

# OPTIMASI PENJADWALAN PEKERJA HARIAN DENGAN MENGGUNAKAN ALGORITMA EVOLUSI

*by abel kaban*

---

**Submission date:** 15-Nov-2019 09:25AM (UTC+0700)

**Submission ID:** 1214128635

**File name:** ADWALAN\_PEKERJA\_HARIAN\_DENGAN\_MENGGUNAKAN\_ALGORITMA\_EVOLUSI.docx (1.76M)

**Word count:** 4429

**Character count:** 28455

## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

PT Pratama Inti Flower merupakan anak perusahaan dari PT Gudang Garam Tbk dimana PT Gudang Garam memegang penuh seluruh sahamnya. PT Pratama Inti Flower bergerak di dalam manajemen kebun dan area taman. PT Pratama Inti Flower ini didirikan untuk terus melakukan pengembangan terhadap manajemen kebun dan area taman untuk mendapatkan hasil produksi yang baik.

Dalam tugasnya, PT Pratama Inti Flower memiliki beberapa Tugas yaitu melakukan perencanaan seluruh perawatan bunga dan area hijau di dalam pabrik PT Gudang Garam Tbk, untuk menyediakan seluruh kebutuhan pupuk dan material tanaman sehingga daerah pabrik terlihat bersih dan terawat. PT Pratama Inti Flower memiliki tugas memanajemen kebun seluruh area.

Setelah pekerjaan dilakukan oleh satuan kerja, kemudian akan dilakukan laporan pengeluaran barang-barang, material dan jasa oleh satuan kerja. Sehingga akan kerap terjadi perbedaan dalam nilai pengeluaran. Untuk meminimalisir terjadinya perbedaan pengeluaran, maka perusahaan perlu melaksanakan evaluasi untuk mengetahui berapa keuntungan buah melon berdasarkan jumlah hari kerja.

Untuk memenuhi kebutuhan tersebut, penulis akan melakukan optimasi perawatan jadwal kebun selama satu bulan sehingga diharapkan dapat menemukan solusi yang paling mendekati untuk jadwal kerja pada kebun sehingga adil saat pembagian kerja dan libur.

Algoritma yang digunakan adalah algoritma genetika yang merupakan salah satu metode optimasi untuk mencari solusi yang mendekati nilai optimal. Algoritma genetika digunakan agar hasil dari data yang telah diberikan lebih cepat di temukan hasilnya.

### 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah yang diambil adalah bagaimana optimasi perawatan jadwal kebun selama satu bulan?

20

### 1.3 Tujuan

Berdasarkan masalah yang telah dirumuskan, maka tujuan dari penelitian ini yaitu untuk mengetahui hasil optimasi perawatan kebun selama satu bulan.

## 1.4 Manfaat

Manfaat yang diperoleh dari pelaksanaan Praktik Kerja Lapangan ini yaitu:

### 1.4.1 Manfaat Bagi Perusahaan:

1. PT Pratama Inti Flower dapat mengetahui kemampuan mahasiswa dan kualitas Pendidikan program studi Teknik Informatika Universitas Brawijaya.
2. Sebagai alat untuk memudahkan perusahaan untuk mengoptimasi perawatan jadwal kebun selama satu bulan.

### 1.4.2 Manfaat Bagi Penulis:

1. Memberikan pengalaman praktik kerja Lapangan
2. Meningkatkan kemampuan mahasiswa dalam keterampilan dan wawasan, baik dalam hardskill maupun softskill.
3. Memberikan gambaran kepada mahasiswa bagaimana lingkungan pekerjaan.

## 1.5 Batasan Masalah

Batasan masalah yang digunakan sebagai pedoman dalam melakukan penelitian ini adalah sebagai berikut:

1. Data yang digunakan yaitu berupa data pekerja selama satu bulan.
2. Metode untuk optimasi penjadwalan.
3. Jumlah Variabel yang digunakan pada penelitian ini adalah jumlah pekerja
4. Bahasa pemrograman yang digunakan adalah python.

## 1.6 Waktu dan Tempat Pelaksanaan

Praktik Kerja Lapangan (PKL) ini dilaksanakan pada tanggal 17 Juli – 17 September 2019 di kantor PT Pratama Inti Flower yang berlokasi di Dsn Terong Dowo, Desa sumber suko, Kec. Prigen, Kab Pasuruan, Jawa Timur.

## 1.7 Sistematika Penulisan

Sistematika penyusunan laporan dilakukan untuk memberikan gambaran dan uraian terkait laporan Praktik Kerja Lapangan(PKL) secara garis besar yang terdiri dari beberapa bab yaitu :

### BAB I PENDAHULUAN

Bab 1 Pendahuluan membahas latar belakang, rumusan masalah, tujuan, manfaat dan batasan masalah mengenai optimasi perawatan kebun selama satu bulan.



## BAB II PROFIL INSTANSI

37

Bab 2 Profil Instansi ini menjelaskan sejarah berdirinya perusahaan, lokasi perusahaan, visi misi perusahaan, struktur organisasi perusahaan, logo perusahaan, kebijakan mutu, tugas dan fungsi, dan fasilitas perusahaan.

## BAB III LANDASAN KEPUSTAKAAN

Bab 3 landasan kepustakaan ini menjelaskan teori-teori yang akan digunakan untuk merancang dan mengimplementasikan sistem dari literature-literature yang telah ada dan pengujian yang telah dilakukan oleh para ahli pada penelitian sebelumnya.

## BAB IV METODOLOGI

Bab 4 metodologi ini menjelaskan tentang metodologi penelitian yang digunakan, yaitu: studi pustaka, pengumpulan data, perancangan sistem, implementasi sistem, pengujian system.

## BAB V HASIL DAN PEMBAHASAN

Bab5 hasil dan pembahasan ini membahas tentang perancangan dan implementasi sistem, antara lain metode yang digunakan, antar muka sistem, serta optimasi jadwal kebutuhan selama satu bulan.

## BAB VI PENUTUP

Bab ini berisi penjelasan tentang kesimpulan seluruh laporan PKL dan juga saran yang dapat dijadikan sebagai koreksi dan pedoman dalam pengembangan sistem kedepannya.

## 1 BAB 2 PROFIL OBJEK PKL

Bab ini menjelaskan tentang profil perusahaan PT Pratama Inti Flower yang berada di kota Pasuruan. Pada bab ini terdiri dari beberapa bagian yaitu, sejarah, lokasi, rantai bisnis perusahaan dan struktur organisasi.

### 2.1 Sejarah Singkat Perusahaan

PT Pratama Inti Flower merupakan anak perusahaan dari PT Gudang Garam Indonesia yang dikenal sebagai salah satu perusahaan industri rokok terkemuka di Indonesia. PT Pratama Inti Flower terletak di daerah Pasuruan, Jawa Timur tepatnya di daerah Gempol. PT Pratama Inti Flower Sesuai dengan Akta Pendirian No c2-12.551 HT 01.01 Th 1995, dengan kegiatan bisnis utama di bidang pertanian, antara lain perkebunan mangga, melon, semangka , bunga, pembibitan jagung hibrida, jasa management kebun dan jasa pembuatan land scaping/taman.

Kehadiran PT Pratama Inti Flower diharapkan membantu management area kebun dan taman di area Direktorat Produksi Gempol di Pasuruan Jawa Timur. Pt Pratama Inti Flower bertanggung jawab agar seluruh daerah area kebun dapat di manajemen dengan sebaik-baiknya dan perawatan area hijau pabrik terlihat bersih dan terawat.

### 2.2 Lokasi

PT Pratama Inti Flower terletak di di Dusun Terong Dowo, Desa Sumbersuko, Kecamatan Prigen, Kabupaten Pasuruan, Jawa Timur.

### 2.3 Rantai Bisnis Perusahaan

Total area produksi/kebun adalah ± 50 ha dan management kebun adalah ± 200 ha.

Kegiatan produksi kebun dibagi dalam beberapa area antara lain :

**Tabel 2.1 Kegiatan produksi kebun**

No.	Area Kebun	Ha
1	Perkebunan Mangga	3 Ha
2	Perkebunan Melon dan Semangka	5 Ha
3	Perkebunan Sirsak	20 Ha
4	Perkebunan Kelengkeng	30 Ha
5	Perkebunan Bibit	5 Ha
6	Garden Management Area	200 Ha

## 2.4 Vendor-Client Perusahaan

Didalam menjalankan unit bisnis Garden Management, PT Pratama Inti Flower memiliki Client-Vendor yaitu:

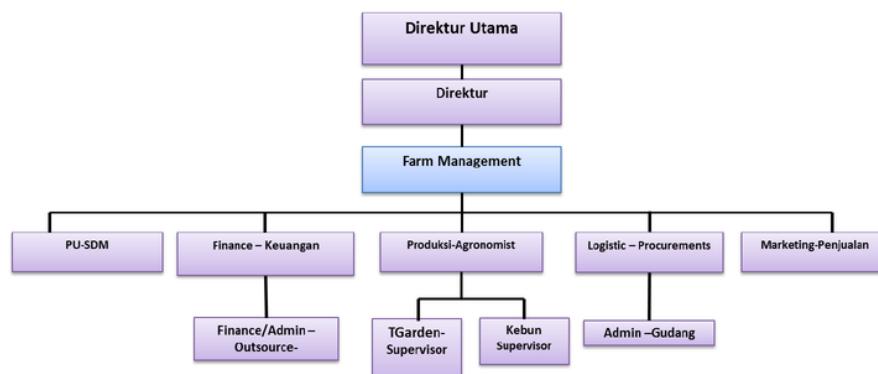
Tabel 2.2 Client-Vendor

No.	Area Garden Management	Ha
1	PT Gudang Garam tbk	200 Ha
2	PT. Dhanistha Surya Pertiwi	50 Ha

Tugas dan tanggung jawab perusahaan kepada Vendor adalah melakukan manajemen area kebun dan taman, yang melingkupi perencanaan seluruh perawatan tanaman dan area hijau pabrik PT Gudang Garam Tbk, menyediakan seluruh kebutuhan pupuk dan material tanaman sehingga area pabrik terlihat bersih dan terawat. Di perkebunan PT Dhanistha Surya Pertiwi, PT Pratama Inti Flower bertanggung jawab untuk menajemen kebun seluruh area perkebunan tanaman Sirsak dan Kelengkeng.

## 2.5 Struktur Organisasi

Untuk menjalankan kegiatan bisnis perusahaan, PT Pratama Inti Flower memperkerjakan 200 karyawan yang setiap hari bertugas di masing-masing unit kerja dan area perkebunan.



Gambar 2.1 Struktur Organisasi PT. Pratama Inti Flower

## BAB 3 METODOLOGI

### 3.1 Kajian Pustaka

Algoritma genetika (Genetic Algorithms, GAs), merupakan tipe EAs yang paling popular dan banyak diterapkan pada masalah-masalah kompleks. awalnya banyak yang menggunakan representasi string biner tapi kemudian berkembang dengan menggunakan vektor bilangan integer dan pecahan (real). Pembangkitkan solusi baru banyak mengandalkan proses tukar silang (crossover). Mutasi biasanya dipakai untuk operator tambahan untuk menjaga keragaman populasi. Susyanto, (2011) algoritma genetika banyak digunakan pada bidang soft computing dalam mendapatkan nilai-nilai parameter yang optimal pada jaringan syaraf tiruan atau system fuzzy. Algoritma genetika diilhami oleh ilmu genetika, karena itu istilah yang digunakan dalam algoritma genetika banyak diadopsi dari ilmu tersebut. Penelitian mengenai Optimasi dengan menggunakan Metode Algoritma Genetika sudah banyak di lakukan dan di aplikasikan dalam berbagai hal antara lain dalam Optimasi Pendistribusian pupuk, Optimasi Pembelian Barang, Optimasi Rute Terbaik, dsb.

(Aditya & Mahmudy, 2016) menjelaskan bagaimana memaksimalkan keuntungan serta mengurangi biaya yang tidak diperlukan untuk mencegah kerugian dalam bidang konveksi baju dengan optimasi produksi(persediaan baju) yang baik menggunakan Algoritma Genetika. Dari hasil pengujian yang mencakup pengujian ukuran populasi (popSize), pengujian jumlah generasi (iterasi) dan pengujian kombinasi crossover rate (cr) dan mutation rate (mr). Didapatkan parameter terbaik dengan ukuran populasi sebesar 250, jumlah generasi sebesar 150, kombinasi crossover rate 0.4 dan mutation rate 0.8. Rincian optimasi berupa jumlah produksi selama 8 minggu yaitu [341.88 , 414.36 , 571.97 , 483.63 , 313.58 , 435.56 , 361.88 , 410.52] dengan total biaya simpan sebesar 20.000, biaya lembur sebesar 120.000 dan kerugian sebesar 5.450, sehingga didapatkan nilai fitness sebesar 0.6875.

(Permadi & Subanar, 2010) melakukan Optimasi pada Penjadwalan Tebangan Hutan yang bertujuan memaksimalkan volume kayu yang ditebang pada setiap periode tebangan dengan penentuan petak petak tebangan menggunakan Algoritma Evolusi. Berdasarkan penelitian dan perancangan yang dilakukan penjadwalan tebangan hutan dengan variasi nilai parameter Pc dan Pm lebih berpengaruh dibandingkan dengan variasi ukuran populasi terhadap keoptimalan yang dihasilkan.

(Fachrurrazi, 2013) melakukan implementasi persoalan Optimasi Rute Pendistribusian Pupuk dengan menggunakan Algoritma Evolusi pada PT Pupuk Iskandar Muda. Penerapan Optimasi menggunakan Algoritma Genetika menghasilkan rute terpendek yang dapat dipakai oleh PT Pupuk Iskandar Muda.

(Putri & Arif, 2018) menjelaskan tentang banyaknya emisi gas rumah kaca (GRK) terutama gas metana ( $\text{CH}_4$ ) pada sawah konvensional sehingga diperlukannya pengelolaan air tanah yang efektif. Penelitian Bertujuan menentukan ketinggian air dan kelembapan tanah yang optimal disetiap tahap pertumbuhan tanaman dengan Algoritma Genetika.

Dari beberapa jurnal tersebut, peneliti memilih penelitian yang dilakukan oleh (Aditya & Mahmudy, 2016) yang berjudul “OPTIMASI PERSEDIAAN BAJU MENGGUNAKAN ALGORITMA GENETIKA” sebagai rujukan utama dalam melakukan pembuatan Penelitian ini.

**Tabel 3.1 Ringkasan Penelitian Kasus Optimasi Menggunakan Metode Algoritma Genetika**

Penelitian	Kasus Optimasi	Metode
(Permadji & Subanar, 2010)	Penjadwalan Tebangan Hutan	Algoritma Evolusi
(Fachrurrazi, 2013)	Konsumsi Energi Primer Indonesia	Algoritma Evolusi
(Aditya & Mahmudy, 2016)	Persediaan Baju	Algoritma Evolusi
(Putri & Arif, 2018)	Optimasi Pengelolaan Air Lahan Padi Sawah Rendah Emisi Gas Metana	Algoritma Evolusi

### 3.2 Dasar Teori

5

#### 3.2.1 Pengertian Optimasi

Optimasi adalah fokus dalam ilmu Matematika yang mempelajari cara mendapatkan nilai minimum atau maksimum secara sistematis dari suatu fungsi, peluang maupun pencarian dalam berbagai macam kasus (Fachrurrazi, 2013). Optimasi juga dapat diartikan sebagai cara untuk mengoptimalkan sesuatu yang selama ini belum optimal. Optimasi sering juga digambarkan seperti “Prinsip Ekonomi” dimana kegiatan usaha dilakukan seminimum mungkin (Low Effort) tetapi menghasilkan keuntungan semaksimum mungkin (High Benefit).

34

#### 3.2.2 Pengertian Optimasi

Algoritma Genetika (Genetic Algorithms) merupakan suatu metode yang menyediakan alternatif untuk teknik pencarian tradisional dengan mengadaptasi mekanisme yang ditemukan dalam dunia genetika (Rooker, 1991).

Kelebihan GAs sebagai metode optimasi adalah sebagai berikut:

- GAs merupakan algoritma yang berbasis populasi yang memungkinkan digunakan pada optimasi masalah dengan ruang pencarian (search space) yang sangat luas dan kompleks. Properti ini juga memungkinkan GAs untuk melompat keluar dari daerah optimum lokal (Gen & Cheng 1997).

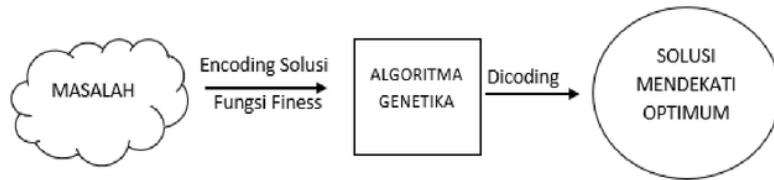
- Individu yang ada pada populasi bisa diletakkan pada beberapa sub-populasi yang diproses pada sejumlah komputer secara paralel. Hal ini bisa mengurangi waktu komputasi pada masalah yang sangat kompleks (Defersha & Chen 2010; Qi, Burns & Harrison 2000). Penggunaan sub-populasi juga bisa dilakukan pada hanya satu komputer untuk menjaga keragaman populasi dan meningkatkan kualitas hasil pencarian (Mahmudy 2009).

- GAs menghasilkan himpunan solusi optimal yang sangat berguna pada penyelesaian masalah dengan banyak obyektif (Mahmudy & Rahman 2011).

- GAs dapat digunakan untuk menyelesaikan masalah yang kompleks dengan banyak variabel. Variabel tersebut bisa kontinyu, diskrit atau campuran keduanya (Haupt & Haupt 2004).

- GAs menggunakan chromosome untuk mengkodekan solusi sehingga bisa melakukan pencarian tanpa memperhatikan informasi derivatif yang spesifik dari masalah yang diselesaikan (Gen & Cheng 1997; Haupt & Haupt 2004).

Solusi dari suatu masalah harus dipetakan (encoding) menjadi string chromosome. String chromosome ini tersusun atas sejumlah gen yang menggambarkan variabel-variabel keputusan yang digunakan dalam solusi. Representasi string chromosome beserta fungsi fitness untuk menilai seberapa bagus sebuah chromosome (untuk menjadi solusi yang layak) dimasukkan ke algoritma genetika. Dalam banyak kasus, bagaimana merepresentasikan sebuah solusi menjadi chromosome sangat menentukan kualitas dari solusi yang dihasilkan (Mahmudy, Marian & Luong 2012a). Dengan menirukan proses genetika dan seleksi alami maka algoritma genetika akan menghasilkan chromosome 'terbaik' setelah melewati sekian generasi. Chromosome 'terbaik' ini harus diuraikan (decoding) menjadi sebuah solusi yang diharapkan mendekati optimum.



Gambar 3.1 Struktur Algoritma Genetika

### 3.3 Analisa Penjadwalan Shift Pekerja

Penjadwalan merupakan sebuah kegiatan yang perlu dalam proses produksi maupun pekerjaan suatu proyek. Penjadwalan dibutuhkan sebagai dasar untuk

mengatur sumberdaya pabrik seperti sumberdaya manusia, maupun peralatan. Contoh masalah penjadwalan adalah jadwal mengajar dosen, penjadwalan ujian, dan penjadwalan pekerja.

Penjadwalan pekerja adalah proses mengatur waktu kerja dan waktu libur di sebuah pabrik/perusahaan. Dalam proses penjadwalan komponen yang digunakan yaitu pekerja, hari, dan shift.

Penjadwalan shift pekerja memiliki tujuan memberikan jadwal harian yang dibagi menjadi beberapa shift selama 30 hari pada masing-masing pekerja.

PT Pratama Inti Flower memiliki 13 orang pekerja adapun pekerjaan dibagi menjadi 3 shift, shift pagi di mulai dari pukul 07.00 WIB Pagi sampai dengan pukul 15.00 WIB Sore, lalu shift siang dimulai dari pukul 15.00 WIB sampai pukul 23.00 WIB Malam, kemudian dilanjutkan shift malam yang dimulai dari pukul 23.00 WIB sampai pukul 07.00 WIB Pagi.

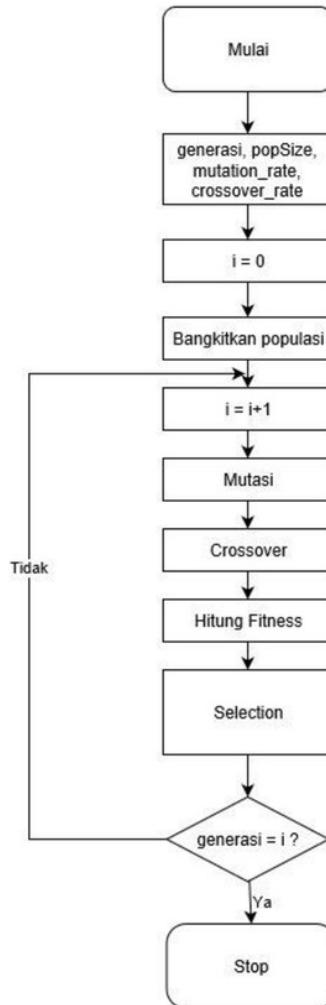
1

## BAB 4 METODOLOGI PELAKSANAAN PKL

Pada bab ini membahas mengenai metode yang digunakan dalam penjadwalan pekerja serta langkah-langkah dalam mengimplementasikan metode pada penjadwalan pekerja. Langkah penelitian digambarkan melalui diagram pada gambar dibawah.

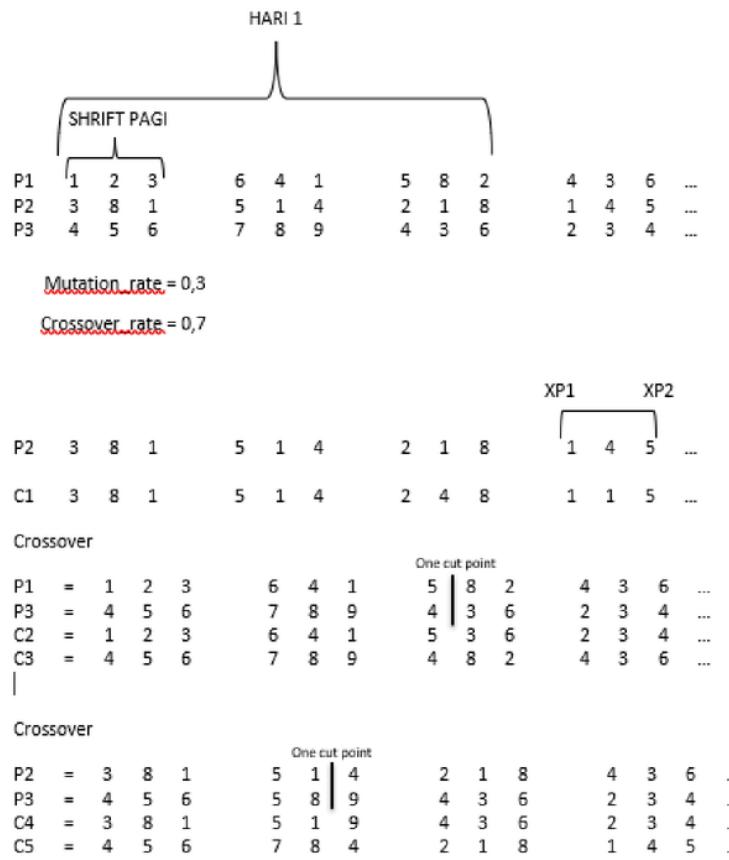
1

### 4.1 Diagram Alir Metode



Gambar 4.1 Diagram Alir Metode

#### 4.1.1 Manualisasi Data



1  
**Gambar 4.2 Proses Manualisasi Data**

Fitness

P1 = 0,411522

P2 = 0,409837

P3 = 0,414937

C1 = 0,419937 → 3

C2 = 0,423728 → 1

C3 = 0,421940 → 2

C4 = 0,40983

C5 = 0,40983

3. Individu Terbaik yang akan dilanjutkan pada generasi selanjutnya adalah

C2 =	1	2	3	6	4	1	5	3	6	2	3	4	...
C3 =	4	5	6	7	8	9	4	8	2	4	3	6	...
C1 =	3	8	1	5	1	4	2	4	8	1	1	5	...

Gambar 4.3 Proses Manualisasi Data

17

Pada gambar 4.2 dan 4.3 menunjukkan manualisasi data menggunakan popsize dan crossover\_rate = 0,7 dan mutation\_rate = 0,3

27

## 4.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data pekerja harian yang bekerja pada kebun PT Pratama Inti Flower Tahun 2019.

## 4.3 Perancangan

3

Proses perancangan algoritma genetika untuk optimasi penjadwalan pekerja adalah sebagai berikut :

1. Implementasi parameter awal:

Parameter pada algoritma genetika mencakup jumlah generasi, ukuran populasi, crossover\_rate, mutation\_rate.

2. Bangkitkan populasi awal

Populasi awal berjumlah populasi yang sudah ditentukan

3. Membentuk populasi baru dengan langkah :

- Melakukan proses crossover pada induk yang terpilih berdasarkan crossover rate untuk mendapatkan anak (offspring) dengan metode one cut point crossover.

- Melakukan proses mutasi pada induk yang dipilih berdasarkan mutation rate untuk mendapatkan anak (*offspring*) dengan reciprocal exchange mutation.
- Menghitung nilai fitness dalam masing-masing individu pada populasi
- Melakukan seleksi elitism untuk memilih individu sebanyak jumlah populasi awal dari gabungan induk dan anak untuk dijadikan populasi pada generasi selanjutnya.

3

Jika telah sampai pada generasi yang ditentukan maka iterasi akan berhenti dan solusi yang terbaik adalah populasi pada generasi tersebut.

#### 4.4 Alur Penyelesaian Masalah

Proses penjadwalan dilakukan menggunakan 13 pekerja didalam satu kebun. jadwal disusun untuk 30 hari, yang akan dibagi menjadi 3 shift terdiri dari pagi, siang, dan malam.

##### 4.4.1 Representasi Kromosom

Kromosom berisi dari 270 angka, pada setiap kromosom berisi angka 1 sampai 13, setiap 9 digit menandakan 1 hari, 3 digit pertama menandakan shift pertama, 3 digit kedua menandakan shift siang, dan 3 digit terakhir menandakan shift malam. Kemudian digit tersebut akan diulang sampai dengan hari ke 30.

##### 4.4.2 Perhitungan Penalty

Untuk menghitung nilai fitness diperlukan jumlah pelanggaran atau penalty yang muncul pada kromosom setiap dilakukannya pelanggaran nilai penalty akan bertambah 1. Daftar pelanggaran dapat dilihat pada table berikut:

Tabel 4.1 Daftar Pelanggaran

Penalty	Keterangan
P1	1 pekerja terdaftar pada hari/shift yang sama
P2	Pekerja memiliki jumlah libur yang lebih dari yang telah ditentukan
P3	Pekerja yang telah shift malam akan bekerja lagi pada shift pagi keesokan harinya

##### 4.4.3 Perhitungan Fitness

Untuk melakukan penilaian terhadap masing-masing kromosom dilakukan perhitungan nilai fitness. Nilai fitness menunjukkan kualitas suatu kromosom,

3

yaitu semakin mendekati angka 1 maka semakin optimum solusi yang diberikan. Perhitungan fitness menggunakan rumus sebagai berikut:

$$Fitness = \frac{100}{(100 + Total\ Penalty)}$$

#### 4.4.4 Mutasi

Metode mutasi yang digunakan adalah *reciprocal exchange mutation*. Metode tersebut bekerja dengan memilih dua posisi (exchange point / xp) secara random lalu menukar nilai kedua posisi tersebut.

#### 4.4.5 Crossover

Metode crossover yang digunakan adalah one-cut-point crossover yang bekerja dengan cara memilih dua induk secara random dan menentukan titik potong secara acak lalu memasang bagian yang terpotong tersebut pada induk lain yang telah terpilih.

#### 4.4.6 Seleksi

Pada makalah ini akan dilakukan 2 proses pengujian yaitu:

1. Pengujian Popsize
2. Uji konvergent

### 4.5 Pengujian

Pengujian berfungsi untuk mengevaluasi keefektifan penerapan algoritma yang telah dibangun. Pada penelitian ini akan dilakukan 3 proses pengujian, yaitu:

1. Pengujian popSize, adalah pengujian untuk mencari nilai populasi yang efektif pada algoritma yang digunakan. Populasi yang dianggap efektif memiliki nilai rata-rata fitness tinggi.
2. Uji konvergen, adalah pengujian untuk mencari nilai generasi yang efektif pada algoritma yang digunakan. Nilai generasi yang efektif adalah nilai perhitungan yang sudah tidak terlalu signifikan lagi perkembangan individunya, setelah melakukan beberapa iterasi.
3. Pengujian nilai mutation rate dan crossover rate, adalah pengujian yang mencari nilai mutation rate dan crossover rate yang paling baik, berdasarkan nilai rata-rata fitness tertinggi yang dihasilkan

## 1 BAB 5 HASIL DAN PEMBAHASAN

### 1 5.1 Implementasi Algoritma

Pada implementasi algoritma, menjelaskan tentang pengimplementasian algoritma pada kode program yang telah dilakukan perhitungan manual.

#### 5.1.1 Algoritma Genetika

Sebelum melakukan perhitungan regresi linier, data excel yang telah dinormalisasi diinputkan. Data tersebut berupa variabel terikat y yang merupakan nilai Berita Acara Serah Terima (BAST) dan variabel bebas x yang merupakan nilai Surat Pesanan (SP), kemudian menghitung nilai  $\alpha$  dan  $b$  dari data. Lalu melakukan perhitungan prediksi nilai BAST menggunakan regresi linier yang diimplementasikan melalui kode program table 5.1 di bawah ini.

```
31
1 import random
2 import copy
3 import matplotlib.pyplot as plt
4 class kromosom:
5     def __init__(self, n=1, individu = []):
6         pekerja = 13
7         if n == 1:
8             individu = []
9             for i in range(30):
10                 harian = []
11                 for j in range(3):
12                     harian.append(random.sample(range(1,pekerja+1),3))
13                     individu += [harian]
14             self.individu = individu
15             self.fitness = -1
16     gene = []
17     pop = []
18     offspring = []
19     population = 100
20     def fitness(pop):
21         for individu in pop:
22             n = individu.individu
23             penalty = count_penalty(n)
24             individu.fitness = 100/(100+penalty)
25     def count_penalty(individu):
26         penalty = 0
27         libur =
28         [[1,2,3,4,5,6,7,8,9,10,11,12,13],[0,0,0,0,0,0,0,0,0,0,0,0,0]]
29         #1 Hari 2x kerja dan hitung hari libur
30         for day in range(len(individu)):
31             seen = set()
32             for shift in range(len(individu[day])):
33                 for i in range(len(individu[day][shift])):
34                     if individu[day][shift][i] in seen:
35                         penalty = penalty + 1
36                     else:
37                         seen.add(individu[day][shift][i])
38             for j in range(13):
39                 if libur[0][j] not in seen: 16
40                     libur[1][j-1] = libur[1][j-1]+1
41             for i in range(13):
42                 if libur[1][i] > 8:
43                     penalty = penalty + (libur[1][i] - 8)
```

```

44     #shift malam tidak boleh dilanjut shift pagi
45     for day in range(len(individu)-1):
46         seen = set()
47         for i in range(len(individu[day][2])):
48             seen.add(individu[day][2][i])
49             for j in range(len(individu[day][1])):
50                 if individu[day+1][0][j] in seen :
51                     penalty = penalty +1
52     return penalty
53 def selection(pop):
54     pop = sorted(pop, key=lambda individu: individu.fitness,
55     reverse=True)
56     pop = pop[:100]
57     return pop
58 def mutasi(gene, mutation_rate):
59     #mutation_rate = 0.6
60     for individu in gene :
61         tes = copy.deepcopy(individu)
62         if random.random() < mutation_rate :
63             tempa= [random.randint(0,29),random.randint(0,2),random.randint(0,2)]
64             tempb=[random.randint(0,29),random.randint(0,2),random.randint(0,2)]
65             temp = tes[tempa[0]][tempa[1]][tempa[2]]
66             tes[tempa[0]][tempa[1]][tempa[2]] =
67             tes[tempb[0]][tempb[1]][tempb[2]]
68             tes[tempb[0]][tempb[1]][tempb[2]] = temp
69             tes[tempb[0]][tempb[1]][tempb[2]] = temp
70             offspring.append(tes)
71     #pop.append(kromosom(0, offspring))
72     #gene.append(offspring)
73     def crossover(gene, crossover_rate):
74         #crossover_rate = 0.4
75         for i in range(len(gene)//2) :
76             if random.random() < crossover_rate :
77                 parent1 = random.choice(gene)
78                 parent2 = random.choice(gene)
79                 child1 = copy.deepcopy(parent1)
80                 child2 = copy.deepcopy(parent2)
81                 split1 = random.randint(0,29)
82                 split2 = random.randint(0,2)
83                 split3 = random.randint(0,2)
84                 child1 = parent1[0:split1] + [parent1[split1][0:split2] +
85                 [parent1[split1][split2][0:split3]+parent2[split1][split2][split3:3]]+
86                 + parent2[split1][split2+1:3]] + parent2[split1+1:30]
87                 child2 = parent2[0:split1] + [parent2[split1][0:split2] +
88                 [parent2[split1][split2][0:split3]+parent1[split1][split2][split3:3]]+
89                 + parent1[split1][split2+1:3]] + parent1[split1+1:30]
90                 offspring.append(child1)
91                 offspring.append(child2)
92     def combine(offspring) :
93         for i in range(len(offspring)):
94             pop.append(kromosom(0,offspring[i]))
95     #def count_avg(pop):
96     #    total = 0
97     #    for i in range(len(pop)):
98     #        total += pop[i].fitness
99     #    return total/len(pop)
100    def generate_individu(n):
101        for i in range(n) :
102            pop.append(kromosom())
103        return pop
104    def init():
105        gene.clear()
106        offspring.clear()
107

```

```

108     for i in range(0,len(pop)):
109         gene.append(pop[i].individu)
110 def main(n):
111     global pop
112     y = []
113     x7= []
114     mutation_rate = 0.6
115     crossover_rate = 0.4
116     for i in range(n):
117         print('Generasi : ' + str(i+1))
118         init()
119         mutasi(gene, mutation_rate)
120         crossover(gene, crossover_rate)
121         combine(offspring)
122         fitness(pop)
123         pop = selection(pop)
124         #favg = count_avg(pop)
125         print('Fitness Terbaik : '+ str(pop[0].fitness))
126         print('Rata-rata Fitness : '+ str(favg))
127         y += [pop[0].fitness]
128         for j in range(n):
129             x += [j+1]
130             plt.plot(x, y)
131             plt.xlabel('Generasi')
132             plt.ylabel('Fitness')
133             plt.show()

```

1

Penjelasan dari kode program tabel 5.1 adalah sebagai berikut :

1. Baris 1 – 3 mengimport package yang dibutuhkan
2. Baris 5-16 mendefenisikan kelas kromosom
3. Baris 18 menginisialisasi variabel gene
4. Baris 19 menginisialisasi variabel pop
5. Baris 20 menginisialisasikan variable offspring
6. Baris 21 menginisialisasi variable population dengan nilai 100
7. Baris 23- 27 mendefenisikan fungsi fitness dengan parameter masukan pop, fungsi ini berfungsi untuk memanggil fungsi count\_penalty lalu menghitung nilai fitness dengan rumus  $100/(100+penalty)$
8. Baris 31-59 mendefenisikan fungsi count\_penalty dengan parameter masukan individu, fungsi ini berfungsi untuk mengembalikan nilai penalty dari tiap individu. Perhitungan penalty dilakukan dengan cara menghitung jumlah pelanggaran yang dilakukan kromosom terhadap aturan yang telah dibuat
9. 61 – 66 mendefenisikan fungsi selection dengan parameter masukan pop dengan cara mengurutkan nilai fitness dari tiap individu dari nilai fitness terbesar sampai nilai fitness terkecil kemudian mengembalikan individu yang berperingkat 100 besar.
10. 69-79 mendefenisikan fungsi mutase dengan parameter masukan gene, mutation\_rate yang bekerja dengan cara memilih dua titik secara acak lalu menukar nilai dari kedua titik tersebut dengan variabel bantuan yaitu temp
11. 84 – 98 mendefenisikan variable crossover dengan parameter gene, crossover\_rate,dengan carakerja memilih dua individu secara acak dan ,memilih titik potong secara acak lalu menyilangkan kedua individu tersebut.
12. 101- 103 mendefenisikan fungsi combine dengan parameter offspring yang berfungsi menggabungkan offspring yang telah didapat dari hasil mutase dan crossover dengan populasi
13. 113-118 mendefenisikan fungsi generate\_individu dengan parameter n yang berfungsi membangkitka populasi awal
14. 120-125 mendefenisikan fungsi init yang berfungsi mereset variable gene dan offspring lalu mengisi variable gene dengan isi kromosom dari variable pop
15. 128-154 mendefenisikan fungsi main dengan parameter n yang berfungsi memanggil semua fungsi yang bertanggung jawab atas proses regenerasi dari populasi, proses tersebut diulangi sebanyak n kali. Lalu melakukan plotting nilai fitness pada setiap iterasi.

## 5.2 Hasil Pengujian

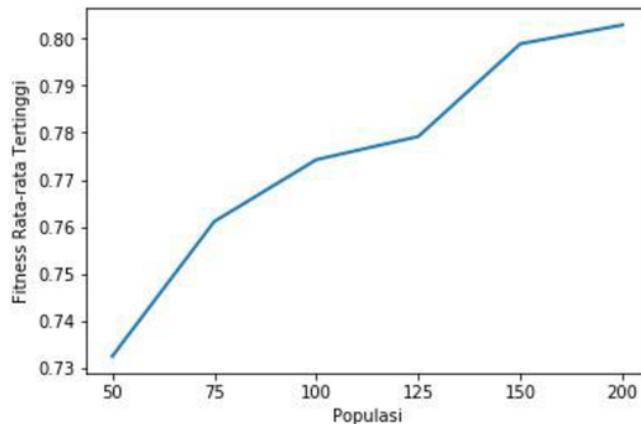
### 5.2.1 Hasil Pengujian Ukuran popsize

Pengujian ukuran populasi terhadap nilai fitness digunakan untuk mencari ukuran *popSize* terbaik, agar dapat menghasilkan nilai *popSize* yang efektif. Pada pengujian ini jumlah generasi yang digunakan adalah 100 lalu menggunakan *crossover\_rate* = 0.4, dan *mutation\_rate* = 0.6. Jumlah populasi yang diuji adalah enam, yaitu 50, 75, 100, 125, 150, 200. Pengujian ukuran populasi dilakukan sebanyak lima kali. Tabel hasil pengujian dapat dilihat pada Tabel 5.1

**Table 5.1 Tabel Pengujian Banyaknya Populasi Terhadap Banyaknya Fitness**

<i>popSize</i>	Rata-rata <i>fitness</i> tertinggi
50	0.7324947298471306
75	0.7611167582950404
100	0.7742435347772848
125	0.7791627261167252
150	0.7988525785181652
200	0.8028315146355547

Berdasarkan hasil pengujian pada Tabel 5.1 dapat disimpulkan semakin besar ukuran *popsizes* maka rata-rata fitness yang dihasilkan semakin besar.

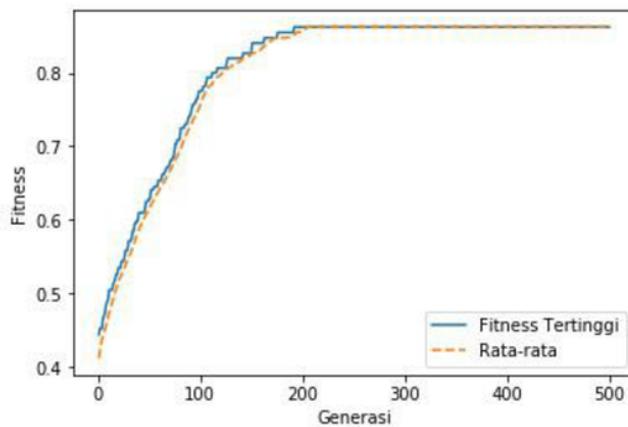


**Gambar 5.1 Grafik Ujicoba Populasi**

Grafik hubungan antara *popSize* dengan rata-rata *fitness* tertinggi dapat dilihat pada Gambar 5.1

### 5.2.2 Uji Konvergensi

Pengujian kedua pada penilitian ini adalah uji konvergensi. Pengujian ini bertujuan untuk mencari tahu pada generasi keberapa perkembangan sebuah individu tidak signifikan atau bahkan berhenti. Pengujian dilakukan dengan menggunakan `popSize = 100`, `crossover_rate = 0.4`, dan `mutation_rate = 0.6`. Grafik hasil pengujian dapat dilihat dari gambar 5.2.



Gambar 5.2 Grafik Nilai Fitness Terhadap Generasi

Pada grafik diatas dapat membuktikan semakin kecil jumlah generasi maka perubahan rata-rata fitness sangat tinggi namun semakin banyak generasi perubahan tersebut menjadi semakin tidak signifikan, bahkan setelah generasi ke 190 tidak terjadi perkembangan terhadap nilai fitness sehingga didapat generasi yang terbaik pada penilitian ini adalah 190.

### 5.2.3 Pengujian Kombinasi `crossover_rate` dan `mutation_rate`

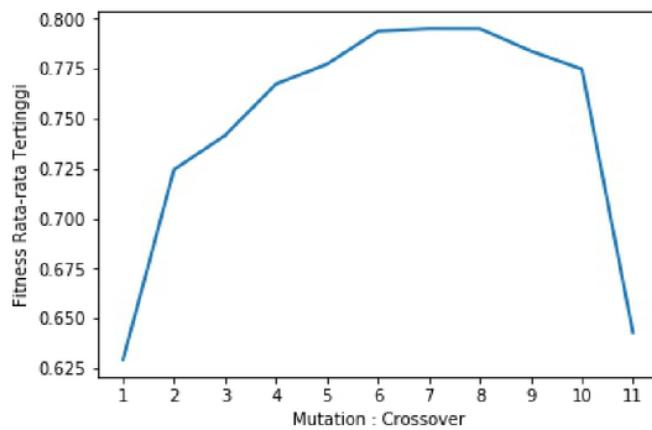
Pengujian ketiga yaitu pengujian nilai `crossover_rate` dan `mutation_rate` yang bertujuan mencari tingkat `crossover_rate` dan `mutation_rate` yang paling efektif dengan cara melakukan lima kali percobaan pada tiap kombinasi `crossover_rate` dan `mutation_rate`. Pengujian ini menggunakan nilai `popSize = 100` dan jumlah generasi adalah 100.

Tabel 5.1 Perhitungan Mutation\_rate dan Crossover\_rate

No	<code>mutation_rate</code>	<code>crossover_rate</code>	Rata-rata fitness tertinggi
1	1	0	0.6292052278217796
2	0.9	0.1	0.7244287810587304
3	0.8	0.2	0.7415166286864328
4	0.7	0.3	0.7672783719295347
5	0.6	0.4	0.777241314953841
6	0.5	0.5	0.7938093972311976

7	0.4	0.6	0.7950620997070488
8	0.3	0.7	0.7950206375412752
9	0.2	0.8	0.7837869424975724
10	0.1	0.9	0.7747479986963829
11	0	1	0.6427972570414389

1 Berdasarkan hasil pengujian pada Tabel 5.3 dapat disimpulkan, rata-rata nilai fitness tertinggi didapat pada nilai *mutation\_rate* = 0.4, *crossover\_rate* = 0.6 dan *mutation\_rate* = 0.3, *crossover\_rate* = 0.7 dengan nilai rata-rata fitness = 0.7950206375412752. Grafik hubungan antara kombinasi *mutation\_rate* dan *crossover\_rate* dengan Rata-rata Fitness tertinggi dapat dilihat pada Gambar 5.3



Gambar 5.3 Grafik Kombinasi nilai *crossover\_rate* dan *mutation\_rate* terhadap nilai *fitness*

## BAB 6 PENUTUP

Bab ini menjelaskan mengenai kesimpulan dan saran serta keberlanjutan laporan Praktik Kerja Lapangan (PKL)

24

### 6.1 Kesimpulan

Dari hasil pengujian diatas maka dapat dihasilkan kesimpulan sebagai berikut:

1. Dari tiga pengujian yang dilakukan, pada pengujian popSize dihasilkan ukuran populasi sebesar 200, pada uji konvergesi nilai fitness mulai stabil pada generasi 190 sehingga dapat disimpulkan generasi tersebut adalah yang terbaik. Pengujian kombinasi crossover rate (cr) dan mutation rate (mr). Menghasilkan rata-rata fitness tertinggi dengan nilai *mutation\_rate* = 0.4, *crossover\_rate* = 0.6 dan *mutation\_rate* = 0.3, *crossover\_rate* = 0.7.

### 6.2 Saran

1. Menambah kompleksitas dari Chromosome dengan cara menambahkan parameter
2. Menambah aturan penalty agar perhitungan nilai *fitness* menjadi semakin rumit

## DAFTAR PUSTAKA

- 1 Aditya, C., & Mahmudy, W. F. (2016). OPTIMASI PERSEDIAAN BAJU MENGGUNAKAN ALGORITMA GENETIKA. *Seminar Nasional Teknologi dan Rekayasa Informasi Tahun 2016*. Malang: SENTRIN.
- 12 Carwoto. (2007). Implementasi Algoritma Genetika untuk Optimasi Penempatan Kapasitor Shunt pada Penyalur Distribusi Tenaga Listrik. *Jurnal Teknologi Informasi DINAMIK*, 122-130.
- 1 Fachrurrazi, S. (2013). Penerapan Algoritma Genetika dalam Optimasi Pendistribusian Pupuk di PT Pupuk Iskandar Muda Aceh Utara. *II(1)*.
- 11 Hari, N. H., Putra, F. P., & Hamdlani. (2018). Optimasi Penjadwalan Menggunakan Metode Algoritma Genetika di Sekolah Menengah Kejuruan Annuqayah - Sumenep. *Jurnal Sistem Informasi*, 66-74.
- 19 Indrianingsih, Y. (2010). ALGORITMA GENETIK UNTUK MENYELESAIKAN MASALAH OPTIMASI FUNGSI BERKENDALA DENGAN PENGKODEAN BILANGAN BULAT. *ANGKASA*, 67-75.
- 21 Iryanto, & Ismantohadi, E. (2017). Optimasi Pemilihan Barang Dagangan bagi Pedagang Keliling dengan Algoritma Genetika. *Jurnal Teknologi Terapan*, 24-28.
- Jatmika, S. (2011). OPTIMASI KENAIKAN DAN PEMBAGIAN KELAS MENGGUNAKAN ALGORITMA GENETIKA. *Inovasi dalam Desain dan Teknologi* (pp. 178-189). Surabaya: IDeaTech.
- 15 Kurniawan, M. (2019). Optimasi Struktur Rangka Batang Menggunakan Metode Algoritma Genetika Dengan Kendala Tegangan Dan Probabilitas Kegagalan. *Saintis*, 15-23.
- 32 Novia, A., Tursina, & Priyanto, H. (n.d.). *Optimasi Tata Letak Barang Dengan Menggunakan Algoritma Genetika*. Pontianak.
- 25 Permadi, I., & Subanar. (2010). Penerapan Algoritma Genetika untuk Optimasi Penjadwalan Tebangan Hutan. *I(1)*.
- Putri, Y. M., & Arif, C. (2018). Penerapan Algoritma Genetika Untuk Optimasi Pengelolaan Air Lahan Padi Sawah Rendah Emisi Gas Metana (CH<sub>4</sub>). *III(3)*.
- 8 Rooker, T. (1991). Genetic Algorithms in Search, Optimization, and Machine Learning. *XII(1)*.
- 9 Suharjanto. (2006). IMPLEMENTASI ALGORITMA GENETIK PADA OPTIMASI BENTUK DAN UKURAN BUKAAN PADA BALOK BAJA PROFIL I DENGAN BUKAAN CELLULAR. *MEDIA KOMUNIKASI TEKNIK SIPIL*, 297-308.
- 1 Suhartono, E. (2015). OPTIMASI PENJADWALAN MATA KULIAH DENGAN ALGORITMA GENETIKA. *INFOKAM*, 132-146.

2

Smith, JE & Eiben, AE 2003, *Introduction to evolutionary computing*, Springer, London

## **DAFTAR LAMPIRAN**



Lampiran 1 Foto pekerja lahan sedang melakukan panen semangka



Lampiran 2 Foto Pekerja Sedang Melakukan Perawatan Kebun



**Lampiran 3 Foto Kegiatan Hasil Panen**

Lampiran 3 merupakan kegiatan hasil panen buah-buahan dari kebun



**Lampiran 4 Foto kegiatan Presentasi dari Hasil Pekerjaan**

Lampiran 4 merupakan kegiatan saat melakukan presentasi dari hasil yang telah dikerjakan selama magang.



**Lampiran 5 Foto Bersama Pembimbing PKL**

Lampiran 5 merupakan foto Bersama Bapak Erik Kharisman Milala selaku pembimbing selama melaksanakan PKL

ID_Pekerja	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
	Hari																														
1	P	M	M	M	S	P	M	M	M	P	S	S	S	S	M	M	S	M	M	S	M	S	M	S	P	P	P	P	P		
2	S	S	M	M	M	S	P	M	S	P	M	S	S	S	M	S	P	M	S	P	M	M	M	M	S	S	S	S	S		
3																															
4	M	M	S	M	P		S	P	P	P	P	P	P	P	P	M	S	P	S	P	M	S	P	M	S	P	P	P	P		
5	S	P		P	P	M	P	P	P	P	S	P	S	P	P	P	M	P	M	M	M	M	M	M	P	P	P	P	P		
6	P	S	P	P	M	M	P	S	M	P	M	S	P	M	S	M	M	P	P	M	P	P	M	P	P	M	P	P	P	P	
7	M	S	P	M	M	S	M	M	M	S	S	P	M	S	P	M	M	M	M	P	S	P	S	P	S	S	S	S	S	S	
8	M	M	P	S	P	P	P	S	P	P	S	P	P	S	P	M	S	S	P	S	P	S	P	S	M	S	S	S	S	S	
9	M	P	S	S	S	S	M	M	M	M	M	M	M	M	M	M	M	M	M	S	M	S	S	S	P	M	S	S	S	S	
10	P	S	S	P	P	S	M	S	S	S	M	M	M	M	M	M	M	M	M	P	M	S	P	M	P	P	P	P	P	P	
11	P	P	P	S	M	S	M	M	M	M	S	P	P	M	P	P	S	P	P	S	P	S	S	M	S	S	S	S	S	S	
12				S	S	M	P	P	S	S	P	P	S	S	P	P	S	P	P	S	P	M	S	S	S	P	M	S	S	S	
13	S	P	M	M	P	P	S	S	S	P	M	S	M	M	P	S	S	M	P	S	S	M	S	M	P	M	M	M	M	M	

P = shift Pagi

S = shift siang  
M = shift Malam

#### Lampiran 6 Decoding Jadwal Harian

Lampiran 6 adalah tabel hasil decoding dari penjadwalan selama 30 hari

# OPTIMASI PENJADWALAN PEKERJA HARIAN DENGAN MENGGUNAKAN ALGORITMA EVOLUSI

ORIGINALITY REPORT



PRIMARY SOURCES

- |   |   |     |
|---|---|-----|
| 1 | Submitted to Universitas Brawijaya<br>Student Paper         | 14% |
| 2 | <a href="#">wayanfm.lecture.ub.ac.id</a><br>Internet Source | 7%  |
| 3 | <a href="#">j-ptiik.ub.ac.id</a><br>Internet Source         | 1%  |
| 4 | Submitted to University of Bath<br>Student Paper            | 1%  |
| 5 | <a href="#">docplayer.info</a><br>Internet Source           | 1%  |
| 6 | <a href="#">es.scribd.com</a><br>Internet Source            | 1%  |
| 7 | Submitted to Unika Soegijapranata<br>Student Paper          | 1%  |
| 8 | <a href="#">www.scribd.com</a><br>Internet Source           | 1%  |
| 9 | <a href="#">ejournal.undip.ac.id</a>                        |     |

	Internet Source	<1 %
10	media.neliti.com Internet Source	<1 %
11	jurnal.uisu.ac.id Internet Source	<1 %
12	publikasiilmiah.unwahas.ac.id Internet Source	<1 %
13	www.wayanfm.lecture.ub.ac.id Internet Source	<1 %
14	journal.ipb.ac.id Internet Source	<1 %
15	journal.uir.ac.id Internet Source	<1 %
16	mail.python.org Internet Source	<1 %
17	Submitted to University of South Australia Student Paper	<1 %
18	eprints.umm.ac.id Internet Source	<1 %
19	eprints.uny.ac.id Internet Source	<1 %
20	text-id.123dok.com Internet Source	<1 %

---

21	jurnal.polindra.ac.id Internet Source	<1 %
22	Submitted to University of Portsmouth Student Paper	<1 %
23	id.123dok.com Internet Source	<1 %
24	Submitted to Fakultas Ekonomi Universitas Indonesia Student Paper	<1 %
25	Submitted to UIN Sultan Syarif Kasim Riau Student Paper	<1 %
26	Submitted to Imperial College of Science, Technology and Medicine Student Paper	<1 %
27	isharmuis.blogspot.com Internet Source	<1 %
28	bukafajarbukadunia.blogspot.com Internet Source	<1 %
29	pt.scribd.com Internet Source	<1 %
30	repository.unpas.ac.id Internet Source	<1 %
31	www.yeayee.com Internet Source	<1 %

---

32	slid/documents.org Internet Source	<1 %
33	adoc.tips Internet Source	<1 %
34	Submitted to STIKOM Surabaya Student Paper	<1 %
35	Submitted to Universitas Islam Indonesia Student Paper	<1 %
36	tjarieloker.blogspot.com Internet Source	<1 %
37	eprints.undip.ac.id Internet Source	<1 %
38	Submitted to University of Edinburgh Student Paper	<1 %
39	Submitted to RMIT University Student Paper	<1 %
40	kabar-loker.blogspot.com Internet Source	<1 %
41	megawatiputri22.blogspot.com Internet Source	<1 %
42	anzdoc.com Internet Source	<1 %
43	Submitted to University of Southampton Student Paper	<1 %

<1 %

---

---

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		