....	25

## DAFTAR TABEL

Tabel 2.1 Kajian Penelitian Terdahulu .....	7
Tabel 2.2 Inisialisasi Data Mata Kuliah .....	12
Tabel 2.3 Inisialisasi Data Ruang .....	12
Tabel 2.4 Inisialisasi Data Hari .....	12
Tabel 2.5 Inisialisasi Data Jam .....	13
Tabel 2.6 Batas $X_{min}$ dan $X_{max}$ pada Setiap Mata Kuliah .....	13
Tabel 2.7 Menentukan Posisi Individu 1 .....	13
Tabel 2.8 Menentukan Posisi Individu 2 .....	13
Tabel 2.9 Menentukan Posisi Individu 3 .....	13
Tabel 2.10 Menentukan Posisi Individu 4 .....	14
Tabel 2.11 Posisi Awal Serigala Secara Acak .....	14
Tabel 2.12 Update Posisi Serigala .....	16

## DAFTAR GAMBAR

Gambar 2.1 Flowchart Algoritma GWO .....	10
Gambar 3.1 Alur Tahapan Penelitian .....	17
Gambar 3.2 Jadwal Penelitian .....	22



## DAFTAR LAMPIRAN

Lampiran 1 Halaman Beranda .....	25
Lampiran 2 Halaman Proses .....	25
Lampiran 3 Halaman Hasil .....	26
Lampiran 4 Tabel Dataset .....	26

## ABSTRAK

Penjadwalan perkuliahan di perguruan tinggi berperan penting dalam memastikan kelancaran dan efektivitas proses belajar mengajar. Saat ini, Program Studi S1 Informatika Universitas Ahmad Dahlan masih melakukan penjadwalan kuliah menggunakan Excel, yang mana proses ini memakan waktu, rentan terhadap kesalahan, dan sulit beradaptasi dengan perubahan. Kondisi ini menciptakan tantangan dalam manajemen waktu dan sumber daya, serta meningkatkan kemungkinan bentrokan jadwal.

Penelitian ini bertujuan untuk mengembangkan sistem penjadwalan mata kuliah berbasis web yang menggunakan algoritma *Grey Wolf Optimizer* (GWO) dengan fitur penentuan konstrain secara dinamis. Sistem ini dirancang untuk mengatasi masalah *hard constraint*, seperti bentrokan jadwal dosen, dan *soft constraint*, seperti meminimalkan perpindahan ruang kelas bagi mahasiswa. Dengan penerapan algoritma GWO, diharapkan sistem ini mampu menghasilkan jadwal yang lebih optimal dan dinamis.

Penelitian ini akan menghasilkan sebuah sistem yang diharapkan dapat membantu dalam proses penjadwalan kondusif, meningkatkan efisiensi pembelajaran, dan mengurangi biaya operasional. Diharapkan bahwa implementasi sistem penjadwalan kuliah dengan fitur penentuan konstrain secara dinamis ini akan memberikan manfaat yang signifikan di Program Studi S1 Informatika UAD.

**Kata Kunci:** *Grey Wolf Optimizer*, Optimasi, Penjadwalan kuliah, Sistem penjadwalan otomatis, Universitas Ahmad Dahlan.

## **BAB I. PENDAHULUAN**

### **1.1. Latar Belakang**

Perguruan tinggi merupakan institusi pendidikan tinggi yang menyelenggarakan pendidikan akademik dan/atau vokasi, serta melakukan penelitian dan pengabdian kepada masyarakat. Saat ini, ekspektasi masyarakat terhadap sektor pendidikan semakin tinggi, khususnya pada perguruan tinggi, yang diharapkan dapat berperan aktif dalam meningkatkan daya saing bangsa dengan menyelenggarakan pendidikan tinggi yang berkualitas, meningkatkan kualitas dosen dan staf pendidikan adalah investasi jangka panjang untuk masa depan pendidikan di Indonesia [1].

Penjadwalan perkuliahan di perguruan tinggi berperan penting dalam memastikan kelancaran dan efektivitas proses belajar mengajar. Tidak hanya dosen sebagai pengajar, Partisipasi aktif mahasiswa dalam proses penjadwalan perkuliahan dapat membantu perguruan tinggi menyusun jadwal yang lebih efektif dan efisien [2]. Keterlibatan semua pihak terkait, termasuk dosen, staf administrasi akademik, dan mahasiswa, sangat penting untuk menghasilkan jadwal mata kuliah yang efektif [3].

Penjadwalan mata kuliah di perguruan tinggi harus mempertimbangkan kebutuhan dan preferensi baik dosen maupun mahasiswa. Sistem penjadwalan mata kuliah yang efisien sangat penting untuk memastikan bahwa dosen dan mahasiswa dapat mengikuti mata kuliah sesuai dengan jadwal yang telah ditetapkan tanpa terjadi tabrakan atau tumpang tindih [4].

Sebelumnya, banyak perguruan tinggi (PT) melakukan penjadwalan kuliah secara manual. Salah satu contohnya pada Program Studi S1 Informatika UAD saat ini masih melakukan penjadwalan kuliah menggunakan Excel, yang berarti staf di PT harus menyusun jadwal secara langsung. Namun, setelah diteliti lebih lanjut, proses ini tidak ideal karena memakan waktu, rentan kesalahan, dan sulit beradaptasi dengan perubahan. Oleh karena itu, banyak perguruan tinggi mulai menggunakan sistem penjadwalan otomatis untuk mengurangi beban staf dan meningkatkan efisiensi proses penjadwalan.

Seiring dengan berjalannya waktu, muncul berbagai cara atau metode untuk menjadwalkan kuliah agar lebih efektif dan efisien yaitu *Particle Swarm Optimization*, *Firefly Algorithm*, dan *Grey Wolf Optimizer*. Peneliti

dari Taiwan menggunakan algoritma Particle Swarm Optimization di dalam penelitiannya dari Cheng-Siu University [5], sementara itu, penelitian di universitas Ma Chung yang menggunakan *Firefly Algorithm* [6], dan peneliti di Universitas Sathyabama, India menggunakan algoritma *Grey Wolf Optimizer* di dalam penelitiannya [7]. Meskipun setiap penelitian menggunakan pendekatan yang berbeda, namun semuanya menunjukkan potensi yang menjanjikan dalam mencapai tujuan tersebut. Hal ini menunjukkan bahwa pendekatan *metaheuristik* dapat menjadi solusi yang efektif dalam menangani kompleksitas masalah penjadwalan kuliah.

Penelitian sebelumnya menunjukkan bahwa terdapat satu komponen fundamental yaitu penerapan batasan atau konstrain. Pembagian konstrain ini terdiri dari dua jenis yaitu *hard constraint* dan *soft constraint*. Contoh dari *hard constraint* yaitu perkuliahan tidak boleh bentrok dengan waktu mengajar dosen di kelas lain, sedangkan *soft constraint* yang diupayakan untuk dipenuhi, meskipun tidak selalu tercapai, contohnya adalah meminimalkan perpindahan ruang kelas bagi mahasiswa. Studi sebelumnya menunjukkan bahwa penerapan batasan atau konstrain dalam penjadwalan mata kuliah dibuat secara statis. Tantangan utama dari pendekatan statis ini adalah sulitnya mengubah konstrain karena perlu dilakukan secara langsung melalui *source code*, yang menyebabkan sistem mengalami kesulitan dalam beradaptasi dengan perubahan.

Penelitian ini dilakukan dengan membangun sebuah sistem penjadwalan mata kuliah di Program Studi S1 Informatika UAD dengan fitur penentuan konstrain secara dinamis. Artinya, penentuan konstrain akan didukung oleh fitur yang langsung dikembangkan pada sistem penjadwalan. Sistem ini dikembangkan menggunakan algoritma *Grey Wolf Optimizer* (GWO). Diharapkan bahwa implementasi sistem penjadwalan kuliah dengan fitur penentuan konstrain secara dinamis ini akan memberikan manfaat yang signifikan di Program Studi S1 Informatika UAD.

## 1.2. Rumusan Masalah

Berdasarkan pembahasan latar belakang di atas, rumusan masalah yang teridentifikasi adalah bagaimana membangun sistem penjadwalan kuliah untuk Program Studi S1 Informatika UAD menggunakan Algoritma GWO dengan fitur menentukan konstrain secara dinamis?

### **1.3. Tujuan Penelitian**

Dari berbagai hal di atas tujuan dari penelitian ini sebagai berikut :

1. Membangun sistem penjadwalan mata kuliah di Program Studi S1 Informatika UAD menggunakan Algoritma GWO yang berbasis web.
2. Mengimplementasikan fitur penentuan konstrain secara dinamis dalam sistem penjadwalan tersebut.
3. Mengevaluasi performa algoritma GWO.
4. Menguji sistem aplikasi penjadwalan mata kuliah yang telah dibangun.

### **1.4. Manfaat Penelitian**

Manfaat dari penelitian ini tentang sistem penjadwalan kuliah dengan penentuan konstrain secara dinamis akan memberikan manfaat bagi tim pengelola program studi S1 Informatika UAD dengan mengurangi waktu yang diperlukan dalam proses penjadwalan, mengurangi risiko kesalahan, dan meningkatkan adaptabilitas terhadap perubahan. Ini berkontribusi pada penciptaan lingkungan belajar yang kondusif, meningkatkan efisiensi pembelajaran, dan mengurangi biaya operasional.

## BAB II. TINJAUAN PUSTAKA

### 2.1. Kajian Penelitian Terdahulu

Penelitian ini mengkaji studi sebelumnya tentang penjadwalan mata kuliah di universitas. Kajian penelitian [8] menggunakan *Algoritma Genetika* (GA), *Simulated Annealing* (SA), *Particle Swarm Optimization* (PSO), *Cuckoo Search* (CS), dan algoritma berbasis *local search*. Tujuan dari penelitian ini untuk mengevaluasi efektivitas metode metaheuristik dalam menyelesaikan masalah penjadwalan mata kuliah di universitas. Hasil penelitian menunjukkan bahwa metode *Simulated Annealing* (SA) memiliki kemampuan eksploitasi yang tinggi dan dapat menemukan solusi berkualitas tinggi dalam waktu terbatas, sementara *Particle Swarm Optimization* (PSO) dan *Cuckoo Search* (CS) lebih baik dalam hal eksploitasi.

Selanjutnya penelitian [9] yang menggunakan metode *Particle Swarm Optimisation* (PSO) dan variasi-variasinya, serta hybridisasi dengan algoritma lain seperti *Genetic Algorithm* (GA) dan *Modified Backtracking Search Algorithm* (MBSA). Tujuan dari penelitian ini adalah untuk mengembangkan algoritma optimasi yang efektif dalam menghasilkan jadwal kelas yang efisien biaya untuk sistem pendidikan. Hasil penelitian ini menunjukkan bahwa variasi PSO yang digunakan, seperti MCPSO + IO(75%):EO(25%), dapat memberikan hasil yang paling baik dalam menghasilkan jadwal kelas yang efisien biaya. Penelitian ini juga menunjukkan bahwa penggunaan hybridisasi algoritma dapat meningkatkan kinerja optimasi dalam menyelesaikan masalah jadwal kelas.

Penelitian ketiga [10] berfokus pada penggunaan metode *Simulated Annealing* (SA) dengan *penalization* untuk masalah penjadwalan mata kuliah pada universitas. Tujuan dari penelitian ini adalah untuk mengetahui seberapa efektif metode ini dalam menyelesaikan masalah penjadwalan mata kuliah di universitas. Hasilnya menunjukkan bahwa metode ini efektif dalam menyelesaikan

masalah penjadwalan mata kuliah di universitas. Dibandingkan dengan metode lainnya, SA dengan *penalization* menunjukkan performa yang unggul dalam hal stabilitas dan kualitas solusi.

Penelitian selanjutnya merujuk pada penelitian [3] menggunakan metode tinjauan literatur untuk menganalisis dan membandingkan berbagai algoritma untuk optimasi penjadwalan mata kuliah. Tujuan dari studi ini adalah untuk menemukan dan menganalisis literatur yang relevan tentang optimasi penjadwalan untuk menemukan algoritma yang paling efektif untuk penjadwalan mata kuliah. Studi ini menganalisis 26 makalah dan menemukan bahwa 88% dari mereka berfokus pada penjadwalan menggunakan metode Algoritma Genetik (GA) pada universitas untuk optimalisasi penjadwalan. Hasil dari penelitian tersebut Algoritma Simulated Annealing (SA) menghasilkan solusi dengan waktu proses antara 0.481-10.102 detik. Algoritma Genetic Algorithm (GA) menghasilkan solusi dengan waktu proses antara 0.964-73.461 detik dan nilai fitness mencapai 1.

Penelitian selanjutnya [4] menganalisis kompleksitas menggunakan ukuran yang dikembangkan oleh para peneliti. Ukuran ini mengukur berbagai aspek dari *instance* penjadwalan untuk menilai tingkat kesulitannya. Tujuan penelitian ini adalah untuk mengembangkan dan menguji ukuran kompleksitas yang dapat digunakan untuk mengevaluasi kesulitan dari *instance* penjadwalan perkuliahan, sehingga dapat membantu dalam pemilihan metode penyelesaian yang sesuai. Hasil penelitian menunjukkan bahwa ukuran kompleksitas yang diusulkan efektif dalam mengevaluasi kesulitan berbagai *instance* penjadwalan. Ukuran ini dapat digunakan sebagai alat bantu untuk memilih metode penyelesaian yang paling tepat berdasarkan tingkat kompleksitas *instance*.

Berdasarkan uraian diatas dapat disimpulkan bahwa metode *Grey Wolf Optimizer* (GWO) belum digunakan dalam penelitian penjadwalan mata kuliah. Oleh karena itu, pada penelitian ini digunakan

algoritma GWO dalam mengoptimasi penjadwalan mata kuliah. Harapannya dapat mengoptimalkan proses penjadwalan mata kuliah secara dinamis pada program studi Informatika.

Berdasarkan analisis sebelumnya, tabel perbandingan studi dapat ditemukan di Tabel 2.1. Tabel ini merangkum artikel jurnal penelitian yang telah diterbitkan sebelumnya



Tabel 2.1 Kajian Penelitian Terdahulu

Peneliti	Dataset	Variabel	Metode	Hasil
Sina Abdipoor, Razali Yaakob, Say Leng Goh, dan Salwani Abdullah	penjadwalan mata kuliah universitas	Kualitas solusi, kemampuan eksploitasi, kemampuan eksplorasi	Algoritma Genetika, Annealing Simulasi, Optimasi Partikel Swarm, Algoritma Pencarian Lokal	Particle Swarm Optimization (PSO) dan Cuckoo Search (CS) lebih baik dalam eksploitasi, Simulated Annealing (SA) memiliki kemampuan eksploitasi yang tinggi
Thatchai Thepphakorn, Saisumpan Sooncharoen, dan Pupong Pongcharoen	penjadwalan mata kuliah universitas	Biaya operasional, waktu komputasi, kecepatan konvergensi, dan waktu komputasi	Hibridisasi SPSO dan MCPSO dengan lima kombinasi rasio IO dan EO dalam HPSOT	Metode hibrid HPSOT memiliki waktu komputasi rata-rata yang lebih cepat dibandingkan dengan SPSO dan MCPSO konvensional.
K. Sylejmani, E. Gashi, dan A. Ymeri	penjadwalan mata kuliah universitas	Kualitas jadwal, konflik jadwal, <i>penalization</i> .	<i>Simulated Annealing</i> (SA) dengan <i>penalization</i>	SA dengan <i>penalization</i> menghasilkan jadwal yang lebih baik
T. Handayani, D. H. Fudholi, dan S. Rani	26 makalah tentang algoritma optimasi penjadwalan mata kuliah	Waktu pengerjaan, Nilai kebugaran	Tinjauan literatur, Analisa algoritma	<i>Simulated Annealing</i> (SA) cenderung lebih cepat dari <i>Genetic Algorithm</i> (GA)
F. de la Rosa-Rivera, J. I. Nunez-Varela, C. A. Puente-Montejano, dan S. E. Nava-Muñoz	Berbagai <i>instance</i> penjadwalan perkuliahan	Kompleksitas <i>instance</i>	Analisis kompleksitas	Ukuran kompleksitas efektif dalam mengevaluasi kesulitan <i>instance</i> , membantu pemilihan metode.

## 2.2. Landasan Teori

### 2.2.1. Penjadwalan

Penjadwalan adalah proses perencanaan yang menentukan kapan dan di mana setiap operasi dalam suatu pekerjaan harus dilakukan menggunakan sumber daya yang terbatas. Menurut Pinedo [11], penjadwalan adalah proses pengambilan keputusan yang memegang peranan yang penting dalam manufaktur dan sistem produksi. Pada dasarnya, penjadwalan mencakup pengurutan aktivitas, pengalokasian aktivitas pada fasilitas, dan pemetaan aktivitas menurut urutan waktu.

Ada dua cara untuk melakukan penjadwalan. Yang pertama adalah Penjadwalan Forward, di mana operasi dijadwalkan dari awal dan bergerak searah dengan waktu sampai operasi selesai. Yang kedua adalah Penjadwalan Secara EDD, atau Tanggal Terakhir Penjadwalan: urutan pekerjaan diatur berdasarkan tanggal jatuh tempo pesanan yang dijanjikan.

### 2.2.2. Optimasi

Algoritma Optimasi (*Optimization Algorithms*) merupakan algoritma atau metode numerik untuk mencari nilai  $x$  sehingga menghasilkan  $f(x)$  dengan tujuan meminimalkan atau memaksimalkan nilai fungsi tersebut, yang biasanya disertai dengan batasan (*constraints*) pada variabel keputusan  $x$  [12].

Konsep dasar dari Optimasi yaitu Fungsi Objektif  $f(x)$  yang nilainya ingin diminimalkan atau dimaksimalkan, lalu variabel keputusan yaitu variabel  $x$  yang nilai-nilainya harus ditentukan untuk mengoptimalkan  $f(x)$ , dan juga batasan (*constraints*) yaitu batasan-batasan yang harus dipenuhi oleh variabel keputusan  $x$  yang bisa berupa persamaan atau pertidaksamaan.

### 2.2.3. Algoritma Grey Wolf Optimizer (GWO)

Algoritma Grey Wolf Optimizer (GWO) merupakan algoritma metaheuristik yang diusulkan oleh Mirjalili *et al.* pada tahun 2014 [13]. Algoritma ini terinspirasi oleh perilaku berburu dan struktur sosial dari serigala abu-abu yang meniru hierarki kepemimpinan dan mekanisme berburu dari serigala abu-abu dalam alam liar termasuk pengaturan formasi, komunikasi, dan strategi penyerangan yang terkoordinasi. Berikut adalah alur dari algoritma GWO:

1. Inisialisasi Populasi: Inisialisasi populasi serigala dengan posisi acak dalam ruang pencarian.
2. Identifikasi *Alpha*, *Beta*, dan *Delta*: Evaluasi *fitness* dari setiap serigala dan identifikasi serigala alpha, beta, dan delta berdasarkan nilai *fitness* terbaik.
3. *Update* Posisi Serigala:
  - a. Serigala alpha, beta, dan delta memberikan panduan bagi serigala lainnya.
  - b. *Update* posisi serigala lainnya berdasarkan persamaan berikut:

$$D_{\alpha} = |C_1 \cdot X_{\alpha} - X| \quad (1)$$

$$X_1 = X_{\alpha} - A_1 \cdot D_{\alpha} \quad (2)$$

$$D_{\beta} = |C_2 \cdot X_{\beta} - X| \quad (3)$$

$$X_2 = X_{\beta} - A_2 \cdot D_{\beta} \quad (4)$$

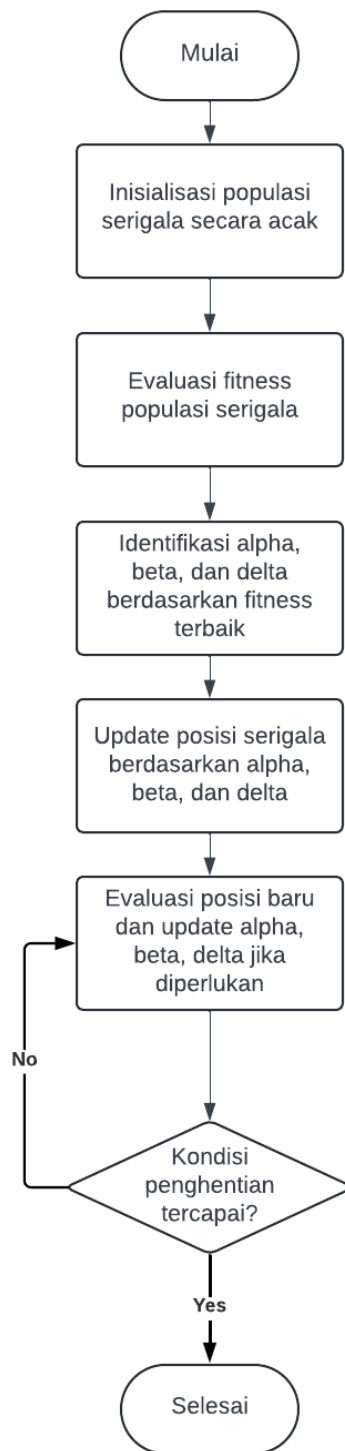
$$D_{\delta} = |C_3 \cdot X_{\delta} - X| \quad (5)$$

$$X_3 = X_{\delta} - A_3 \cdot D_{\delta} \quad (6)$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \quad (7)$$

- c. Di mana *A* dan *C* adalah koefisien yang di-update setiap iterasi, dan *X* adalah posisi serigala.
4. Evaluasi dan Update Posisi: Evaluasi posisi baru dari serigala dan update nilai alpha, beta, dan delta jika diperlukan.
5. Penghentian: Ulangi langkah 3 dan 4 hingga kondisi penghentian tercapai (misalnya jumlah iterasi maksimum atau konvergensi).

Flowchart Algoritma GWO dapat dilihat pada gambar 2.1 Berikut.



Gambar 2.1 Flowchart Algoritma GWO

Pseudocode Algoritma GWO dapat dilihat pada tabel berikut.

1. Inisialisasi populasi serigala secara acak
2. Evaluasi fitness setiap serigala
3. Identifikasi alpha, beta, dan delta berdasarkan fitness terbaik
4. While (kondisi penghentian tidak tercapai) do
5.   For setiap serigala (i) in populasi do
6.     Update koefisien A dan C
7.     Hitung D\_alpha, D\_beta, D\_delta
8.     Update posisi serigala (X) berdasarkan alpha, beta, dan delta
9.   End for
10. Evaluasi posisi baru dari setiap serigala
11. Update alpha, beta, dan delta jika diperlukan
12. End while
13. Return alpha sebagai solusi terbaik

Function update\_koefisien(iterasi, iterasi\_max)

1.  $a = 2 - \text{iterasi} * (2 / \text{iterasi\_max})$
2.  $A = 2 * a * \text{rand}() - a$
3.  $C = 2 * \text{rand}()$
4. Return A, C

Function hitung\_D(X\_serigala, X\_target, C)

1.  $D = |C * X\_target - X\_serigala|$
2. Return D

Function update\_posisi(X\_serigala, D, A, X\_target)

1.  $X\_new = X\_target - A * D$
2. Return X\_new

#### 2.2.4. Studi kasus *Grey Wolf Optimizer (GWO)* dalam penjadwalan mata kuliah

Sebelum memasuki tahap penerapan algoritma *Grey Wolf Optimization*, peneliti harus menerapkan terlebih dahulu batasan (constraints). Jika terdapat pelanggaran terhadap batasan keras, maka dikenakan penalti sebesar 1. jika terdapat pelanggaran terhadap batasan lunak, maka dikenakan penalti sebesar 0,5. dan jika tidak ada pelanggaran terhadap batasan keras dan lunak, penalti yang diberikan sebesar 0. Batasan keras pada studi kasus ini adalah tidak ada kelas yang dijadwalkan pada jam yang sama, sedangkan batasan lunak adalah perkuliahan tidak dijadwalkan pada slot waktu paling awal dan slot waktu paling akhir. Setelah menentukan batasan, peneliti melakukan tahapan pertama yang terdapat pada gambar 2.1.

- a. Inisialisasi populasi serigala secara acak

Tabel 2.2, 2.3, 2.4, dan 2.5 menunjukkan inisialisasi dari data mata kuliah, ruang, hari, dan jam yang akan digunakan untuk studi kasus penjadwalan perkuliahan menggunakan algoritma GWO:

No.	Mata Kuliah	Kelas
1	Strategi Algoritma	A
2	Rekayasa Web	B
3	Interaksi Manusia dan Komputer	C
4	Kriptografi	D

Tabel 2.2 Inisialisasi Data Mata Kuliah

No.	Ruang
1	4.1.5.01
2	4.1.5.02
3	4.1.5.03

Tabel 2.3 Inisialisasi Data Ruang

No.	Hari
1	Senin
2	Selasa

Tabel 2.4 Inisialisasi Data Hari

No.	Jam	
1	7:00	9:30
2	10:00	12:30

3	13:00	13:30
4	16:00	16:30

Tabel 2.5 Inisialisasi Data Jam

Iterasi maksimal yang digunakan adalah 1, dan jumlah mata kuliah yang digunakan adalah 4, sehingga populasi serigala akan berjumlah 4. Selanjutnya menentukan nilai batas maksimal yang akan disesuaikan pada jumlah mata kuliah, ruang, hari, dan jam.

Batas Max /Min	Hari	Jam	Ruangan
$X_{min}$	1	1	1
$X_{max}$	2	4	3

Tabel 2.6 Batas  $X_{min}$  dan  $X_{max}$  pada Setiap Mata Kuliah

$$X = X_{min} + rand (X_{max} - X_{min}) \quad (8)$$

$X$  = posisi awal

$X_{min}$  = batas minimal

$X_{max}$  = batas maksimal

$rand$  = nilai random antara 0 dan 1

Menggunakan persamaan (8) dapat diperoleh posisi secara acak lokasi serigala seperti pada tabel 2.7, 2.8, 2.9, dan 2.10.

Strategi Algoritma		Pembulatan
Hari	1.06	1
Jam	1.61	2
Ruangan	2.88	3

Tabel 2.7 Menentukan Posisi Individu 1

Rekayasa Web		Pembulatan
Hari	1.18	1
Jam	2.58	3
Ruangan	2.36	2

Tabel 2.8 Menentukan Posisi Individu 2

Interaksi Manusia dan Komputer		Pembulatan
Hari	1.52	2
Jam	3.1	3
Ruangan	1.37	1

Tabel 2.9 Menentukan Posisi Individu 3

Kriptografi		Pembulatan
Hari	1.85	2
Jam	1.1	1
Ruangan	2.99	3

Tabel 2.10 Menentukan Posisi Individu 4

Hasil dari penentuan posisi serigala secara acak dapat dilihat pada tabel 2.11 berikut:

Hari	Jam		Ruang		
			4.1.5.01	4.1.5.02	4.1.5.03
Senin	7:00	9:30			
	10:00	12:30			Strategi Algoritma
	13:00	13:30		Rekayasa Web	
	16:00	16:30			
Selasa	7:00	9:30			Kriptografi
	10:00	12:30			
	13:00	13:30	Interaksi Manusia dan Komputer		
	16:00	16:30			

Tabel 2.11 Posisi Awal Serigala Secara Acak

- b. Evaluasi *fitness* populasi serigala

Menghitung nilai *fitness* menggunakan persamaan (9).

$$f = \frac{1}{(fk + gk + 1)} \quad (9)$$

$f$  = nilai *fitness*

$fk$  = batasan keras

$gk$  = batasan lunak

- c. Identifikasi alpha, beta, dan delta berdasarkan nilai *fitness* terbaik

*Fitness* untuk masing-masing posisi serigala adalah:

Strategi Algoritma  $f = \frac{1}{(0 + 0 + 1)} = 1$

Rekayasa Web  $f = \frac{1}{(0 + 0 + 1)} = 1$



Kriptografi	$f = \frac{1}{(0 + 1 + 1)} = 0.67$
Interaksi Manusia dan Komputer	$f = \frac{1}{(0 + 0 + 1)} = 1$

Seluruh kelas dijadwalkan pada jam yang berbeda dan di ruangan yang berbeda, sehingga tidak ada pelanggaran batasan keras ( $fk = 0$ ). Sedangkan kelas Kriptografi melanggar batasan lunak ( $gk = 0.5$ ). Alpha berdasarkan nilai fitness tertinggi yaitu Strategi Algoritma, Rekayasa Web, dan Interaksi Manusia dan Komputer dengan nilai *fitness* yang sama. Sedangkan delta berdasarkan nilai *fitness* terendah adalah Kriptografi.

d. Update posisi serigala berdasarkan alpha, beta, dan delta

Pada proses *update* posisi serigala dalam melibatkan persamaan (7) yang mempertimbangkan posisi alpha, beta, dan delta. Misalkan parameter  $r_1$  dan  $r_2$  untuk seluruh serigala adalah sebagai berikut:

- Serigala Alpha (Strategi Algoritma):  $r_1 = 0.07, r_2 = 0.04$
- Serigala Beta (Rekayasa Web):  $r_1 = 0.36, r_2 = 0.24$
- Serigala Delta (Interaksi Manusia dan Komputer):  $r_1 = 0.73, r_2 = 0.66$

Selanjutnya hitung parameter  $A$  dan  $C$  untuk setiap serigala menggunakan persamaan (10) dan (11).

$$A = 2 \cdot r_1 - 1 \quad (10)$$

$$C = 2 \cdot r_2 \quad (11)$$

$$\text{Serigala Alpha } A_1 = -0.86, C_1 = 0.08$$

$$\text{Serigala Beta } A_2 = -0.28, C_2 = 0.48$$

$$\text{Serigala Delta } A_3 = 0.46, C_3 = 1.32$$

Hitung Posisi Baru untuk mata kuliah kriptografi berdasarkan posisi serigala alpha, beta, dan delta menggunakan persamaan (1), (2), (3), (4), (5), (6), dan (7).

$$D_\alpha = 1.92, 0.84, 2.76$$

$$X_1 = 2.6512, 2.7224, 5.3736$$

$$D_\beta = 1.52, 0.44, 2.04$$

$$X_2 = 1.4256, 3.1232, 2.5712$$

$$D_\delta = 0.64, 2.96, 1.68$$

$$X_3 = 1.7056, 1.6384, 0.2272$$

$$X_{Kriptografi} = 2, 2, 3$$

Hasil dari *update* posisi serigala berdasarkan alpha, beta, dan delta dapat dilihat pada tabel

2.12.

Hari	Jam		Ruangan		
			4.1.5.01	4.1.5.02	4.1.5.03
Senin	7:00	9:30			
	10:00	12:30			Strategi Algoritma
	13:00	13:30		Rekayasa Web	
	16:00	16:30			
Selasa	7:00	9:30			
	10:00	12:30			Kriptografi
	13:00	13:30	Interaksi Manusia dan Komputer		
	16:00	16:30			

Tabel 2.12 Update Posisi Serigala

e. Evaluasi *fitness* populasi serigala

Evaluasi *fitness* pada iterasi untuk masing-masing posisi serigala adalah:

$$\text{Strategi Algoritma} \quad f = \frac{1}{(0 + 0 + 1)} = 1$$

$$\text{Rekayasa Web} \quad f = \frac{1}{(0 + 0 + 1)} = 1$$

$$\text{Kriptografi} \quad f = \frac{1}{(0 + 0 + 1)} = 1$$

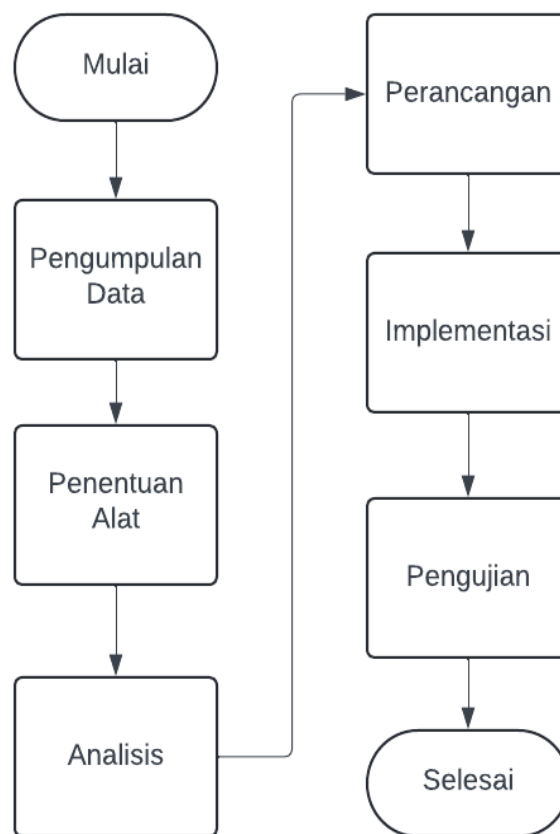
$$\begin{array}{l} \text{Interaksi Manusia dan} \\ \text{Komputer} \end{array} \quad f = \frac{1}{(0 + 0 + 1)} = 1$$

Dengan demikian, seluruh serigala mencapai nilai *fitness* maksimal setelah update posisi.

## BAB III. METODE PENELITIAN

### 3.1. Tahapan Penelitian

Alur Tahapan Penelitian pada Gambar 3.1 untuk Mengembangkan Sistem Penjadwalan Mata Kuliah Secara Dinamis dimulai dari pengumpulan data, penentuan alat, analisis, perancangan, implementasi, dan pengujian untuk mengembangkan sistem penjadwalan mata kuliah secara dinamis. Hasil yang diharapkan pada penelitian ini adalah penjadwalan mata kuliah menggunakan algoritma *Grey Wolf Optimizer* (GWO) mampu meningkatkan efisiensi dalam melakukan penyusunan jadwal mata kuliah di program studi Informatika Universitas Ahmad Dahlan.



Gambar 3.1 Alur Tahapan Penelitian

### 3.2. Pengumpulan Data

Metode pengumpulan data dalam penelitian ini dilakukan melalui wawancara dengan Bapak Guntur Zamroni, yang menjabat sebagai Sekretaris Program Studi (Sekprodi) Informatika di Universitas Ahmad Dahlan (UAD). Tujuan wawancara ini adalah untuk mengetahui kebutuhan pengguna dan bagaimana proses penjadwalan berlangsung di Program Studi Informatika.

Selain wawancara, penelitian ini juga menggunakan metode studi literatur untuk memperbanyak informasi yang dibutuhkan. Studi literatur melibatkan pengumpulan dan analisis data sekunder yang berasal dari berbagai sumber seperti jurnal ilmiah, buku, laporan, dan dokumen terkait lainnya. Data yang diperoleh mencakup informasi penting seperti total mahasiswa, daftar dosen, mata kuliah yang diajarkan, dan ruang kelas yang tersedia. Seluruh data tersebut tersedia dalam bentuk *file* Excel.

### 3.3. Penentuan Alat

#### 3.3.1. *Hardware*

*Hardware* yang digunakan dalam penelitian ini:

1. Laptop ASUS TUF A15 FA506IC
2. Processor AMD Ryzen 7 4800H
3. RAM 16GB
4. GPU NVIDIA GeForce RTX 3050 Laptop GPU

#### 3.3.2. *Software*

*Software* yang digunakan dalam penelitian ini:

1. Visual Studio Code
2. Windows 11

3. Brave
4. Node.Js
5. ExpressJS
6. ReactJS
7. Tailwind
8. MySQL
9. Python
10. JavaScript
11. Github
12. Xampp
13. NGROK

### **3.4. Analisis**

Analisis kebutuhan sistem dilakukan untuk menguraikan kebutuhan pengguna. Analisis sistem mencakup identifikasi kebutuhan fungsional dan non-fungsional, serta fitur dan fungsi yang harus ada seperti mengelola data total siswa, dosen, mata kuliah, dan ruang kelas. Selain itu, sistem harus memungkinkan pengguna menjadwalkan secara dinamis.

### **3.5. Perancangan**

Perancangan sistem mencakup pembuatan desain arsitektur sistem, diagram alur data, dan skema basis data. Desain antarmuka pengguna juga dibuat untuk memastikan kemudahan penggunaan dan efisiensi dalam proses penjadwalan.

### 3.5.1. Perancangan Model

Pada tahap perancangan model, dilakukan pengembangan model matematika dan algoritma yang digunakan dalam sistem penjadwalan. Model ini mencakup representasi formal dari berbagai elemen penjadwalan seperti mata kuliah, dosen, mahasiswa, dan ruang kelas. Tujuannya adalah untuk merumuskan masalah penjadwalan sebagai masalah optimisasi yang dapat diselesaikan menggunakan algoritma tertentu, seperti *Grey Wolf Optimizer* (GWO). Model ini juga mencakup berbagai konstrain yang perlu dipenuhi, seperti ketersediaan dosen dan ruang kelas.

### 3.5.2. Perancangan Proses

Perancangan proses melibatkan pembuatan diagram alur proses yang menggambarkan langkah-langkah operasional dalam sistem penjadwalan. Proses ini mencakup langkah-langkah dari pengumpulan data *input*, pemrosesan data, hingga menghasilkan *output* berupa jadwal. Diagram alur ini membantu dalam memahami urutan dan keterkaitan antara berbagai komponen sistem serta memastikan bahwa semua langkah penting telah teridentifikasi dan diakomodasi dalam desain sistem.

### 3.5.3. Perancangan Database

Perancangan database mencakup pembuatan skema database yang akan digunakan untuk menyimpan data terkait penjadwalan. Skema ini meliputi tabel-tabel yang menyimpan informasi tentang dosen, mata kuliah, mahasiswa, dan ruang kelas, serta relasi antara tabel-tabel tersebut. Desain database juga memastikan integritas data dan efisiensi dalam pengambilan serta penyimpanan data. ERD (*Entity Relationship Diagram*) dan DFD (*Data Flow Diagram*) digunakan untuk menggambarkan struktur database dan aliran data secara visual.

### 3.5.4. Perancangan Antarmuka

Perancangan antarmuka melibatkan pembuatan desain UI/UX untuk aplikasi penjadwalan. Antarmuka pengguna dirancang agar mudah digunakan dan intuitif, memungkinkan pengguna untuk mengakses dan mengelola informasi penjadwalan dengan mudah. Desain antarmuka mencakup tata letak halaman, navigasi, elemen interaktif, serta estetika visual yang mendukung pengalaman pengguna yang baik. *Wireframe* dan *mockup* digunakan untuk menggambarkan tampilan dan interaksi antarmuka sebelum implementasi.

### 3.6. Implementasi

Implementasi dilakukan dengan mengembangkan aplikasi berdasarkan desain yang telah dibuat. Pada tahap ini dilakukan pengkodean, pengujian unit, dan integrasi komponen sistem dilakukan untuk menguji setiap fitur dan memastikan bahwa mereka berfungsi. Sistem yang dihasilkan diharapkan mampu melakukan penjadwalan otomatis dengan fitur penentuan konstrain secara dinamis

### 3.7. Pengujian

Pengujian dilakukan untuk memastikan bahwa sistem yang telah dikembangkan berfungsi sesuai dengan spesifikasi dan kebutuhan pengguna. Metode pengujian yang digunakan adalah *blackbox testing*, di mana pengujian difokuskan pada *input* dan *output* sistem tanpa memeriksa struktur internal atau kode program. Pengujian ini mencakup berbagai skenario penggunaan untuk memeriksa apakah sistem dapat menghasilkan jadwal yang valid dan memenuhi konstrain yang telah ditentukan. Hasil pengujian digunakan untuk mengidentifikasi dan memperbaiki *bug* serta memastikan kualitas dan keandalan sistem.





## DAFTAR PUSTAKA

- [1] O. : Kadek, H. Primayana, M. Program, S. Magister, and P. Dasar Undiksha, "MANAJEMEN SUMBER DAYA MANUSIA DALAM PENINGKATAN MUTU PENDIDIKAN DI PERGURUAN TINGGI."
- [2] Y. Sari, M. Alkaff, E. S. Wijaya, S. Soraya, and D. P. Kartikasari, "OPTIMASI PENJADWALAN MATA KULIAH MENGGUNAKAN METODE ALGORITMA GENETIKA DENGAN TEKNIK TOURNAMENT SELECTION," vol. 6, no. 1, pp. 85–92, 2019, doi: 10.25126/jtiik.201961262.
- [3] T. Handayani, D. H. Fudholi, and S. Rani, "Kajian Algoritma Optimasi Penjadwalan Mata Kuliah," *PETIR*, vol. 13, no. 2, pp. 212–222, Sep. 2020, doi: 10.33322/petir.v13i2.1027.
- [4] F. de la Rosa-Rivera, J. I. Nunez-Varela, C. A. Puente-Montejano, and S. E. Nava-Muñoz, "Measuring the complexity of university timetabling instances," *Journal of Scheduling*, vol. 24, no. 1, pp. 103–121, Feb. 2021, doi: 10.1007/s10951-020-00641-y.
- [5] Shu-Chuan Chu, Yi-Tin Chen, and Jiun-Huei Ho, "Timetable Scheduling Using Particle Swarm Optimization," *Institute of Electrical and Electronics Engineers (IEEE)*, Oct. 2006, pp. 324–327. doi: 10.1109/icicic.2006.541.
- [6] I. Algoritma Kunang-Kunang Untuk Penjadwalan Mata Kuliah di Universitas Ma Chung Hendry Setiawan, L. Hanjaya Hanafi, and K. Rega Prilianti, "Implementasi Algoritma Kunang-Kunang untuk Penjadwalan Mata Kuliah di Universitas Ma Chung 269."

- [7] G. Natesan and A. Chokkalingam, "An improved grey wolf optimization algorithm based task scheduling in cloud computing environment," *International Arab Journal of Information Technology*, vol. 17, no. 1, pp. 73–81, 2020, doi: 10.34028/iajit/17/1/9.
- [8] S. Abdipoor, R. Yaakob, S. L. Goh, and S. Abdullah, "Meta-heuristic approaches for the University Course Timetabling Problem," *Intelligent Systems with Applications*, vol. 19. Elsevier B.V., Sep. 01, 2023. doi: 10.1016/j.iswa.2023.200253.
- [9] T. Thepphakorn, S. Sooncharoen, and P. Pongcharoen, "Particle Swarm Optimisation Variants and Its Hybridisation Ratios for Generating Cost-Effective Educational Course Timetables," *SN Comput Sci*, vol. 2, no. 4, Jul. 2021, doi: 10.1007/s42979-021-00652-2.
- [10] K. Sylejmani, E. Gashi, and A. Ymeri, "Simulated annealing with penalization for university course timetabling," *Journal of Scheduling*, vol. 26, no. 5, pp. 497–517, Oct. 2023, doi: 10.1007/s10951-022-00747-5.
- [11] M. L. Pinedo, *Scheduling*, 6th ed. Cham: Springer International Publishing, 2022. doi: 10.1007/978-3-031-05921-6.
- [12] Suyanto, *Algoritma optimasi deterministik atau probabilitik*. Graha Ilmu, 2010.
- [13] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.

## LAMPIRAN

PENJADWALAN

Upload Dataset

*Lampiran 1 Halaman Beranda*

PENJADWALAN

File.csv

Proses Penjadwalan

*Lampiran 2 Halaman Proses*

## PENJADWALAN

HASIL PENJADWALAN								
Hari	Kode	Mata Kuliah	Kelas	SKS	Jam	Semester	Dosen	Ruang
Minggu	50000	Algoritma Pemrograman	E	3	1,2	3	Dr., Ardiyansyah, S.T., M.Cs	4.1.5.67
Minggu	50000	Algoritma Pemrograman	E	3	1,2	3	Dr., Ardiyansyah, S.T., M.Cs	4.1.5.67
Minggu	50000	Algoritma Pemrograman	E	3	1,2	3	Dr., Ardiyansyah, S.T., M.Cs	4.1.5.67
Minggu	50000	Algoritma Pemrograman	E	3	1,2	3	Dr., Ardiyansyah, S.T., M.Cs	4.1.5.67
Minggu	50000	Algoritma Pemrograman	E	3	1,2	3	Dr., Ardiyansyah, S.T., M.Cs	4.1.5.67

## Lampiran 3 Halaman Hasil

Semester 1						
No	Kode MK	Nama Mata Kuliah	Sifat	Sem	SKS	Kode MK Prasyarat
1	211810230	Dasar Pemrograman	Wajib	1	3	
2	211810331	Dasar Sistem Komputer	Wajib	1	3	
3	211810430	Kalkulus Informatika	Wajib	1	3	
4	211810531	Logika Informatika	Wajib	1	3	
5	211810630	Manajemen Data dan Informasi	Wajib	1	3	
6	211810720	Pancasila	Wajib	1	2	
7	211810811	Praktikum Dasar Pemrograman	Wajib	1	1	
	Jumlah sks Semester				20	

## Lampiran 4 Tabel Dataset