

**SISTEM KECERDASAN BUATAN PENGELOLAAN  
SUMBER DAYA DEVELOPER MENGGUNAKAN  
ALGORITMA GENETIKA**

**SKRIPSI**

Oleh :  
**MASFU'ULAJI EKA PUTRA**  
**17650050**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG  
2024**

**SISTEM KECERDASAN BUATAN PENGELOLAAN  
SUMBER DAYA DEVELOPER MENGGUNAKAN  
ALGORITMA GENETIKA**

**SKRIPSI**

Diajukan kepada:  
Universitas Islam Negeri Maulana Malik Ibrahim Malang  
Untuk memenuhi Salah Satu Persyaratan dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :  
**MASFU'UL AJI EKA PUTRA**  
NIM. 17650050

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2024**

**HALAMAN PERSETUJUAN**

**SISTEM KECERDASAN BUATAN PENGELOLAAN  
SUMBER DAYA DEVELOPER MENGGUNAKAN  
ALGORITMA GENETIKA**

**SKRIPSI**

**Oleh :  
MASFU'ULAJI EKA PUTRA  
NIM. 17650050**

Telah Diperiksa dan Disetujui untuk Diuji:  
Tanggal: 13 juni 2024

Pembimbing I,



Okta Qomaruddin Aziz, M.Kom  
NIP. 19911019 201903 1 013

Pembimbing II,



Dr. H. M. Imamudin Lc, MA  
NIP. 19740602 200901 1 010

Mengetahui,  
Ketua Program Studi Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Fachrul Kurniawan, M.MT, IPM  
NIP. 19771020 200912 1 001

## HALAMAN PENGESAHAN

### SISTEM KECERDASAN BUATAN PENGELOLAAN SUMBER DAYA MENGGUNAKAN ALGORITMA GENETIKA

#### SKRIPSI

Oleh :  
**MASFU'UL AJI EKA PUTRA**  
NIM. 17650050

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer ( S.Kom )  
Tanggal: 13 Juni 2024

#### Susunan Dewan Penguji

Ketua Penguji : Fajar Rohman Hariri, M. Kom  
NIP. 19890515 201801 1 001

(  )

Anggota Penguji I : Khadijah F.H. Holle, M.Kom  
NIP. 19900626 202203 2 002

(  )

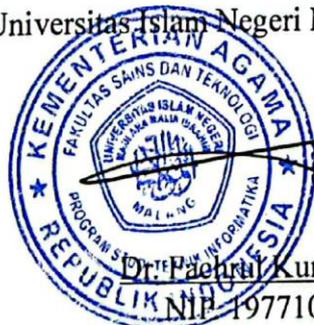
Anggota Penguji II : Okta Qomaruddin Aziz, M.Kom  
NIP. 19911019 201903 1 013

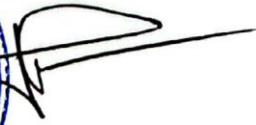
(  )

Anggota Penguji III : Dr. H. M. Imamudin Lc, MA  
NIP. 19740602 200901 1 010

(  )

Mengetahui dan Mengesahkan,  
Ketua Program Studi Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



  
Dr. Faehru Kurniawan, M.MT, IPM  
NIP. 19771020 200912 1 001

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Masfu'ul Aji Eka Putra  
NIM : 17650050  
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika  
Judul Skripsi : Sistem Kecerdasan Buatan Pengelolaan  
Sumber Daya Menggunakan Algoritma Genetika

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 17 Juni 2024  
Yang membuat pernyataan,



Masfu'ul Aji Eka Putra  
NIM.17650050

## **MOTTO**

*... Keberhasilan adalah buah dari kerja keras. ...*

## HALAMAN PERSEMBAHAN

Dengan penuh rasa syukur dan bangga, skripsi ini saya persembahkan kepada:

1. Kedua orang tua tercinta

Ayah dan Ibu, yang selalu memberikan cinta, doa, dan dukungan tanpa henti dalam setiap langkah perjalanan saya. Tanpa kasih sayang dan pengorbanan kalian, saya tidak akan mampu mencapai titik ini.

2. Para dosen dan pembimbing

Pak Okta dan Pak Imamudin yang dengan sabar membimbing dan memberikan ilmu serta arahan selama proses penyusunan skripsi ini. Terima kasih atas waktu, pengetahuan, dan kebijaksanaan yang telah diberikan.

3. Seluruh rekan mahasiswa,

yang telah bersama-sama berjuang dalam suka dan duka selama menempuh pendidikan di universitas ini. Semoga kita semua dapat meraih kesuksesan dan cita-cita masing-masing.

4. Almamater tercinta,

Universitas Islam Negeri Maulana Malik Ibrahim Malang yang telah memberikan tempat dan kesempatan untuk belajar serta mengembangkan diri. Semoga kita semua dapat membawa nama baik almamater di masa depan.

## KATA PENGANTAR

Puji syukur kehadirat Allah *subhanahu wa ta'ala* yang sudah memberikan rahmat serta hidayah-Nya, sehingga penulis mampu menyelesaikan proposal skripsi “Sistem Kecerdasan Buatan Pengelolaan Sumber Daya Developer Menggunakan Algoritma Genetika”. Terima kasih pada seluruh pihak yang sudah mendukung, membimbing, maupun membantu penulis pada penyusunan proposal skripsi ini. Jika tidak ada bimbingan, arahan, serta bimbingan melalui semua pihak, proposal skripsi ini tidak akan selesai dengan baik. Maka, penulis menyatakan terima kasih yang sebesar-besar mungkin kepada.

1. Prof. Dr. M. Zainuddin, M.A. selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang
2. Prof. Dr. Sri Harini, M.Si. selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang
3. Dr. Fachrul Kurniawan, M.MT selaku Ketua Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Okta Qomaruddin Aziz, M.Kom selaku Dosen Pembimbing I dan Dr. H. M. Imamudin Lc, MA selaku Dosen Pembimbing II, yang membimbing penulis dengan penuh kesabaran dan keikhlasan dalam meluangkan waktunya untuk membimbing sehingga penulis dapat menyelesaikan tugas akhir ini.
5. Seluruh dosen dan staf di Jurusan Teknik Informatika, dengan ikhlas memberikan ilmu, bantuan, serta dorongan semangat selama perkuliahan.

6. Ayahanda tercinta Marji dan Ibunda Sutik Isrukha yang senantiasa memberi do'a, dukungan, nasehat serta restunya pada penulis dalam menuntut ilmu.
7. Sejumlah teman penulis yang senantiasa mendengarkan keluh kesah, memberikan dorongan dan semangat kepada penulis

Penulis menyadari pada pembuatan proposal skripsi ini terdapat banyak keterbatasan. Oleh karenanya, dengan seluruh kerendahan hati, penulis mengharapkan saran maupun kritik yang bermanfaat untuk menyempurnakan proposal skripsi ini. Semoga pembaca dan penulis mendapatkan manfaat darinya.

Malang, 17 Juni 2024

Penulis

## DAFTAR ISI

<b>HALAMAN PENGAJUAN</b> .....	<b>II</b>
<b>HALAMAN PERSETUJUAN</b> .....	<b>III</b>
<b>HALAMAN PENGESAHAN</b> .....	<b>IV</b>
<b>PERNYATAAN KEASLIAN TULISAN</b> .....	<b>V</b>
<b>MOTTO</b> .....	<b>VI</b>
<b>HALAMAN PERSEMBAHAN</b> .....	<b>VII</b>
<b>KATA PENGANTAR</b> .....	<b>VIII</b>
<b>DAFTAR ISI</b> .....	<b>X</b>
<b>DAFTAR TABEL</b> .....	<b>XII</b>
<b>DAFTAR GAMBAR</b> .....	<b>XIII</b>
<b>ABSTRAK</b> .....	<b>XIV</b>
<b>ABSTRACT</b> .....	<b>XV</b>
<b>مستخلص البحث</b> .....	<b>XVI</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	5
1.3 Tujuan Penelitian .....	6
1.4 Manfaat Penelitian: .....	6
1.5 Batasan Masalah .....	7
<b>BAB II LANDASAN TEORI</b> .....	<b>8</b>
2.1 Penelitian Terkait .....	8
2.2 Studi Pustaka .....	11
2.2.1 Manajemen Sumber Daya .....	11
2.2.2 Algoritma Genetika .....	12
<b>BAB III METODOLOGI PENELITIAN</b> .....	<b>15</b>
3.1 Desain Penelitian .....	15
3.2 Tempat dan Waktu Penelitian .....	16
3.3 Sumber Data .....	16
3.4 Desain Sistem .....	16
3.4.1 Input Data .....	17
3.4.2 Inisialisasi Kromosom .....	19
3.4.3 Evaluasi Fitness .....	19
3.4.4 Proses Seleksi .....	23
3.4.5 Proses Crossover .....	24
3.4.6 Proses Mutasi .....	25
3.4.7 Hasil .....	26
3.5 Skenario Uji Coba .....	27
3.6 Implementasi .....	29
3.6.1 Inisialisasi Kromosom .....	29
3.6.2 Evaluasi Fitness .....	30
3.6.3 Proses Seleksi .....	33
3.6.4 Proses Crossover .....	34

3.6.5 Proses Mutasi.....	35
<b>BAB IV UJI COBA DAN PEMBAHASAN .....</b>	<b>36</b>
4.1 Hasil Uji Coba.....	36
4.1.1 Uji Coba Skenario 1.....	36
4.1.2 Uji Coba Skenario 2.....	37
4.1.3 Uji Coba Skenario 3.....	39
4.1.4 Uji Coba Skenario 4.....	40
4.1.5 Uji Coba Skenario 5.....	41
4.1.6 Uji Coba Skenario 6.....	43
4.2 Pembahasan.....	44
4.3 Integrasi Al-Qur-an Dan Hadits .....	47
4.3.1 <i>Hablum Minallah</i> .....	49
4.3.2 <i>Hablum Minannas</i> .....	51
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>53</b>
5.1 Kesimpulan .....	53
5.2 Saran.....	54
<b>DAFTAR PUSTAKA</b>	
<b>LAMPIRAN</b>	

## DAFTAR TABEL

Tabel 2. 1 Penelitian terkait yang telah dilakukan sebelumnya .....	10
Tabel 3. 1 Contoh data developer.....	18
Tabel 3. 2 Contoh data Proyek .....	18
Tabel 3. 3 Kromosom 1 .....	19
Tabel 3. 4 Kromosom 2 .....	19
Tabel 3. 5 Kromosom 3 .....	19
Tabel 3. 6 Evaluasi Fitness Kromosom 1 .....	22
Tabel 3. 7 Evaluasi Fitness Kromosom 2 .....	22
Tabel 3. 8 Evaluasi Fitness Kromosom 3 .....	23
Tabel 3. 9 Evaluasi Fitness Kromosom 4 .....	23
Tabel 3. 10 Seleksi Kromosom 1 .....	24
Tabel 3. 11 Seleksi Kromosom 2 .....	24
Tabel 3. 12 Crossover kromosom 1 individu pertama .....	25
Tabel 3. 13 Crossover kromosom 2 individu pertama .....	25
Tabel 3. 14 Proses mutasi kromosom 1 .....	26
Tabel 3. 15 Proses mutasi kromosom 2.....	26
Tabel 3. 16 Hasil kromosom 1 iterasi pertama.....	26
Tabel 3. 17 Hasil kromosom 2 iterasi pertama.....	27
Tabel 3. 18 Skenario uji coba dengan pembagian data latih dan uji .....	28
Tabel 4. 1 Hasil Prediksi pada proyek APM .....	36
Tabel 4. 2 Hasil uji coba skenario 1 .....	37
Tabel 4. 3 Hasil Prediksi pada proyek APM .....	38
Tabel 4. 4 Hasil uji coba skenario 2 .....	38
Tabel 4. 5 Hasil Prediksi pada proyek APM .....	39
Tabel 4. 6 Hasil uji coba skenario 3 .....	39
Tabel 4. 7 Hasil Prediksi pada proyek APM .....	40
Tabel 4. 8 Hasil uji coba skenario 4 .....	41
Tabel 4. 9 Hasil Prediksi pada proyek APM .....	42
Tabel 4. 10 Hasil uji coba skenario 5 .....	42
Tabel 4. 11 Hasil Prediksi pada proyek APM .....	43
Tabel 4. 12 Hasil uji coba skenario 6 .....	44
Tabel 4. 13 Recognition Rate Hasil Pengujian Skenario 1 – 6 .....	45

## DAFTAR GAMBAR

Gambar 2. 1 Blok Diagram Algoritma Genetika .....	14
Gambar 3. 1 Flowchart Aplikasi .....	17
Gambar 3. 2 Inisialisasi Populasi .....	29
Gambar 3. 3 Fungsi fitness untuk menghitung duplikat developer .....	30
Gambar 3. 4 Fungsi fitness kesesuaian proyek dengan developer.....	30
Gambar 3. 5 Fungsi fitness Penggunaan Senior Developer.....	31
Gambar 3. 6 Fungsi fitness Penggunaan Keahlian Developer.....	32
Gambar 3. 7 Fungsi fitness Kompleksitas Proyek .....	33
Gambar 3. 8 Proses Seleksi.....	33
Gambar 3. 9 Proses Crossover .....	34
Gambar 3. 10 Proses Mutasi .....	35
Gambar 4. 1 Grafik pengaruh crossover pada recognition rate .....	46

## ABSTRAK

Aji Eka Putra, Masfu'ul. 2024. **Sistem Kecerdasan Buatan Pengelolaan Sumber Daya Developer Menggunakan Algoritma Genetika**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Promotor: (I) Okta Qomaruddin Aziz, M.Kom, (II) Dr. H. M. Imamudin Lc, MA.

**Kata Kunci:** Algoritma Genetika, Kecerdasan Buatan, Pengelolaan Sumber Daya

Pengelolaan sumber daya developer yang efektif dan efisien merupakan tantangan utama dalam industri pengembangan perangkat lunak. Masalah utama yang dihadapi adalah penempatan developer pada proyek yang tidak sesuai dengan keahlian mereka, yang dapat mengakibatkan penurunan produktivitas dan kualitas hasil kerja. Untuk mengatasi masalah ini, penelitian ini mengusulkan penggunaan sistem kecerdasan buatan dengan algoritma genetika sebagai solusi untuk mengoptimalkan penempatan developer pada proyek yang sesuai dengan keahlian mereka. Algoritma genetika adalah metode komputasi yang meniru proses evolusi alami untuk menemukan solusi optimal. Metode ini dimulai dengan membangkitkan populasi awal yang diinisialisasi secara acak, di mana setiap individu dalam populasi merepresentasikan solusi potensial. Proses evolusi kemudian diterapkan melalui seleksi, crossover (pindah silang), dan mutasi. Siklus ini berulang hingga mencapai solusi yang optimal atau kondisi penghentian tertentu. Hasil uji coba menunjukkan bahwa skenario ketiga, dengan tingkat crossover sebesar 0,4 dan mutasi sebesar 0,2, memberikan performa terbaik dengan recognition rate mencapai 83%. Hal ini menandakan bahwa kombinasi parameter tersebut berhasil mencapai keseimbangan yang optimal antara eksplorasi dan eksploitasi dalam ruang solusi.

## ABSTRACT

Aji Eka Putra, Masfu'ul. 2024. **Artificial Intelligence System for Developer Resource Management Using Genetic Algorithms**. Thesis. Department of Informatics Engineering, Faculty of Science and Technology, Maulana Malik Ibrahim Islamic State University, Malang. Promotor: (I) Okta Qomaruddin Aziz, M.Kom, (II) Dr. H. M. Imamudin Lc, MA.

**Key words:** Genetic Algorithms, Artificial Intelligence, Resource Management

Effective and efficient management of developer resources is a major challenge in the software development industry. The main problem faced is the placement of developers on projects that do not match their expertise, which can result in decreased productivity and quality of work. To address this issue, this research proposes the use of an artificial intelligence system with a genetic algorithm as a solution to optimize the placement of developers on projects that match their expertise. The genetic algorithm is a computational method that mimics the natural evolution process to find optimal solutions. This method begins by generating an initial population randomly, where each individual in the population represents a potential solution. The evolution process is then applied through selection, crossover, and mutation. This cycle repeats until an optimal solution or a predetermined stopping condition is reached. The experimental results show that the third scenario, with a crossover rate of 0.4 and a mutation rate of 0.2, provides the best performance with a recognition rate of 83%. This indicates that the combination of these parameters achieves an optimal balance between exploration and exploitation within the solution space.

## مستخلص البحث

أجي إيكافوترا، مسفعل. ٢٠٢٤. نظام الذكاء الاصطناعي لإدارة الموارد المطورين باستخدام الخوارزميات الجينية. قسم الهندسة المعلوماتية كلية العلوم والتكنولوجيا الجامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانج. المحاضر: (1) الأستاذ أوكتا قمر الدين عزيز الماجستير (2) الأستاذ الدكتور الحاج محمد إمام الدين الماجستير.

**الكلمات الرئيسية:** الخوارزميات الجينية، الذكاء الاصطناعي، إدارة الموارد.

تمثل الإدارة الفعالة والمؤثرة لموارد المطورين تحديًا كبيرًا في صناعة تطوير البرمجيات. المشكلة الرئيسية التي نواجهها هي وضع المطورين في مشاريع لا تتناسب مع مهاراتهم، مما قد يؤدي إلى انخفاض الإنتاجية وجودة مخرجات العمل. للتغلب على هذه المشكلة يقترح هذا البحث استخدام نظام الذكاء الاصطناعي مع الخوارزميات الجينية كحل لتحسين وضع المطورين في المشاريع التي تناسب مهاراتهم. الخوارزمية الجينية هي طريقة حسابية تحاكي العملية التطورية الطبيعية لإيجاد الحلول المثلى. تبدأ هذه الطريقة بتوليد مجتمع أولي تمت تهيئته بشكل عشوائي، حيث يمثل كل فرد في المجتمع حلاً محتملاً. يتم بعد ذلك تنفيذ العملية التطورية من خلال الاختيار والعبور والطفرة. تتكرر هذه الدورة حتى الوصول إلى الحل الأمثل أو حالة توقف معينة. وأظهرت نتائج الاختبار أن السيناريو الثالث بمعدل عبور 0.4 وطفرة 0.2 يقدم أفضل أداء بمعدل التعرف الذي يصل إلى 83%. وهذا يدل على أن مجموعة المعلمات نجحت في تحقيق التوازن الأمثل بين الاستكشاف والاستغلال في مساحة الحل.

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Pengembangan perangkat lunak merupakan salah satu industri yang berkembang pesat dalam era digital ini. Permintaan akan aplikasi dan sistem yang inovatif terus meningkat seiring dengan kebutuhan pengguna yang semakin kompleks. Dalam menghadapi tantangan ini, perusahaan perangkat lunak harus mampu mengembangkan solusi yang cepat, efisien, dan optimal (Arifudin, n.d.). Alokasi sumber daya menjadi tantangan utama dalam manajemen proyek. Karena kerumitan dan alokasi sumber daya yang tepat, seringkali sulit untuk mencari solusi optimal bagi proyek yang dikerjakan (Tarek Hegazy, n.d.).

Proses pengembangan perangkat lunak melibatkan berbagai tahap, mulai dari perencanaan, desain, pengembangan, pengujian, hingga implementasi. Setiap tahap membutuhkan alokasi sumber daya yang tepat, terutama dalam hal tenaga kerja. Alokasi sumber daya developer menjadi salah satu faktor kunci dalam menentukan keberhasilan proyek perangkat lunak. Jika sumber daya developer tidak dialokasikan dengan benar, hal ini dapat menyebabkan penundaan proyek, biaya yang tidak terkontrol, dan ketidakpuasan pelanggan.

Penting bagi perusahaan perangkat lunak untuk mengoptimalkan alokasi sumber daya developer agar proyek dapat diselesaikan dengan efisien dan sesuai dengan tenggat waktu yang ditetapkan (Xie, Chen, and Chang 2021). Namun, pengambilan keputusan dalam alokasi sumber daya developer merupakan tugas yang kompleks dan menantang. Manajer proyek harus mempertimbangkan

berbagai faktor seperti keahlian, pengalaman, preferensi, dan tenggat waktu proyek untuk mengoptimalkan alokasi tersebut. Pendekatan manual dalam proses ini seringkali tidak efektif dan rentan terhadap kesalahan serta bias subjektif.

Beberapa tahun terakhir, kecerdasan buatan (artificial intelligence) telah menjadi tren utama dalam industri teknologi. Salah satu bidang yang berkembang pesat dalam kecerdasan buatan adalah algoritma genetika. Algoritma genetika adalah metode optimasi yang terinspirasi dari prinsip seleksi alam evolusi (Oktarina and Hajjah 2019). Metode ini menggunakan teknik pemodelan genetika untuk mencari solusi yang optimal dalam ruang pencarian yang kompleks.

Penerapan algoritma genetika dalam alokasi sumber daya developer dapat membantu perusahaan-perusahaan perangkat lunak dalam memecahkan masalah kompleks dan mencapai efisiensi yang lebih tinggi. Dengan menggunakan algoritma genetika, sistem kecerdasan buatan dapat mempertimbangkan berbagai faktor seperti keahlian dan pengalaman developer, preferensi tim, tenggat waktu proyek, dan kebutuhan sumber daya lainnya untuk menghasilkan alokasi sumber daya yang optimal.

Konteks alokasi sumber daya developer, penggunaan algoritma genetika dapat membantu mencari solusi yang optimal dengan mempertimbangkan berbagai faktor yang relevan. Algoritma ini dapat membantu manajer proyek dalam menentukan penugasan developer ke proyek-proyek yang sedang berjalan dengan cara yang efisien dan akurat. Dengan memanfaatkan kecerdasan buatan dan algoritma genetika, sistem alokasi sumber daya developer dapat menghasilkan

alokasi yang lebih baik, yang pada akhirnya akan meningkatkan efisiensi dan efektivitas dalam pengembangan perangkat lunak.

Al-Qur'an mengemukakan gambaran yang jelas tentang berbagai persoalan aspek kehidupan manusia. Di antara aspek terpenting tersebut adalah manajemen sumber daya manusia unggul. Allah *subhanahu wa ta'ala* berfirman dalam Al Qur'an surat Al Jatsiyah ayat 13

وَسَخَّرَ لَكُمْ مَّا فِي السَّمٰوٰتِ وَمَا فِي الْاَرْضِ جَمِيعًا مِّنْهُ اِنَّ فِيْ ذٰلِكَ لَآٰيٰتٍ لِّقَوْمٍ يَّتَفَكَّرُوْنَ

*“Dan Dia telah menundukkan untukmu apa yang di langit dan apa yang di bumi semuanya, (sebagai rahmat) daripada-Nya. Sesungguhnya pada yang demikian itu benar-benar terdapat tanda-tanda (kekuasaan Allah) bagi kaum yang berfikir.” (Q.S Al Jatsiyah : 13)*

Menurut tafsir Al-Mukhtashar yang berisi “Dan Allah *subhanahu wa ta'ala* menundukkan bagi kalian apa yang ada di langit berupa matahari, bulan dan bintang-bintang, dan apa yang ada di bumi berupa sungai-sungai, pepohonan, langit-langit dan lain-lain. Sesungguhnya dalam hal ditundukkannya semua itu untuk kalian benar-benar terdapat bukti-bukti atas kekuasaan Allah dan keesaan-Nya bagi orang-orang yang berfikir tentang ayat-ayatnya dan mengambil pelajaran darinya.” (TafsirWeb, 2024). Ayat di atas menjelaskan bahwa segala potensi yang telah diberikan Allah harus dijaga dan dimanfaatkan dengan sebaik baiknya. Potensi manusia berfikir harus selalu bergerak dan bermanfaat kehidupannya.

Pengelolaan SDM yang baik tidak hanya melibatkan administrasi tenaga kerja, tetapi juga mencakup penempatan yang tepat, pengembangan kemampuan, serta pemberian motivasi yang sesuai. Dalam perspektif Islam, prinsip-prinsip

manajemen SDM dapat ditemukan dalam ajaran Al-Qur'an yang memberikan petunjuk mengenai keadilan, tanggung jawab, dan amanah. Salah satu ayat yang secara eksplisit memberikan panduan terkait manajemen SDM adalah Surah An-Nisa ayat 58:

إِنَّ اللَّهَ يَأْمُرُكُمْ أَنْ تُؤَدُّوا الْأَمَانَاتِ إِلَىٰ أَهْلِهَا وَإِذَا حَكَمْتُمْ بَيْنَ النَّاسِ أَنْ تَحْكُمُوا بِالْعَدْلِ ۗ إِنَّ اللَّهَ نِعِمَّا يَعِظُكُمْ بِهِ ۗ إِنَّ اللَّهَ كَانَ سَمِيعًا بَصِيرًا

*"Sesungguhnya Allah menyuruh kamu menyampaikan amanat kepada yang berhak menerimanya, dan (menyuruh kamu) apabila menetapkan hukum di antara manusia supaya kamu menetapkan dengan adil. Sesungguhnya Allah memberi pengajaran yang sebaik-baiknya kepadamu. Sesungguhnya Allah adalah Maha Mendengar lagi Maha Melihat."(Q.S An-Nisa : 58)*

Tafsir Al-Muyassar menjelaskan ayat ini dengan “Sesungguhnya Allah memerintahkan kalian untuk menunaikan amanat yang berbeda-beda yang kalian dipercaya untuk menyampaikannya kepada para pemiliknya, maka janganlah kalian melalaikan amanat-amanat itu. Dan Dia memerintahkan kalian untuk memutuskan perkara diantara manusia dengan dasar keadilan dan obyektif, bila kalian memutuskan permasalahan diantara mereka. Dan itu adalah sebaik-baik nasihat yang Allah sampaikan kepada kalian dan memberi petunjuk kalian kepadanya. Sesungguhnya Allah Maha Mendengar ucapan-ucapan kalian,meneliti seluruh perbuatan kalian lagi Maha Melihatnya.” (TafsirWeb, 2024). Ayat ini menekankan pentingnya menempatkan amanah atau tanggung jawab kepada pihak yang berhak dan mampu, serta menegakkan keadilan dalam setiap keputusan yang diambil. Dalam konteks manajemen SDM, hal ini dapat diartikan sebagai pentingnya rekrutmen yang adil dan transparan, penempatan karyawan sesuai dengan

kompetensi dan potensi mereka, serta pengambilan keputusan yang objektif dan berdasarkan meritokrasi.

Penelitian ini, akan dikembangkan sebuah sistem alokasi sumber daya developer menggunakan algoritma genetika untuk meningkatkan efisiensi dan efektivitas dalam pengembangan perangkat lunak. Sistem ini akan dirancang dengan mempertimbangkan berbagai faktor penting, seperti keahlian, pengalaman, preferensi, dan tenggat waktu proyek. Dengan memanfaatkan kecerdasan buatan, sistem ini akan memproses data dan informasi yang diperoleh untuk menghasilkan alokasi sumber daya developer yang optimal.

Melalui penelitian ini, diharapkan dapat memberikan kontribusi dalam pengembangan metode alokasi sumber daya yang lebih efisien dan akurat dalam industri perangkat lunak. Dengan menggunakan algoritma genetika dan kecerdasan buatan, perusahaan-perusahaan perangkat lunak akan dapat mengoptimalkan alokasi sumber daya developer mereka, mengurangi penundaan proyek, mengendalikan biaya, dan meningkatkan kepuasan pelanggan. Dengan demikian, industri perangkat lunak akan semakin maju dan dapat memenuhi permintaan yang terus meningkat dari pengguna.

## **1.2 Rumusan Masalah**

1. Bagaimana pengaruh crossover rate dan mutation rate terhadap kinerja algoritma genetika?
2. Berapa persentase recognition rate yang dapat dicapai oleh algoritma genetika?

### **1.3 Tujuan Penelitian**

1. Mengidentifikasi dan menganalisis bagaimana variasi dalam nilai crossover rate dan mutation rate mempengaruhi kinerja dan hasil akhir dari algoritma genetika
2. Mengukur persentase recognition rate yang dicapai oleh algoritma genetika dalam aplikasi tertentu serta mengevaluasi efektivitasnya dalam konteks yang ditentukan.

### **1.4 Manfaat Penelitian:**

1. Perusahaan-perusahaan perangkat lunak dapat menggunakan hasil penelitian ini untuk meningkatkan efisiensi dan efektivitas dalam alokasi sumber daya developer. Dengan adanya sistem kecerdasan buatan yang menggunakan algoritma genetika, penugasan developer dapat dilakukan secara otomatis dan optimal, mengurangi waktu dan biaya yang diperlukan untuk alokasi sumber daya manusia.
2. Sistem yang dikembangkan dapat membantu manajer proyek dalam membuat keputusan yang tepat dalam alokasi sumber daya developer. Dengan mempertimbangkan faktor-faktor seperti keahlian dan pengalaman, sistem dapat memastikan bahwa developer yang paling sesuai ditugaskan ke proyek yang sesuai dengan kemampuannya.
3. Penelitian ini memberikan kontribusi dalam pengembangan metode dan teknik kecerdasan buatan dalam industri perangkat lunak. Penggunaan

algoritma genetika untuk alokasi sumber daya developer dapat menjadi acuan bagi penelitian dan pengembangan selanjutnya di bidang ini.

### **1.5 Batasan Masalah**

1. Penelitian ini berfokus pada sumber daya developer seperti keahlian, prestasi, dan pengalaman. Faktor-faktor seperti anggaran, peralatan dan infrastruktur tidak dipertimbangkan dalam penelitian ini.
2. Parameter lain dalam algoritma genetika, seperti ukuran populasi, jumlah generasi, dan seleksi, tidak akan dibahas secara mendalam

## **BAB II**

### **LANDASAN TEORI**

Bab ini menjelaskan beberapa studi pustaka yang digunakan untuk penelitian. Bab ini juga menjelaskan tentang argumentasi ilmiah yang dipakai sebagai referensi untuk menilai seberapa jauh penelitian pada topik tersebut telah dilakukan

#### **2.1 Penelitian Terkait**

Ada beberapa penelitian yang sudah dilakukan sebelumnya yang terkait dengan penelitian ini diantaranya adalah:

Zhi Li, dkk pada tahun 2020 melakukan penelitian tentang Optimalisasi Offloading dan Alokasi Sumber Daya Berbasis Algoritma Genetika dalam Komputasi Mobile-Edge. Penelitian ini dilakukan dengan mengoptimalkan bandwidth atau sumber daya komputasi Mobile Edge Server (MES) secara terpisah untuk dapat mempercepat penyelesaian tugas. Kinerja algoritma pada penelitian ini dievaluasi melalui simulasi matlab dengan skenario menghitung waktu penyelesaian tugas berdasarkan jumlah perangkat dengan bandwidth  $10\text{MHz}$ , dan sumberdaya komputasi MES sebesar  $5\text{GHz/s}$ . Tanpa menggunakan algoritma genetika, waktu penyelesaian tugas adalah 10 detik. Algoritma Genetika dapat mempercepat tugas dibawah 2 detik Namun, ketika jumlah perangkat meningkat hingga di atas 20, algoritma genetika baru mendekati nilai tersebut. Kesimpulannya, baik bandwidth yang dioptimalkan maupun sumber daya komputasi dapat memperoleh total waktu penyelesaian yang cepat.

Damayanti, dkk pada tahun 2023 melakukan penelitian tentang Implementasi Algoritma Genetika Dalam Keterbatasan Sumber Daya Manusia Pada Proyek Konstruksi. Data yang digunakan meliputi waktu pelaksanaan proyek, Kurva S, dan distribusi sumber daya manusia pada proyek tersebut. Metode algoritma genetika menghasilkan lebih dari satu solusi optimal. Penelitian ini menghasilkan 5 alternatif solusi dengan menggunakan Microsoft Excel. Setiap solusi memiliki durasi dan jumlah sumber daya manusia yang berbeda. Durasi jadwal proyek optimal yang terbentuk melalui Microsoft Excel adalah 60 hari dengan nilai fitness 0,555. Durasi awal pekerjaan pemasangan atap pada Proyek adalah 70 hari. Hasil pengolahan data menggunakan metode algoritma genetika dengan Microsoft Excel dapat mempercepat pekerjaan sebesar 14.29%. Optimasi menggunakan metode algoritma genetika dapat menjadikan durasi pekerjaan menjadi lebih pendek dari durasi awal.

Amalia Rizqi Utami, dkk pada tahun 2022 melakukan penelitian dengan judul Analisis Performa Sistem High Altitude Platforms Menggunakan Algoritma Genetika Untuk Pengalokasian Subcarrier. Penelitian ini menggunakan Matlab untuk mengalokasikan daya yang dibagi sama rata tiap resource block. Pada setiap resource block terdapat 12 subcarrier. Simulasi dilakukan dengan variasi jumlah pengguna dan bandwidth untuk mendapatkan hasil yang optimal. Pada simulasi, terdapat 2 jumlah resource block yang berbeda, yaitu 15 dan 25. Dengan menggunakan Algoritma Genetika, nilai rata-rata kapasitas sistem untuk RB=15 adalah 4.9428, sedangkan tanpa Algoritma Genetika, nilai rata-rata kapasitas sistem adalah 4.0151. Untuk RB=25, dengan Algoritma Genetika, nilai rata-rata kapasitas sistem mencapai 9.8049, sementara tanpa Algoritma Genetika, nilai rata-rata

kapasitas sistem adalah 7.8051. Simulasi menunjukkan bahwa peningkatan jumlah pengguna akan menurunkan kapasitas rata-rata sistem. Selain itu, kapasitas juga dipengaruhi oleh jumlah subcarrier yang dialokasikan menggunakan Algoritma Genetika, di mana peningkatan jumlah subcarrier akan meningkatkan kapasitas sistem.

Tabel 2. 1 Penelitian terkait yang telah dilakukan sebelumnya

No	Judul	Variabel	Metode	Hasil
1	Optimalisasi Offloading dan Alokasi Sumber Daya Berbasis Algoritma Genetika dalam Komputasi Mobile-Edge	Sumber daya komputer MES, bandwidth, data tugas, cpu yang diperlukan untuk 1 tugas, kapasitas komputasi perangkat, jumlah perangkat	Algoritma Genetika	Akurasi algoritma dapat mempercepat proses tugas sebesar 80% jika ada kurang dari 20 perangkat
2	Implementasi Algoritma Genetika Dalam Keterbatasan Sumber Daya Manusia Pada Proyek Konstruksi	waktu pelaksanaan proyek (time schedule), Kurva S, dan distribusi sumber daya manusia pada proyek.	Algoritma Genetika	Algoritma dapat mempercepat jadwal proyek sebesar 14.29% dengan nilai fitness 0,555
3	Analisis Performa Sistem High Altitude Platforms Menggunakan Algoritma Genetika Untuk Pengalokasian Subcarrier	jumlah user dan bandwidth	Algoritma Genetika	Algoritma Genetika memperoleh nilai rata-rata optimasi sebesar 18.75%

Pada tabel penelitian di atas dapat diketahui bahwa dalam penelitian 1, peneliti menggunakan algoritma genetika untuk pengoptimasian offloading dan sumber daya dari edge server untuk mempercepat pengerjaan task dengan terbatasnya edge server akan tetapi perangkat mobile dapat terus bertambah.

Pengalokasian tugas pada edge server didapatkan mengurangi waktu tugas sebesar 80%.

## **2.2 Studi Pustaka**

Bahan-bahan studi pustaka dapat diperoleh dari berbagai sumber seperti hasil-hasil penelitian yang telah diuji kebenarannya seperti, jurnal penelitian, laporan penelitian, buku teks, laporan seminar, diskusi ilmiah, dan terbitan-terbitan resmi dari pemerintah.

### **2.2.1 Manajemen Sumber Daya**

Manajemen sumber daya manusia (SDM) memainkan peran penting dalam pembentukan dan pengelolaan developer. Dalam penelitian yang dilakukan oleh Belbin (2010), ditemukan bahwa komposisi tim yang seimbang, dengan anggota tim yang memiliki beragam keterampilan, kepribadian, dan gaya kerja, dapat meningkatkan kinerja tim secara keseluruhan. Oleh karena itu, pemilihan anggota tim yang tepat berdasarkan pada analisis peran dan kontribusi yang dibutuhkan dalam proyek menjadi kunci dalam manajemen SDM pada tahap pembentukan tim.

Selain itu, komunikasi yang efektif juga merupakan aspek penting dalam manajemen SDM pada pembentukan tim proyek. Menurut penelitian oleh Project Management Institute (PMI) (2017), komunikasi yang jelas dan terbuka antara anggota tim, pemimpin proyek, dan pemangku kepentingan lainnya dapat memperkuat koordinasi, kolaborasi, dan pemecahan masalah dalam tim. Selain itu, didukung oleh penelitian oleh Katzenbach dan Smith (2015), yang menunjukkan

bahwa tim proyek yang memiliki komunikasi yang baik cenderung lebih efektif dalam mencapai tujuan proyeknya.

Tidak hanya itu, manajemen konflik juga merupakan aspek penting dalam manajemen SDM pada pembentukan tim proyek. Menurut penelitian oleh Robbins dan Judge (2017), konflik dalam tim proyek tidak dapat dihindari, namun dapat dikelola dengan baik untuk meningkatkan kreativitas, inovasi, dan kinerja tim secara keseluruhan. Oleh karena itu, manajer proyek perlu memiliki keterampilan dalam mengidentifikasi, mencegah, dan menyelesaikan konflik yang muncul di dalam tim guna memastikan kelancaran dan keberhasilan proyek.

### **2.2.2 Algoritma Genetika**

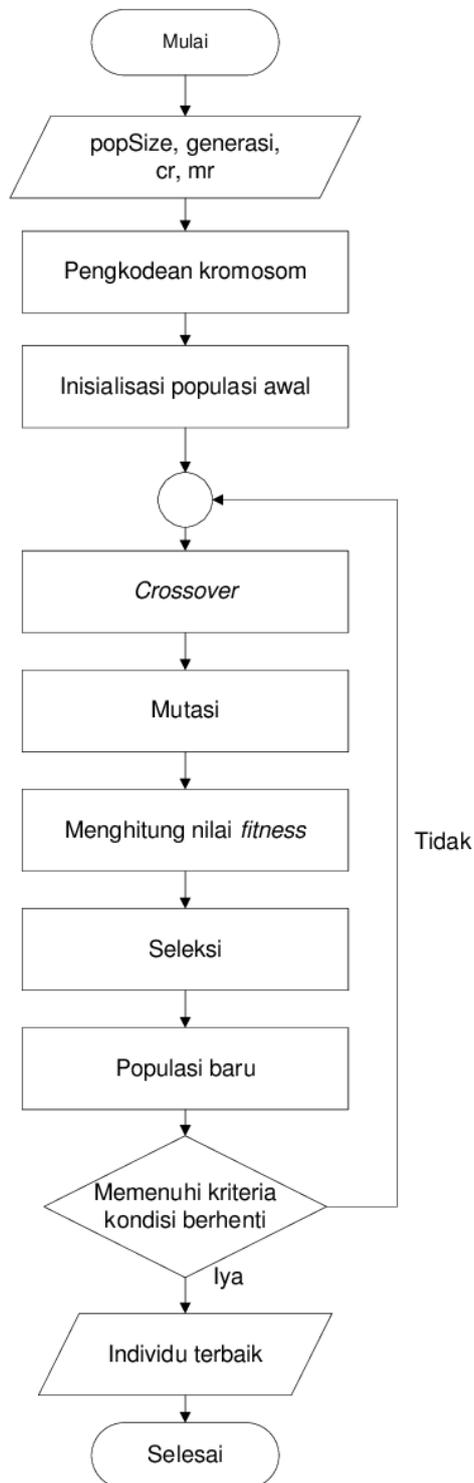
Algoritma Genetika (AG) adalah salah satu teknik optimasi yang terinspirasi dari teori evolusi dan genetika. Dalam penelitian oleh Holland (1975), AG pertama kali diperkenalkan sebagai metode untuk menyelesaikan masalah pencarian dan optimasi yang kompleks. AG bekerja dengan cara menggabungkan prinsip-prinsip seleksi alam, rekombinasi genetik, dan mutasi untuk mencari solusi yang optimal atau mendekati solusi terbaik dari suatu masalah.

Salah satu keunggulan utama dari AG adalah kemampuannya untuk menemukan solusi dalam ruang pencarian yang besar dan kompleks. Menurut penelitian oleh Goldberg (1989), AG dapat menangani masalah optimasi yang melibatkan banyak variabel dan batasan dengan efisien, sehingga sering digunakan dalam berbagai bidang seperti rekayasa, ilmu komputer, keuangan, dan biologi.

Selain itu, AG juga dikenal karena kemampuannya untuk menemukan solusi yang sub-optimal dalam waktu yang relatif singkat. Menurut penelitian oleh Mitchell (1996), AG dapat mengeksplorasi ruang pencarian secara paralel, melakukan operasi genetik pada populasi solusi potensial, dan secara bertahap meningkatkan kualitas solusi dari generasi ke generasi.

Namun, AG juga memiliki keterbatasan dan tantangan tertentu. Salah satu tantangan utama adalah penyesuaian parameter AG yang tepat agar sesuai dengan karakteristik masalah yang dihadapi. Menurut penelitian oleh Whitley (1994), pemilihan parameter AG yang tidak tepat dapat mengakibatkan konvergensi yang lambat atau bahkan terjebak dalam optimum lokal.

Meskipun demikian, AG tetap menjadi salah satu teknik optimasi yang sangat populer dan banyak digunakan dalam berbagai aplikasi di berbagai bidang, dan terus menjadi fokus penelitian untuk meningkatkan kinerja dan efisiensinya dalam menyelesaikan masalah optimasi yang semakin kompleks dan besar. Secara umum, blok diagram dari mekanisme kerja algoritma genetika adalah seperti berikut :



Gambar 2. 1 Blok Diagram Algoritma Genetika

## **BAB III**

### **METODOLOGI PENELITIAN**

Pada bab ini membahas tentang penjelasan tentang tahapan dan prosedur penelitian yang akan dilakukan. Bab ini juga membahas tentang kebutuhan sistem dan perangkat lunak yang digunakan.

Metode Penelitian ini menggunakan metode deskriptif yakni dengan cara melakukan penelitian studi kasus dengan mempelajari aktivitas di lapangan, mengamati dan wawancara kepada pihak yang terkait. Metode deskriptif adalah suatu metode penelitian yang ditunjukkan untuk membuat gambaran atau lukisan secara sistematis, aktual, dan akurat melalui data sampel atau populasi sebagaimana adanya.

#### **3.1 Desain Penelitian**

Desain penelitian ini bertujuan untuk mempermudah penulis dalam menentukan tahapan-tahapan penelitian yang akan dilakukan. Dalam melakukan penelitian hal pertama yang harus dilakukan adalah membuat studi literatur dengan tujuan untuk mendapatkan referensi terhadap apa yang diteliti. Selanjutnya adalah mengidentifikasi masalah sehingga setiap kebutuhan dalam penelitian dapat dianalisis. Kemudian melakukan pengumpulan data terhadap masalah yang diteliti, selanjutnya sistem mulai dirancang dan diimplementasikan dalam sistem. Setelah sistem bisa berjalan maka dilakukan pengujian dan evaluasi terhadap sistem, dalam tahap ini membahas tentang pengujian dari sistem dan akurasi hasil prediksi. Setelah sistem berjalan dan telah diuji.

### **3.2 Tempat dan Waktu Penelitian**

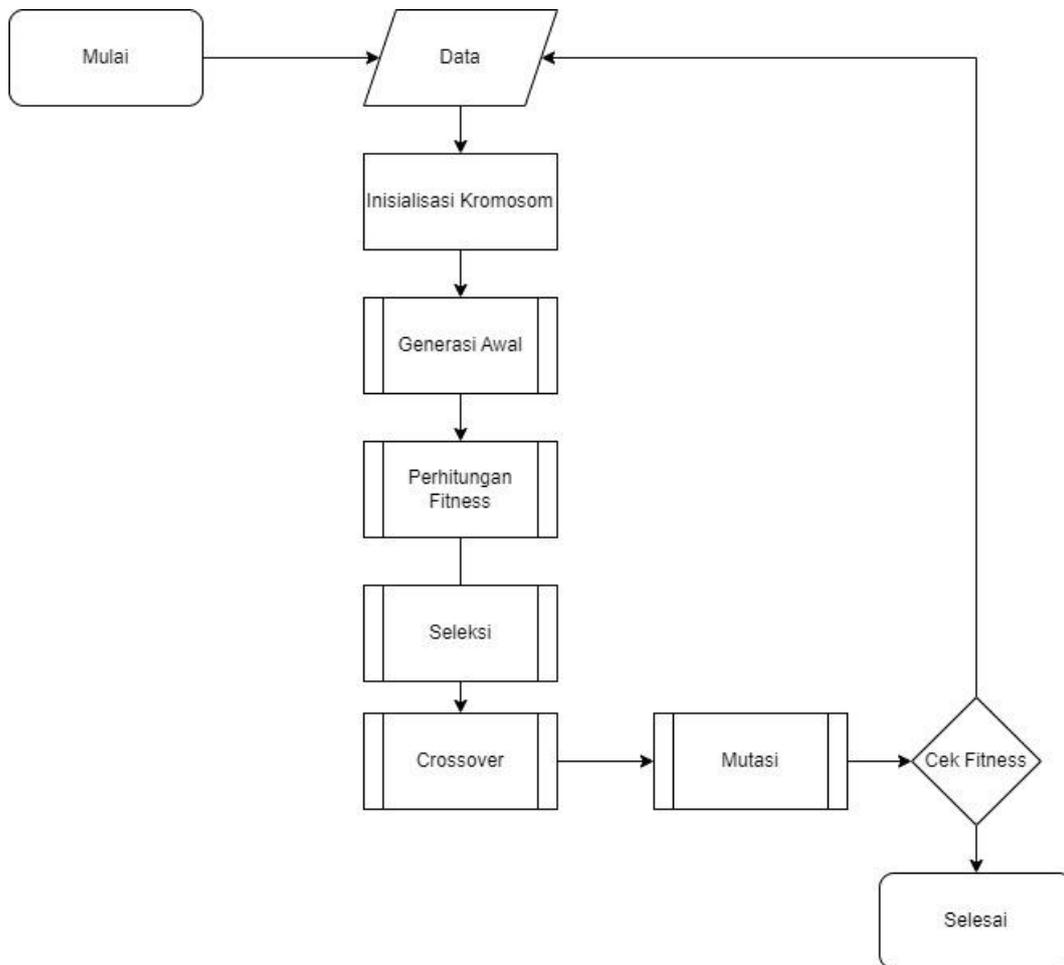
Penelitian ini dilakukan di lingkungan PT Sekawan Media Informatika, sebuah perusahaan di bidang software house yang telah memberikan dukungan penuh untuk kelancaran proses penelitian. Diharapkan bahwa penelitian ini dapat berjalan dengan baik berkat kerjasama dan fasilitas yang disediakan oleh perusahaan. Waktu penelitian dimulai pada bulan November dan berlangsung hingga penyelesaian secara keseluruhan, memungkinkan untuk analisis yang menyeluruh dan mendalam terhadap data yang terkumpul.

### **3.3 Sumber Data**

Pengumpulan data pada penelitian ini berupa data pegawai dan proyek pada perusahaan software house di PT Sekawan Media Informatika yang dilakukan pada bulan Maret 2023. Data tersebut mencakup informasi rinci mengenai profil pegawai, alokasi tugas, serta berbagai parameter lainnya yang relevan untuk analisis menggunakan algoritma genetika dalam pengelolaan sumber daya manusia. Data tersebut dikumpulkan secara sistematis untuk memastikan keakuratan dan relevansi dalam konteks penelitian yang sedang dilakukan.

### **3.4 Desain Sistem**

Pada bab ini menjelaskan mengenai bagaimana sistem dibuat dalam mempermudah penelitian untuk mengimplementasikan sistem, evaluasi, dan analisis hasil yang didapat. Sistem akan dibangun dengan menggunakan php.



Gambar 3. 1 Flowchart Aplikasi

Berikut ini penjelasan dari flowchart pada gambar 3.1 adalah sebagai berikut

### 3.4.1 Input Data

Data yang diperoleh adalah data pengalaman developer, bidang keahlian developer, tipe proyek.

### A. Data Developer

Data developer mencakup informasi tentang pengalaman dan keahlian utama dalam proyek-proyek yang telah dijalani. Struktur data developer dapat ditemukan dalam Tabel 3.1. Setiap developer akan ditempatkan dalam sebuah tim yang bertanggung jawab atas berbagai jenis proyek. Misalnya, tim back office akan fokus pada pengembangan aplikasi perkantoran seperti sistem absensi, sementara tim bisnis akan menangani proyek-proyek terkait keuangan seperti pengembangan aplikasi koperasi. Dalam setiap proyek, minimal satu developer senior akan ditugaskan untuk mengawasi jalannya proyek. Selain itu, spesialisasi developer akan ditunjukkan jika proyek memerlukan keahlian khusus tertentu.

Tabel 3. 1 Contoh data developer

No	Nama	Jabatan	Tim	Spealisasi
1	D1	Senior Developer	Back Office	Web
2	D2	Developer	Back Office	Web
3	D3	Project Admin	Custom	Backend
4	D4	Senior Developer	Production	Fullstack
5	D5	Developer	Bussines	Mobile

### B. Data Proyek

Data Proyek menggunakan data project yang akan diisi oleh developer yang ada. Kompleksitas project mempengaruhi sulit tidaknya proyek yang dikerjakan. Tingkat Kompleksitas dibagi menjadi 3 yaitu tinggi, sedang dan rendah. Struktur data project ditunjukkan pada Tabel 3.2.

Tabel 3. 2 Contoh data Proyek

No	Nama	Kompleksitas	Tipe	Platform
1	P1	Sedang	Office	Web
2	P2	Tinggi	Custom	Mobile

### 3.4.2 Inisialisasi Kromosom

Kromosom merupakan representasi struktur data yang menggambarkan solusi potensial atau individu dalam ruang pencarian. Ukuran kromosom bervariasi tergantung pada kompleksitas masalah yang akan diselesaikan. Salah satu contoh dari kromosom yang diinisialisasi secara acak dapat tampak seperti ini.

Tabel 3. 3 Kromosom 1

<b>Proyek</b>	<b>Developer</b>
P1	D1, D2
P2	D4, D5

Tabel 3. 4 Kromosom 2

<b>Proyek</b>	<b>Developer</b>
P1	D3, D5
P2	D1, D2

Tabel 3. 5 Kromosom 3

<b>Proyek</b>	<b>Developer</b>
P1	D1, D4
P2	D5, D5

Tabel 3.3 merupakan representasi dari alokasi developer ke dalam proyek-proyek tertentu dalam lingkungan perusahaan. Dalam tabel ini menunjukkan proyek P1 mendapatkan developer D1 dan D2 sedangkan proyek P2 mendapatkan proyek D4 dan D5. Hasil dari inisialisasi awal ini nantinya akan dievaluasi dan dioptimalkan oleh algoritma genetika sehingga mendapatkan solusi yang terbaik.

### 3.4.3 Evaluasi Fitness

Proses evaluasi fitness yang dirancang untuk mengevaluasi kualitas dari setiap solusi untuk menentukan seberapa baik solusi tersebut menyelesaikan

masalah yang diberikan. Evaluasi fitness dilakukan dengan membandingkan kinerja solusi terhadap kriteria atau fungsi tujuan yang telah ditetapkan.

Fungsi tujuan pertama yang digunakan adalah meminimalisir bentrokan antar developer. Bentrokan terjadi ketika satu atau lebih developer yang sama ditempatkan pada satu proyek yang sama secara bersamaan. Misalnya, jika developer A dan developer B keduanya ditempatkan pada proyek yang sama, itu dianggap sebagai bentrokan. Semakin sedikit bentrokan yang terjadi, semakin tinggi nilai fitnessnya. Ini karena meminimalkan bentrokan antar developer membantu memastikan bahwa tim pengembang terdiri dari anggota yang berbeda dan tidak tumpang tindih, sehingga mengoptimalkan efisiensi dan produktivitas dalam pelaksanaan proyek.

Fungsi tujuan kedua adalah menghitung pemanfaatan senior developer dengan menghitung jumlah senior developer yang terpakai di setiap proyek. Senior developer memiliki pengalaman dan keahlian yang berharga, dan optimalisasi penggunaannya dapat meningkatkan kualitas dan efisiensi pelaksanaan proyek. Evaluasi penggunaan senior developer dilakukan dengan menghitung bahwa setiap proyek minimal memiliki satu senior developer. Ketika sebuah proyek tidak memiliki satupun senior developer, hal ini akan mengurangi nilai fitnessnya.

Fungsi tujuan ketiga adalah memastikan penggunaan developer sesuai dengan keahliannya untuk setiap proyek yang diberikan. Ini berarti menilai sejauh mana developer yang dialokasikan ke proyek memiliki keterampilan dan keahlian yang sesuai dengan kebutuhan proyek tersebut. Misalnya, jika sebuah proyek membutuhkan pengembangan situs web, idealnya developer yang memiliki

keahlian dan pengalaman dalam pengembangan situs web akan dialokasikan ke proyek tersebut. Dalam mengevaluasi fungsi tujuan ini, algoritma akan memeriksa kecocokan antara keahlian developer dan kebutuhan proyek dengan mengacu pada spesifikasi teknis dan persyaratan proyek yang telah ditetapkan. Jika developer yang dialokasikan ke proyek tidak sesuai dengan keahliannya, hal ini dapat mengurangi nilai fitness solusi tersebut. Berikut adalah rumus yang digunakan dalam menentukan fitness dari setiap populasi

Fungsi tujuan keempat dalam algoritma genetika ini adalah memastikan setiap proyek mendapatkan developer yang sesuai dengan kompleksitas yang dimiliki oleh setiap proyek tersebut. Dalam konteks ini, proyek yang lebih kompleks memerlukan developer dengan keahlian dan pengalaman yang lebih tinggi untuk menjamin bahwa proyek dapat diselesaikan dengan kualitas yang optimal dan dalam waktu yang efisien. Untuk mencapai hal ini, algoritma genetika akan mengevaluasi kecocokan antara keahlian developer dan kebutuhan spesifik proyek, memastikan bahwa semakin kompleks proyek yang dikerjakan maka semakin banyak developer yang diperlukan. Dengan cara ini, fungsi tujuan ini berkontribusi terhadap peningkatan produktivitas dan kualitas hasil kerja keseluruhan dalam pengelolaan sumber daya developer.

$$\text{Fitness} = \frac{1}{1 + (\sum D + \sum A + \sum L + \sum K)}$$

Ket :

D = Bentrok developer diproyek yang sama

A = Penggunaan keahlian yang sesuai

L = Tidak adanya senior developer dalam proyek

K = Kompleksitas Proyek

Simulasi proses perhitungan fitness akan menggunakan data developer yang tercantum pada Tabel 3.1 dan data proyek pada Tabel 3.2. Simulasi ini akan menunjukkan proses perhitungan dari keempat fungsi tujuan yang telah ditetapkan.

Tabel 3. 6 Evaluasi Fitness Kromosom 1

<b>Proyek</b>	<b>Developer</b>	<b>Nilai</b>
P1	D1, D2	0
P2	D4, D5, D2	0

Fitness = 1

Pada table 3.6 memiliki nilai fitness yang tinggi karena memenuhi semua fungsi tujuan yang telah ditetapkan. Pada proyek P1 dan P2, tidak terjadi bentrokan developer, memiliki satu senior developer pada setiap proyek, setiap proyek dikerjakan oleh developer sesuai dengan bidang keahliannya dan pada proyek P2 memiliki tiga developer karena kompleksitas proyek yang tinggi.

Tabel 3. 7 Evaluasi Fitness Kromosom 2

<b>Proyek</b>	<b>Developer</b>	<b>Nilai</b>
P1	D3, D5	2
P2	D1, D2, D5	1

Fitness = 0.25

Pada Tabel 3.7, dapat dilihat bahwa pada proyek P1 tidak memiliki senior developer dan juga proyek dikerjakan oleh developer yang tidak sesuai dengan bidang keahliannya. Oleh karena itu proyek P1 mendapatkan nilai penalti sebesar 2. Hal ini menyebabkan penurunan nilai fitnessnya.

Tabel 3. 8 Evaluasi Fitness Kromosom 3

<b>Proyek</b>	<b>Developer</b>	<b>Nilai</b>
P1	D2, D3	2
P2	D5, D5, D1	1

Fitness = 0.25

Pada Tabel 3.8, terlihat bahwa proyek P1 tidak memiliki senior developer dan dikerjakan oleh developer yang tidak sesuai dengan bidang keahliannya, yang mengakibatkan nilai penalty sebesar 2. Hal ini menyebabkan penurunan nilai fitness proyek tersebut. Selain itu, pada Tabel 3.8, juga terlihat bahwa terdapat bentrok developer pada proyek P2, yang mengakibatkan nilai penalty sebesar 1, sehingga juga menyebabkan penurunan nilai fitnessnya.

Tabel 3. 9 Evaluasi Fitness Kromosom 4

<b>Proyek</b>	<b>Developer</b>	<b>Nilai</b>
P1	D2, D3	1
P2	D5, D2	2

Fitness = 0.25

Pada Tabel 3.9, dapat dilihat bahwa proyek P2 adalah proyek dengan tingkat kompleksitas tinggi akan tetapi pada tabel menunjukkan proyek tersebut hanya memiliki dua developer dan juga tidak memiliki senior developer sehingga menyebabkan mendapatkan nilai penalti sebesar 2. Hal ini menyebabkan penurunan nilai fitnessnya.

#### 3.4.4 Proses Seleksi

Setelah evaluasi fitness dilakukan, dilanjutkan dengan proses seleksi. Salah satu metode seleksi yang umum digunakan dalam algoritma genetika adalah proses

seleksi menggunakan turnamen. Dalam metode ini, seleksi dilakukan dengan mengadakan kompetisi kecil antara beberapa individu dalam sebuah turnamen.

Pertama, dari populasi saat ini, tiga individu secara acak dipilih untuk menjadi peserta turnamen. Dari ketiga individu tersebut, dipilih satu individu dengan nilai fitness tertinggi. Individu yang memiliki nilai fitness tertinggi ini kemudian akan dipilih untuk disalin ke generasi berikutnya, sehingga gen-gen mereka dapat diwariskan ke generasi mendatang. Proses seleksi menggunakan turnamen dengan ukuran turnamen 3 memungkinkan variasi dalam pemilihan individu, yang dapat membantu dalam mempertahankan keragaman genetik dalam populasi.

Tabel 3. 10 Seleksi Kromosom 1

<b>Proyek</b>	<b>Developer</b>
P1	D1, D2
P2	D4, D5

Tabel 3. 11 Seleksi Kromosom 2

<b>Proyek</b>	<b>Developer</b>
P1	D3, D5
P2	D1, D2

### 3.4.5 Proses Crossover

Proses crossover melibatkan dua induk untuk membentuk kromosom baru. Pindah silang menghasilkan keturunan baru dalam ruang pencarian yang siap diuji. Sebuah titik pada kromosom kedua induk diambil secara acak, dan ditetapkan sebagai 'titik persilangan'. Bagian di sebelah kanan titik tersebut dipertukarkan di antara kedua kromosom induk. Hal ini menghasilkan dua keturunan, masing-masing membawa sejumlah informasi genetik dari kedua induknya. Pada setiap titik

persilangan, kesempatan untuk melakukan crossover dapat ditentukan oleh crossover rate, yang biasanya diekspresikan sebagai persentase, misalnya 80%, yang menunjukkan probabilitas bahwa crossover akan terjadi pada setiap pasangan induk.

Tabel 3. 12 Crossover kromosom 1 individu pertama

<b>Proyek</b>	<b>Developer</b>	<b>Hasil</b>
P1	D1, D2	D3, D5
P2	D4, D5	D4, D5

Tabel 3. 13 Crossover kromosom 2 individu pertama

<b>Proyek</b>	<b>Developer</b>	<b>Hasil</b>
P1	D3, D5	D1, D2
P2	D1, D2	D1, D2

Dalam Tabel 3.12 dan 3.13 di atas, titik persilangan yang ditentukan adalah pada developer ke-1 dari setiap kromosom. Selanjutnya, setiap developer pada proyek P1 pada kromosom pertama akan menempati proyek P1 pada kromosom kedua, dan begitu pula sebaliknya.

### 3.4.6 Proses Mutasi

Proses mutasi untuk optimasi dimulai dengan iterasi melalui setiap proyek dalam pembagian tim. Selanjutnya, untuk setiap developer dalam proyek tersebut, dilakukan pengambilan angka acak antara 0 dan 1. Jika angka acak tersebut lebih kecil dari tingkat mutasi yang telah ditentukan yaitu 0.4, developer tersebut dihapus dari posisi awalnya dalam tim. Kemudian, developer baru dipilih secara acak untuk penempatan developer yang telah dihapus. Setelah itu, proyek disisipkan kembali ke posisi baru dalam list tim. Proses ini dilakukan untuk setiap developer dalam

proyek, dengan tujuan memperkenalkan variasi ke dalam populasi solusi dan meningkatkan kemungkinan menemukan solusi baru yang optimal atau mendekati optimal dalam jangka waktu yang lebih cepat.

Tabel 3. 14 Proses mutasi kromosom 1

<b>Proyek</b>	<b>Developer</b>	<b>Hasil</b>
P1	D3, D5	D3, D5
P2	D4, D5	D4, D5

Tabel 3. 15 Proses mutasi kromosom 2

<b>Proyek</b>	<b>Developer</b>	<b>Hasil</b>
P1	D1, D2	D1, D2
P2	D1, D2	D1, D4

Pada Tabel 3.14, dapat dilihat bahwa hasil mutasi populasi pertama tidak berubah, sedangkan pada Tabel 3.15, pada proyek P1 terjadi perubahan. Ini disebabkan karena populasi pertama tidak melalui tingkat mutasi yang ditetapkan, sedangkan pada populasi kedua telah melewati tingkat mutasi yang ditetapkan. Maka, pada proyek P1, developer akan dipilih secara acak, lalu akan digantikan dengan developer baru secara acak.

### 3.4.7 Hasil

Setelah semua proses telah dilakukan maka akan dilakukan Kembali proses evaluasi fitness. Kemudian akan dipilih kembali individu terbaik setelahnya.

Tabel 3. 16 Hasil kromosom 1 iterasi pertama

<b>Proyek</b>	<b>Developer</b>	<b>Nilai</b>
P1	D3, D5	1
P2	D4, D5	0

Fitness = 0.5

Tabel 3. 17 Hasil kromosom 2 iterasi pertama

<b>Proyek</b>	<b>Developer</b>	<b>Nilai</b>
P1	D1, D2	0
P2	D1, D4	2

Fitness = 0.33

Hasil dari proses algoritma genetika menghasilkan kromosom terbaik yang didapatkan yaitu kromosom 1 dengan nilai fitness 0.5.

### 3.5 Skenario Uji Coba

Bab Pengujian ini dilakukan dengan tujuan untuk mengetahui implementasi algoritma genetika yang telah dibuat sudah berjalan dengan baik atau belum. Program dibuat menggunakan Bahasa pemrograman PHP. Proses pengujian dilakukan dengan cara melakukan pengimplementasikan algoritma genetika dengan mengubah nilai threshold dari mutasi dan crossover lalu menghitung seberapa banyak populasi yang dihasilkan dengan algoritma genetika sesuai dengan data populasi yang dibuat oleh manusia.

Tahapan implementasi dalam pembuatan sistem dilakukan dengan tahapan pengiputan data, pembuatan populasi, perhitungan fitness, seleksi populasi terbaik. Selanjutnya pada proses crossover dan mutasi tingkat threshold dari kedua proses tersebut akan diubah dengan jumlah yang beragam antara 0 sampai dengan 1. Proses pengujian dalam penelitian ini dilaksanakan pada setiap iterasi dengan mengikuti skenario yang dijelaskan pada tabel di bawah ini.

Tabel 3. 18 Skenario uji coba dengan pembagian data latih dan uji

<b>Uji Coba</b>	<b>Crossover</b>	<b>Mutasi</b>	<b>Keterangan</b>
1	0.2	0.2	Tingkat Threshold crossover 0.2, sedangkan threshold mutasi sebesar 0.2
2	0.2	0.4	Tingkat Threshold crossover 0.2, sedangkan threshold mutasi sebesar 0.4
3	0.4	0.2	Tingkat Threshold crossover 0.4, sedangkan threshold mutasi sebesar 0.2
4	0.4	0.4	Tingkat Threshold crossover 0.4, sedangkan threshold mutasi sebesar 0.4
5	0.6	0.2	Tingkat Threshold crossover 0.6, sedangkan threshold mutasi sebesar 0.2
6	0.6	0.4	Tingkat Threshold crossover 0.6, sedangkan threshold mutasi sebesar 0.4

Hasil dari setiap uji coba akan dihitung recognition rate-nya dengan cara membandingkan data asli dari setiap proyek dengan lima data hasil algoritma genetika. Jika salah satu data hasil algoritma genetika sama dengan data asli, maka prediksi tersebut dinilai benar. Recognition rate dihitung dengan menghitung jumlah prediksi benar dan membandingkannya dengan jumlah keseluruhan data yang ada. Dengan kata lain, recognition rate menunjukkan persentase prediksi yang benar dari total data yang diuji. Berikut adalah rumus yang digunakan dalam menentukan recognition rate dari setiap uji coba, yang memberikan gambaran mengenai akurasi algoritma genetika dalam menyelesaikan tugas yang diberikan.

$$\text{Recognition rate} = \frac{B}{P}$$

Ket :

B = Total proyek dengan prediksi benar

P = Total semua proyek yang ada

### 3.6 Implementasi

Pada bagian ini akan dilakukan pengiputan data, pembuatan populasi, perhitungan fitness, seleksi populasi terbaik, crossover dan mutasi menggunakan Bahasa pemrograman php dan framework Laravel.

#### 3.6.1 Inisialisasi Kromosom

Data yang sudah di masukan selanjutnya akan di iterasi sebanyak 100 kali untuk membuat populasi awal. Setiap populasi di isi data proyek yang telah di masukan. Selanjutnya data proyek tersebut diiterasi dan di setiap iterasi proyek tersebut di isi developer secara acak.

```

$population = [];

for ($i = 0; $i < $this->populationSize; $i++) {
    $teams = [];

    for ($i = 0; $i < $this->num; $i++) {
        $remainingKuota = $this->ProjectLenght;
        $availableSlot = array_keys($this->classHours);
        while ($remainingKuota > 2 ) {
            $randomDev = $availableSlot[array_rand($availableSlot)];

            if ($this->classHours[$randomDev][0] <= $remainingKuota) {
                $teams[$this->projek[$i]][] = $randomDev;
                $remainingKuota -= $this->classHours[$randomDev][0];
            }
        }
    }
    $population[] = $teams;
}
return $population;

```

Gambar 3. 2 Inisialisasi Populasi

Pada Gambar 3.2 di atas menunjukkan kode yang akan melakukan iterasi sejumlah besar populasi yang sudah ditentukan. Kemudian, setiap populasi akan melakukan iterasi sejumlah proyek yang ada, lalu secara acak akan diisi dengan developer.

### 3.6.2 Evaluasi Fitness

Setelah proses inialisasi populasi selesai setiap kromosom di evaluasi menggunakan proses fitness dengan tujuan mengukur seberapa bagus solusi pada setiap kromosom. Berdasarkan masalah yang akan di selesaikan dalam pembuatan tim proyek maka pada penelitian ini dibuat beberapa fungsi fitness. Fungsi fitness pertama adalah mengecek apakah dalam satu proyek ada developer yang duplikat ditunjukkan pada gambar 3.3. Semakin banyak duplikat maka akan mengurangi nilai fitness.

```
$conflictCount = 0;
foreach ($schedule as $project => $developer) {
    $classCount = count($developer);
    $uniqueDeveloper = array_unique($developer);
    $uniqueDeveloperCount = count($uniqueDeveloper);
    $conflictCount += $classCount - $uniqueDeveloperCount;
}
$fitness += $conflictCount;
```

Gambar 3. 3 Fungsi fitness untuk menghitung duplikat developer

Fungsi Fitness selanjutnya menghitung skala proyek dan developer yang di distribusikan didalamnya. Semakin besar proyeknya maka semakin banyak developer yang dibutuhkan dan begitu pula sebaliknya. Pada fungsi ini setiap proyek akan di iterasi dan pada setiap iterasi akan di cek bobot proyek yang diberikan dan jumlah developer yang ada.

```
$developerCountsPerProject = [];
foreach ($schedule as $project => $developer) {
    $developerCountsPerProject[$project] = count($developer);
}
$maxProjectCount = max($developerCountsPerProject);
$minProjectCount = min($developerCountsPerProject);
$fitness += ($maxProjectCount - $minProjectCount);
```

Gambar 3. 4 Fungsi fitness kesesuaian proyek dengan developer

Fungsi fitness selanjutnya akan melakukan pengecekan pada setiap tim yang terbentuk dalam populasi. Pada tahap ini, dilakukan evaluasi terhadap setiap tim atau proyek dalam populasi. Dua kriteria yang diperiksa adalah keberadaan senior developer dan jumlah senior developer yang terlibat dalam satu proyek. Pada setiap proyek atau tim, fungsi fitness memeriksa apakah terdapat senior developer yang terlibat. Selain itu, fungsi fitness juga memeriksa apakah terdapat lebih dari satu senior developer yang terlibat dalam satu proyek. Meskipun kehadiran senior developer penting, memiliki terlalu banyak senior developer dalam satu proyek mungkin tidak efisien secara sumber daya.

```

foreach ($teams as $project => $devs) {
    $lead = 0;
    foreach ($devs as $dev) {
        if($this->devTier[$dev][0] == 4){
            $lead +=1;
        }
    }
    if($lead != 1){
        $fitness += 5;
    }
}

```

Gambar 3. 5 Fungsi fitness Penggunaan Senior Developer

Fungsi tujuan selanjutnya adalah memastikan bahwa penggunaan developer sesuai dengan keahliannya untuk setiap proyek yang diberikan. Ini dilakukan dengan mengiterasi setiap tim dan kemudian mengiterasi setiap developer dalam tim tersebut. Selanjutnya, dilakukan pengecekan apakah setiap developer dalam proyek tersebut sesuai dengan spesifikasi yang dibutuhkan pada proyek tersebut.

```

foreach ($teams as $proyek => $devs) {
    foreach ($devs as $dev) {
        if($this->projek_var[$proyek][0] != $this->projectDevs[$dev][1]){
            $fitness += 5;
        }
        if($this->projek_var[$proyek][1] == 0){
            if($this->projectDevs[$dev][2] != 0){
                $fitness += 3;
            }
        }
        if($this->projek_var[$proyek][1] == 1){
            if($this->projectDevs[$dev][2] != 1){
                $fitness += 3;
            }
        }

        if($this->projek_var[$proyek][1] == 2){
            if ($this->projectDevs[$dev][2] != 2){
                $fitness += 1;
            }
        }

        if($this->projek_var[$proyek][1] == 3){
            if ($this->projectDevs[$dev][2] != 3){
                $fitness += 1;
            }
        }
    }
}
}

```

Gambar 3. 6 Fungsi fitness Penggunaan Keahlian Developer

Fungsi tujuan selanjutnya adalah memastikan setiap proyek mendapatkan developer sesuai dengan kompleksitas setiap proyek. Ini dilakukan dengan mengiterasi setiap tim dan kemudian mengiterasi setiap developer dalam tim tersebut. Selanjutnya, dilakukan pengecekan apakah developer-developer yang mengerjakan proyek tersebut memenuhi syarat minimal proyek dengan cara semakin kompleks proyek tersebut maka semakin banyak developer yang diperlukan.

```

$totalComp = $numProject * $projectComp;
$devUsed = 0;
foreach ($schedule as $key => $teams) {
    foreach ($teams as $dev) {
        $devUsed += $this->Project[$dev][0];
    }
}
$unusedHours = $totalComp - $devUsed;

```

Gambar 3. 7 Fungsi fitness Kompleksitas Proyek

### 3.6.3 Proses Seleksi

Proses seleksi dilakukan dengan mengiterasi 100 populasi lalu memilih kromosom secara acak lalu dimasukkan ke dalam array sebanyak yang diinginkan, pada penelitian ini akan diambil 5 kromosom di setiap populasi. Selanjutnya dari 5 kromosom yang dipilih akah di urutkan berdasarkan nilai fitnessnya dan yang memiliki nilai fitness tertinggi akan dimasukkan kedalam array populasi baru.

```

$parents = [];
$populationSize = count($population);

while (count($parents) < $populationSize) {
    // Randomly select participants for the tournament
    $participants = [];
    for ($i = 0; $i < $tournamentSize; $i++) {
        $randomIndex = rand(0, $populationSize - 1);
        $participants[] = $population[$randomIndex];
    }

    // Sort the participants by their fitness scores
    usort($participants, function ($a, $b) {
        return $this->calculateFitness($a) <=> $this->calculateFitness($b);
    });

    // Select the fittest individual as a parent
    $parents[] = $participants[0];
}

return $parents;

```

Gambar 3. 8 Proses Seleksi

Kode pada Gambar 3.8 menjelaskan bahwa populasi akan melakukan iterasi setiap individu dalam populasi. Pada setiap iterasi, beberapa individu akan diambil secara acak, lalu akan dilakukan turnamen. Pemenang dari turnamen tersebut akan diurutkan dengan nilai fitness tertinggi berada di bagian atas, sehingga individu-individu pada posisi atas akan digunakan pada generasi berikutnya.

### 3.6.4 Proses Crossover

Selanjutnya adalah proses crossover dengan cara menentukan 2 pasangan kromosom yang terpilih atau disebut parent untuk membuat solusi baru. Hal ini biasanya dilakukan dengan menukar bagian materi genetik (gen) untuk menghasilkan kandidat solusi baru. Harapannya, keturunannya akan mewarisi sifat-sifat terbaik induknya.

```

$offspring = [];

$crossoverDay = rand(1, count($parent1));

for ($day = 0; $day < count($parent1); $day++) {
    $random = mt_rand() / mt_getrandmax();
    $crossoverRate = 0.2; // Adjust the crossover rate as desired
    if ($random < $crossoverRate) {
        if ($day <= $crossoverDay) {
            $offspring[$this->projek[$day]] = $parent1[$this->projek[$day]];
        } else {
            $offspring[$this->projek[$day]] = $parent2[$this->projek[$day]];
        }
    }
}

return $offspring;

```

Gambar 3. 9 Proses Crossover

Tahapan crossover dimulai dengan memilih 2 induk yang akan disilangkan. Selanjutnya ditentukan crossover rate untuk menentukan kemungkinan terjadinya

crossover. Induk yang melewati crossover rate akan menukar salah satu kromosom yang terpilih dan menghasilkan solusi baru yang diharapkan lebih baik.

### 3.6.5 Proses Mutasi

Mutasi merupakan proses perubahan yang terjadi bersifat acak. Perubahan ini dapat terjadi kapan saja pada setiap individu dengan kemungkinan tertentu. Mutasi rate menentukan kemungkinan terjadinya mutasi pada setiap kromosom pada suatu individu.

```

$mutatedSchedule = $schedule;

for ($day = 0; $day < count($mutatedSchedule)-1; $day++) {
    foreach ($mutatedSchedule[$this->projek[$day]] as $index => $class) {
        $random = mt_rand() / mt_getrandmax();
        $mutationRate = 0.2; // Adjust the mutation rate as desired
        if ($random < $mutationRate) {
            $removedClass = array_splice($schedule[$this->projek[$day]], 1);
            $newDay = rand(1, $this->numDays-1);

            array_splice($mutatedSchedule[$this->projek[$newDay]], 1, null, $removedClass);
        }
    }
}

return $mutatedSchedule;

```

Gambar 3. 10 Proses Mutasi

Tujuan utama dari mutasi adalah untuk memperkenalkan genetik baru ke dalam populasi. Sementara seleksi dan crossover cenderung menggunakan individu dengan fitness tertinggi, mutasi mengeksplorasi area baru dengan menggunakan perubahan acak.

## BAB IV

### UJI COBA DAN PEMBAHASAN

#### 4.1 Hasil Uji Coba

Beberapa skenario uji coba pada penelitian ini dilakukan untuk mengetahui variasi nilai akurasi yang ada pada metode pengklasifikasian yang akan diuji. Hasil uji coba tingkat akurasi pada skenario pengujian yang dilakukan mulai dari iterasi pertama hingga iterasi ke-tiga dikelompokkan dalam tabel 3.18.

##### 4.1.1 Uji Coba Skenario 1

Pada uji coba skenario pertama dilakukan dengan tingkat kemungkinan crossover 0.2 dan tingkat kemungkinan mutasi 0.2 yang artinya ketika proses crossover dan mutasi terjadi, setiap individu memiliki kemungkinan 20% crossover dan 20% mutasi akan dijalankan. Dari setiap proyek dipilih 5 populasi teratas yang memiliki nilai fitness tertinggi, kemudian jika salah satu individu dari sebuah populasi sama dengan data sebenarnya maka prediksi dinilai benar.

Tabel 4. 1 Hasil Prediksi pada proyek APM

<b>Populasi</b>	<b>Developer</b>	<b>Fitness</b>
1	Yayak Z, Macmud E	0.0044
2	M Zainul, Dinda, Yudanty	0.0040
3	Rizky Sena, M Wahyu	0.0040
4	Achmad Uwais, Dimas, M Rizky	0.0039
5	Edwin, M Rizky	0.0039
6	M Rizky, M Zainul	0.0039
7	Edwin, Ibnu, M Rizky	0.0038
...	...	
50	M Zainul, Adi N	0.0028

Tabel 4.1 menunjukkan daftar proyek APM yang telah diurutkan dengan nilai fitness terbesar di bagian atas. Dari daftar populasi tersebut, dipilih lima

proyek dengan nilai fitness tertinggi. Dari kelima proyek yang terpilih, terdapat salah satu populasi yang memiliki daftar developer yang sama dengan data asli pada proyek APM. Oleh karena itu, prediksi proyek APM pada uji coba 1 dinilai benar.

Tabel 4. 2 Hasil uji coba skenario 1

<b>Proyek</b>	<b>Developer</b>	<b>Prediksi</b>	<b>Prediksi</b>
APM	Rizky Sena, M Wahyu	Rizky Sena, M Wahyu	Benar
HIS-VMS	M Thoha, M. Rizky A	M Thoha, M. Rizky A	Benar
APW-ERP	M Wahyu A, Wildan S	Yayak Wildan S	Salah
WVI	Masfuul, M Thoha	M. Rizky A, M Thoha	Salah
...	...	...	...
Total			21

Setelah semua proyek diperiksa untuk menentukan apakah lima populasi teratas memiliki developer yang sama dengan data proyek yang asli, diperoleh hasil uji coba seperti yang ditunjukkan pada Tabel 4.2. Dari tabel tersebut, terlihat bahwa dari 30 data proyek, terdapat 21 data yang bernilai benar. Hasil prediksi dari klasifikasi uji coba di atas kemudian akan digunakan untuk menghitung nilai recognition rate dengan rumus berikut.

$$Recognition\ Rate\ (sk1) = \frac{21}{30} = 0.7$$

#### 4.1.2 Uji Coba Skenario 2

Pada uji coba skenario pertama dilakukan dengan tingkat kemungkinan crossover 0.2 dan tingkat kemungkinan mutasi 0.4 yang artinya ketika proses crossover dan mutasi terjadi, setiap individu memiliki kemungkinan 20% crossover dan 40% mutasi akan dijalankan. Dari setiap proyek dipilih 5 populasi teratas yang memiliki nilai fitness tertinggi, kemudian jika salah satu individu dari sebuah populasi sama dengan data sebenarnya maka prediksi dinilai benar.

Tabel 4. 3 Hasil Prediksi pada proyek APM

<b>Populasi</b>	<b>Developer</b>	<b>Fitness</b>
1	Adi N, Annisa A.K	0.0044
2	Adi N, M. Rizky A	0.0040
3	M Wahyu A, Rosa K	0.0040
4	Dinda A, Wildan S	0.0039
5	Melia, M Zainul	0.0039
6	Rizky Sena, M Wahyu	0.0038
7	M Rizky, M Zainul	0.0038
...	...	
50	Dimas, Yayak	0.0030

Tabel 4.3 menunjukkan daftar proyek APM yang telah diurutkan dengan nilai fitness terbesar di bagian atas. Dari daftar populasi tersebut, dipilih lima proyek dengan nilai fitness tertinggi. Dari kelima proyek yang terpilih, terdapat salah satu populasi yang memiliki daftar developer yang sama dengan data asli pada proyek APM. Oleh karena itu, prediksi proyek APM pada uji coba 1 dinilai benar.

Tabel 4. 4 Hasil uji coba skenario 2

<b>Proyek</b>	<b>Developer</b>	<b>Prediksi</b>	<b>Prediksi</b>
APM	Rizky Sena, M Wahyu	Rizky Sena, M Wahyu	Benar
HIS-VMS	M Thoha, M. Rizky A	M Thoha, M. Rizky A	Benar
APW-ERP	M Wahyu A, Wildan S	Yayak Wildan S	Salah
WVI	Masfuul, M Thoha	Masfuul, M Thoha	Benar
...	...	...	...
Total			21

Setelah semua proyek diperiksa untuk menentukan apakah lima populasi teratas memiliki developer yang sama dengan data proyek yang asli, diperoleh hasil uji coba seperti yang ditunjukkan pada Tabel 4.4. Dari tabel tersebut, terlihat bahwa dari 30 data proyek, terdapat 24 data yang bernilai benar. Hasil prediksi dari klasifikasi uji coba di atas kemudian akan digunakan untuk menghitung nilai recognition rate dengan rumus berikut.

$$\text{Recognition Rate (sk1)} = \frac{21}{30} = 0.7$$

### 4.1.3 Uji Coba Skenario 3

Pada uji coba skenario pertama dilakukan dengan tingkat kemungkinan crossover 0.4 dan tingkat kemungkinan mutasi 0.2 yang artinya ketika proses crossover dan mutasi terjadi, setiap individu memiliki kemungkinan 40% crossover dan 20% mutasi akan dijalankan. Dari setiap proyek dipilih 5 populasi teratas yang memiliki nilai fitness tertinggi, kemudian jika salah satu individu dari sebuah populasi sama dengan data sebenarnya maka prediksi dinilai benar.

Tabel 4. 5 Hasil Prediksi pada proyek APM

Populasi	Developer	Fitness
1	Adi N, Yudanty, Dinda	0.0043
2	M Zainul, Achmad Uwais	0.0040
3	Rizky Sena, M Wahyu	0.0040
4	Dinda, Imamah A	0.0039
5	Dimas, M Thoha	0.0039
6	Tommy, Machmud Efendi	0.0039
7	Annisa, Machmud Efendi	0.0038
...	...	
50	Achmad Uwais, Dinda	0.0028

Tabel 4.5 menunjukkan daftar proyek APM yang telah diurutkan dengan nilai fitness terbesar di bagian atas. Dari daftar populasi tersebut, dipilih lima proyek dengan nilai fitness tertinggi. Dari kelima proyek yang terpilih, tidak terdapat populasi yang memiliki daftar developer yang sama dengan data asli pada proyek APM. Oleh karena itu, prediksi proyek APM pada uji coba 1 dinilai salah.

Tabel 4. 6 Hasil uji coba skenario 3

Proyek	Developer	Prediksi	Prediksi
APM	Rizky Sena, M Wahyu	Rizky Sena, M Wahyu	Benar
HIS-VMS	M Thoha, M. Rizky A	Dinda, M Thoha	Salah
APW-ERP	M Wahyu A, Wildan S	M Wahyu A, Wildan S	Benar
WVI	Masfuul, M Thoha	Masfuul, M Thoha	Benar

...	...	...	...
Total			25

Setelah semua proyek diperiksa untuk menentukan apakah lima populasi teratas memiliki developer yang sama dengan data proyek yang asli, diperoleh hasil uji coba seperti yang ditunjukkan pada Tabel 4.6. Dari tabel tersebut, terlihat bahwa dari 30 data proyek, terdapat 25 data yang bernilai benar. Hasil prediksi dari klasifikasi uji coba di atas kemudian akan digunakan untuk menghitung nilai recognition rate dengan rumus berikut.

$$\text{Recognition Rate (sk1)} = \frac{25}{30} = 0.83$$

#### 4.1.4 Uji Coba Skenario 4

Pada uji coba skenario pertama dilakukan dengan tingkat kemungkinan crossover 0.4 dan tingkat kemungkinan mutasi 0.4 yang artinya ketika proses crossover dan mutasi terjadi, setiap individu memiliki kemungkinan 40% crossover dan 40% mutasi akan dijalankan. Dari setiap proyek dipilih 5 populasi teratas yang memiliki nilai fitness tertinggi, kemudian jika salah satu individu dari sebuah populasi sama dengan data sebenarnya maka prediksi dinilai benar.

Tabel 4. 7 Hasil Prediksi pada proyek APM

Populasi	Developer	Fitness
1	Rizky Sena, M Wahyu	0.0045
2	Adi N, M. Rizky A	0.0042
3	M Wahyu A, Rosa K	0.0040
4	Adi N, Annisa A.K	0.0039
5	Dinda A, Rosa	0.0039
6	Adi N, Wildan S	0.0038
7	M Wahyu A, Rosa K	0.0038
...	...	
50	Yudanty, M Rizky	0.0030

Tabel 4.7 menunjukkan daftar proyek APM yang telah diurutkan dengan nilai fitness terbesar di bagian atas. Dari daftar populasi tersebut, dipilih lima proyek dengan nilai fitness tertinggi. Dari kelima proyek yang terpilih, terdapat salah satu populasi yang memiliki daftar developer yang sama dengan data asli pada proyek APM. Oleh karena itu, prediksi proyek APM pada uji coba 1 dinilai benar.

Tabel 4. 8 Hasil uji coba skenario 4

<b>Proyek</b>	<b>Developer</b>	<b>Prediksi</b>	<b>Prediksi</b>
APM	Rizky Sena, M Wahyu	Rizky Sena, M Wahyu	Benar
HIS-VMS	M Thoha, M. Rizky A	M Thoha, M. Rizky A	Benar
APW-ERP	M Wahyu A, Wildan S	Rosa K , Wildan S	Salah
WVI	Masfuul, M Thoha	Masfuul, M Thoha	Benar
...	...	...	...
Total			23

Setelah semua proyek diperiksa untuk menentukan apakah lima populasi teratas memiliki developer yang sama dengan data proyek yang asli, diperoleh hasil uji coba seperti yang ditunjukkan pada Tabel 4.8. Dari tabel tersebut, terlihat bahwa dari 30 data proyek, terdapat 23 data yang bernilai benar. Hasil prediksi dari klasifikasi uji coba di atas kemudian akan digunakan untuk menghitung nilai recognition rate dengan rumus berikut.

$$Recognition\ Rate\ (sk1) = \frac{23}{30} = 0.77$$

#### 4.1.5 Uji Coba Skenario 5

Pada uji coba skenario pertama dilakukan dengan tingkat kemungkinan crossover 0.6 dan tingkat kemungkinan mutasi 0.2 yang artinya ketika proses crossover dan mutasi terjadi, setiap individu memiliki kemungkinan 20% crossover dan 40% mutasi akan dijalankan. Dari setiap proyek dipilih 5 populasi teratas yang

memiliki nilai fitness tertinggi, kemudian jika salah satu individu dari sebuah populasi sama dengan data sebenarnya maka prediksi dinilai benar.

Tabel 4. 9 Hasil Prediksi pada proyek APM

<b>Populasi</b>	<b>Developer</b>	<b>Fitness</b>
1	Adi N, Annisa A.K	0.0042
2	Adi N, M. Rizky A	0.0040
3	Rizky Sena, M Wahyu	0.0040
4	Dinda A, Wildan S	0.0039
5	M Wahyu A, Rosa K	0.0039
6	M Wahyu A, Tommy	0.0039
7	Tommy, Adi N	0.0038
...	...	
50	Ibnu, Bima	0.0030

Tabel 4.9 menunjukkan daftar proyek APM yang telah diurutkan dengan nilai fitness terbesar di bagian atas. Dari daftar populasi tersebut, dipilih lima proyek dengan nilai fitness tertinggi. Dari kelima proyek yang terpilih, terdapat salah satu populasi yang memiliki daftar developer yang sama dengan data asli pada proyek APM. Oleh karena itu, prediksi proyek APM pada uji coba 1 dinilai benar.

Tabel 4. 10 Hasil uji coba skenario 5

<b>Proyek</b>	<b>Developer</b>	<b>Prediksi</b>	<b>Prediksi</b>
APM	Rizky Sena, M Wahyu	Rizky Sena, M Wahyu	Benar
HIS-VMS	M Thoha, M. Rizky A	M Thoha, M. Rizky A	Benar
APW-ERP	M Wahyu A, Wildan S	Ibnu, Rosa K	Salah
WVI	Masfuul, M Thoha	Masfuul, M Thoha	Benar
...	...	...	...
Total			24

Setelah semua proyek diperiksa untuk menentukan apakah lima populasi teratas memiliki developer yang sama dengan data proyek yang asli, diperoleh hasil uji coba seperti yang ditunjukkan pada Tabel 4.10. Dari tabel tersebut, terlihat bahwa dari 30 data proyek, terdapat 24 data yang bernilai benar. Hasil prediksi dari

klasifikasi uji coba di atas kemudian akan digunakan untuk menghitung nilai recognition rate dengan rumus berikut.

$$\text{Recognition Rate (sk1)} = \frac{24}{30} = 0.8$$

#### 4.1.6 Uji Coba Skenario 6

Pada uji coba skenario pertama dilakukan dengan tingkat kemungkinan crossover 0.6 dan tingkat kemungkinan mutasi 0.4 yang artinya ketika proses crossover dan mutasi terjadi, setiap individu memiliki kemungkinan 20% crossover dan 40% mutasi akan dijalankan. Dari setiap proyek dipilih 5 populasi teratas yang memiliki nilai fitness tertinggi, kemudian jika salah satu individu dari sebuah populasi sama dengan data sebenarnya maka prediksi dinilai benar.

Tabel 4. 11 Hasil Prediksi pada proyek APM

<b>Populasi</b>	<b>Developer</b>	<b>Fitness</b>
1	Rizky Sena, M Wahyu	0.0042
2	Melia D	0.0041
3	M Wahyu, Yayak	0.0040
4	Rizky Sena, M Wahyu	0.0039
5	M Inam, Rosa	0.0039
6	M Wahyu A, Rosa K	0.0039
7	Adi N, Adi N	0.0038
...	...	
50	Eko M	0.0028

Tabel 4.11 menunjukkan daftar proyek APM yang telah diurutkan dengan nilai fitness terbesar di bagian atas. Dari daftar populasi tersebut, dipilih lima proyek dengan nilai fitness tertinggi. Dari kelima proyek yang terpilih, terdapat salah satu populasi yang memiliki daftar developer yang sama dengan data asli pada proyek APM. Oleh karena itu, prediksi proyek APM pada uji coba 1 dinilai benar.

Tabel 4. 12 Hasil uji coba skenario 6

<b>Proyek</b>	<b>Developer</b>	<b>Prediksi</b>	<b>Prediksi</b>
APM	Rizky Sena, M Wahyu	Rizky Sena, M Wahyu	Benar
HIS-VMS	M Thoha, M. Rizky A	M Thoha, M. Rizky A	Benar
APW-ERP	M Wahyu A, Wildan S	M Wahyu A, Wildan S	Benar
WVI	Masfuul, M Thoha	Eko M, Rizky Sena	Salah
...	...	...	...
Total			21

Setelah semua proyek diperiksa untuk menentukan apakah lima populasi teratas memiliki developer yang sama dengan data proyek yang asli, diperoleh hasil uji coba seperti yang ditunjukkan pada Tabel 4.12. Dari tabel tersebut, terlihat bahwa dari 30 data proyek, terdapat 21 data yang bernilai benar. Hasil prediksi dari klasifikasi uji coba di atas kemudian akan digunakan untuk menghitung nilai recognition rate dengan rumus berikut.

$$Recognition\ Rate\ (sk1) = \frac{21}{30} = 0.7$$

## 4.2 Pembahasan

Pembahasan pada sub bab ini merupakan hasil yang didapat dari hasil uji coba yang telah dilakukan menggunakan metode algoritma genetika dengan mengubah nilai crossover dan 44rofes. Uji 44rofessi bertujuan untuk mengevaluasi kinerja algoritma dalam berbagai kondisi dan parameter yang berbeda, sehingga dapat menentukan konfigurasi yang paling optimal untuk menyelesaikan masalah yang diberikan.

Tabel 4. 13 Recognition Rate Hasil Pengujian Skenario 1 – 6

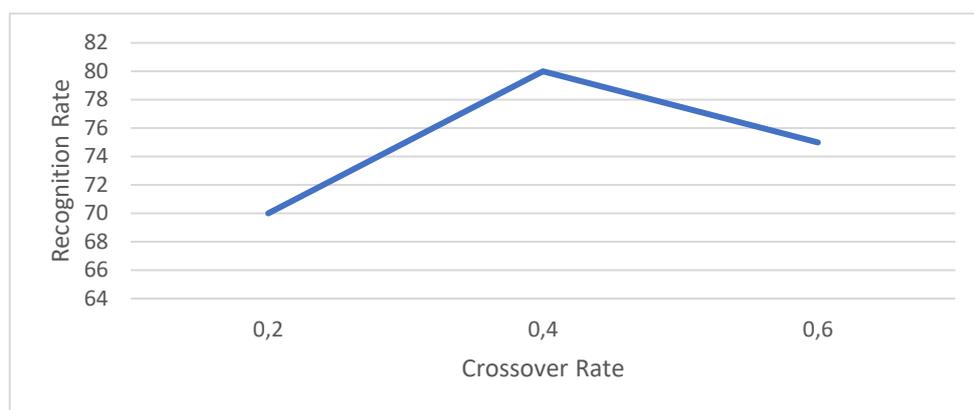
No	Pengujian	Recognition Rate
1	Skenario 1	70%
2	Skenario 2	70%
3	Skenario 3	83%
4	Skenario 4	77%
5	Skenario 5	80%
6	Skenario 6	70%

Crossover rate dan mutasi rate yang tepat adalah kunci untuk mencapai keseimbangan antara eksplorasi (mencari ruang solusi baru) dan eksploitasi (memperbaiki solusi yang ada). Dalam skenario yang diteliti, tingkat crossover sebesar 0,4 berarti ada peluang 40% setiap kromosom dalam individu akan mengalami crossover. Proses crossover ini penting untuk memastikan bahwa algoritma genetika dapat menggabungkan dan menguji kombinasi genetik yang berbeda, sehingga menemukan solusi yang lebih baik dari generasi ke generasi.

Di sisi lain, tingkat mutasi sebesar 0,2 berarti ada peluang 20% setiap kromosom dalam individu akan mengalami mutasi. Mutasi ini memberikan variasi tambahan yang diperlukan untuk mencegah algoritma terjebak dalam solusi lokal yang suboptimal. Namun, tingkat mutasi yang tepat harus dipilih dengan hati-hati. Mutasi rate yang terlalu tinggi dapat menyebabkan solusi menjadi acak dan kehilangan struktur yang baik, membuat algoritma sulit untuk menemukan solusi optimal karena solusi yang baik sering kali hancur oleh mutasi yang berlebihan.

Dengan kombinasi crossover rate 0,4 dan mutasi rate 0,2, algoritma berhasil mencapai keseimbangan yang tepat. Ini memberikan variasi yang cukup untuk mengeksplorasi ruang pencarian tanpa terlalu mengganggu struktur solusi yang baik. Oleh karena itu, kombinasi parameter ini memungkinkan algoritma genetika untuk

mengeksplorasi berbagai kemungkinan solusi secara efektif sekaligus memfokuskan perbaikan pada solusi yang sudah ditemukan, yang pada akhirnya menghasilkan performa yang lebih baik.



Gambar 4. 1 Grafik pengaruh crossover pada recognition rate

Analisis pada Grafik pada Tabel 4.1 mengungkapkan dampak yang signifikan dari variasi nilai crossover terhadap rata-rata recognition rate. Crossover rate mengatur seberapa sering individu dalam populasi bereproduksi dan bertukar informasi genetik. Dari grafik, terlihat bahwa crossover dengan nilai 0,4 menonjol sebagai pilihan yang optimal dengan recognition rate tertinggi. Hal ini menandakan bahwa keberadaan nilai crossover yang moderate, di tengah-tengah rentang kemungkinan nilai, cenderung memberikan hasil yang lebih baik. Crossover rate yang terlalu rendah dapat menghambat kemampuan algoritma untuk mengeksplorasi ruang pencarian secara luas, sedangkan crossover rate yang terlalu tinggi dapat menyebabkan konvergensi yang cepat ke solusi yang suboptimal.

Pentingnya pemilihan nilai crossover yang tepat terletak pada kemampuannya untuk menjaga keseimbangan antara eksplorasi (mencari solusi

baru) dan eksploitasi (mengoptimalkan solusi yang ada). Ketika nilai crossover berada pada titik tengah seperti 0,4, algoritma dapat menjelajahi berbagai solusi potensial tanpa mengorbankan stabilitas atau keandalan hasil. Sebaliknya, nilai crossover yang terlalu rendah mungkin gagal memanfaatkan variasi genetik yang cukup, sementara nilai yang terlalu tinggi dapat mempercepat konvergensi ke solusi yang kurang optimal. Oleh karena itu, pemilihan crossover rate yang sesuai dapat memberikan kontribusi yang signifikan dalam meningkatkan kinerja dan akurasi algoritma genetika dalam konteks pengelolaan sumber daya manusia.

Hasil evaluasi telah dilakukan pada uji coba skenario pertama hingga keenam dengan menggunakan tingkat crossover dan mutasi yang berbeda-beda. Dari hasil evaluasi tersebut, dapat dilihat bahwa hasil dari skenario ketiga, yang menggunakan tingkat crossover sebesar 0,2 dan tingkat mutasi sebesar 0,4, menunjukkan performa yang cukup signifikan dengan nilai sebesar 83%. Hal ini menunjukkan bahwa kombinasi parameter tersebut memberikan hasil yang optimal dibandingkan dengan skenario lainnya yang diuji coba.

### **4.3 Integrasi Al-Qur-an Dan Hadits**

Dengan adanya sistem yang dibuat diharapkan dapat membantu proses pengelolaan sumber daya developer dengan tujuan dapat mengurangi biaya, waktu dan tenaga oleh pihak perusahaan. Sistem yang dibangun diharapkan dapat bermanfaat dan menjadi model evaluasi bagi pihak perusahaan. Penggunaan tenaga ahli yang tepat sangat penting dalam mencapai tujuan strategis dan operasional. Dalam Islam memilih tenaga ahli yang tepat merupakan sesuatu yang penting, sebagaimana firman Allah *subhanahu wa ta'ala* dalam QS Al-Qashash ayat 26

قَالَتْ إِحْدَاهُمَا يَا أَبَتِ اسْتَجِرْهُ إِنَّ خَيْرَ مَنِ اسْتَجَرْتَ الْقَوِيُّ الْأَمِينُ

*“Ya bapakku ambillah ia sebagai orang yang bekerja (pada kita), karena sesungguhnya orang yang paling baik yang kamu ambil untuk bekerja (pada kita) ialah orang yang kuat lagi dapat dipercaya”.*(Q.S Al-Qashash : 26)

Menurut tafsir Al-Muyassar maksud dari ayat pada potongan ayat QS Al-Qashash ayat 26 tersebut adalah Salah seorang 48rofes itu berkata kepada bapaknya, “Wahai ayah, jadikanlah ia orang yang bekerja menggembala ternak untukmu. Sesungguhnya sebaik-baik orang yang engkau pekerjakan untuk menggembalakan ternak adalah orang yang kuat untuk menjaga ternakmu lagi dapat dipercaya yang engkau tidak khawatir ia akan berkhianat dalam urusan yang engkau percayakan kepadanya.”. Dari tafsir ayat tersebut dapat disimpulkan bahwa pekerja harus memiliki dua sifat penting, yaitu kuat dan 48rofes. Kuat dalam ayat diatas dapat dimaknai setidaknya tiga hal, yaitu kapabilitas keahlian, psikis, dan kepribadian. Sifat-sifat ini akan memastikan bahwa tugas yang diberikan dikelola dengan efisien dan tanpa risiko pengkhianatan atau kelalaian.

Ayat ini menekankan pentingnya memilih orang yang tepat untuk bekerja, yang memiliki kekuatan (kompetensi) dan dapat dipercaya (integritas). Dalam konteks penelitian ini, ayat ini sangat relevan dengan prinsip penggunaan tenaga ahli yang tepat dalam pengelolaan sumber daya developer. Dalam sistem kecerdasan buatan untuk pengelolaan sumber daya developer menggunakan algoritma genetika, prinsip ini diterapkan dengan cara memastikan bahwa setiap proyek dikerjakan oleh developer yang memiliki keahlian dan pengalaman yang sesuai. Evaluasi developer dilakukan berdasarkan kriteria yang ketat untuk

memastikan bahwa mereka memiliki kompetensi teknis dan kualitas yang diperlukan untuk menyelesaikan proyek dengan sukses.

Integrasi QS. Al-Qashash ayat 26 dalam penelitian ini menegaskan pentingnya penempatan developer yang tepat berdasarkan keahlian dan integritas mereka. Sistem kecerdasan buatan yang dikembangkan bertujuan untuk memastikan bahwa setiap proyek mendapat tenaga ahli yang paling sesuai, sebagaimana diajarkan dalam ayat tersebut. Dengan demikian, diharapkan proyek-proyek dapat diselesaikan dengan hasil yang optimal dan sesuai dengan prinsip-prinsip yang diajarkan dalam Al-Qur'an.

#### 4.3.1 *Hablum Minallah*

*Hablum Minallah*, yang berarti hubungan atau interaksi dengan Allah, mencakup aspek ibadah dan ketaatan dalam menjalankan aktivitas sehari-hari termasuk dalam manajemen sumber daya. Dalam konteks penelitian ini, kita dapat mengacu pada QS An-Nahl ayat 90 yang menekankan pentingnya keadilan, ihsan (kebaikan), dan memberikan hak-hak kepada yang berhak sebagai prinsip dalam manajemen.

إِنَّ اللَّهَ يَأْمُرُ بِالْعَدْلِ وَالْإِحْسَانِ وَإِيتَاءِ ذِي الْقُرْبَىٰ وَيَنْهَىٰ عَنِ الْفَحْشَاءِ وَالْمُنْكَرِ وَالْبَغْيِ ۗ يَعِظُكُمْ لَعَلَّكُمْ تَذَكَّرُونَ

*“Sesungguhnya Allah menyuruh (kamu) berlaku adil dan berbuat 49rofessio, memberi kepada kaum kerabat, dan Dia melarang (melakukan) perbuatan keji, kemungkaran dan permusuhan. Dia memberi pengajaran kepadamu agar kamu dapat mengambil 49rofessio.”(Q.S An-Nahl : 90)*

Tafsir Quraish Shihab mengartikan ayat ini dengan “Allah memerintahkan para hamba-Nya untuk berlaku adil dalam setiap perkataan dan perbuatan. Allah

menyuruh mereka untuk selalu berusaha menuju yang lebih baik dalam setiap usaha dan mengutamakan yang terbaik dari lainnya. Allah memerintahkan mereka untuk memberikan apa yang dibutuhkan oleh para kerabat sebagai cara untuk memperkokoh ikatan kasih 50rofes antar keluarga. Allah melarang mereka berbuat dosa, lebih-lebih dosa yang amat buruk dan segala perbuatan yang tidak dibenarkan oleh syariat dan akal sehat. Allah melarang mereka menyakiti orang lain. Dengan perintah dan larangan itu, Allah bermaksud membimbing kalian menuju kemaslahatan dalam setiap aspek kehidupan, agar kalian selalu ingat karunia-Nya dan menaati firman-firman-Nya.” (Quraish Shihab).

Ayat ini mengajarkan pentingnya keadilan dan kebaikan dalam segala aspek kehidupan, termasuk dalam pengelolaan sumber daya manusia. Prinsip ini dapat diterapkan dalam penelitian ini sebagai berikut: Pertama, keadilan dalam penempatan developer sangat penting. Menempatkan developer pada proyek yang sesuai dengan keahlian dan pengalaman mereka memastikan setiap individu dapat bekerja secara optimal dan merasa dihargai. Ini mencerminkan prinsip keadilan dalam pengelolaan sumber daya. Kedua, ihsan (kebaikan) harus diterapkan dalam pengelolaan sumber daya dengan cara yang baik dan bijak, memastikan bahwa setiap 50rofessio yang diambil tidak hanya efisien tetapi juga memperhatikan kesejahteraan dan perkembangan 50rofessional setiap developer. Ketiga, pemberian hak-hak kepada developer adalah esensial, termasuk peluang untuk berkembang, penghargaan yang sesuai dengan kontribusi, dan lingkungan kerja yang kondusif.

### 4.3.2 *Hablum Minannas*

*Hablum Minannas* dalam konteks penelitian ini, prinsip tersebut dapat dihubungkan dengan pengelolaan sumber daya developer yang efektif dan efisien. QS Al-Baqarah ayat 286 menyatakan bahwa Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya, yang berarti setiap orang harus diberi tugas yang sesuai dengan kemampuan dan keahliannya.

لَا يُكَلِّفُ اللَّهُ نَفْسًا إِلَّا وُسْعَهَا ۗ لَهَا مَا كَسَبَتْ وَعَلَيْهَا مَا اكْتَسَبَتْ ۗ رَبَّنَا لَا تُؤَاخِذْنَا إِنْ نَسِينَا أَوْ أَخْطَأْنَا ۗ رَبَّنَا وَلَا تَحْمِلْ عَلَيْنَا إصْرًا كَمَا حَمَلْتَهُ عَلَى الَّذِينَ مِنْ قَبْلِنَا ۗ رَبَّنَا وَلَا تُحَمِّلْنَا مَا لَا طَاقَةَ لَنَا بِهِ ۗ وَاعْفُ عَنَّا وَارْحَمْنَا ۗ أَنْتَ مَوْلَانَا فَانصُرْنَا عَلَى الْقَوْمِ الْكَافِرِينَ

*"Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya. Ia mendapat pahala (dari kebaikan) yang diusahakannya dan ia mendapat siksa (dari kejahatan) yang dikerjakannya. (Mereka berdoa): 'Ya Tuhan kami, janganlah Engkau hukum kami jika kami lupa atau kami melakukan kesalahan. Ya Tuhan kami, janganlah Engkau bebaskan kepada kami beban yang berat sebagaimana Engkau bebaskan kepada orang-orang sebelum kami. Ya Tuhan kami, janganlah Engkau pikulkan kepada kami apa yang tak sanggup kami memikulnya. Maafkanlah kami; ampunilah kami; dan rahmatilah kami. Engkaulah penolong kami, maka tolonglah kami terhadap kaum yang kafir'." (Q.S Al-Baqarah : 286)*

Tafsir Quraish Shihab mengartikan ayat ini dengan "Allah tidak membebani hamba-hamba-Nya kecuali dengan sesuatu yang dapat dilaksanakan. Maka, setiap orang yang mukallaf, amalnya akan dibalas: yang baik dengan kebaikan, dan yang jelek dengan kejelekan. Tunduklah kamu sekalian, hai orang-orang Mukmin, dengan berdoa, "Ya Tuhan, jangan hukum kami jika kami lupa dalam melaksanakan perintah-Mu, atau bersalah karena beberapa sebab. Janganlah Engkau beratkan syariat untuk kami seperti Engkau memberatkan orang-orang Yahudi oleh sebab

kekerasan dan kelaliman mereka. Dan janganlah Engkau bebankan kepada kami tugas yang tidak mampu kami lakukan. Berilah kami maaf dengan kemuliaan-Mu. Ampunilah kami dengan karunia-Mu. Berikan kami rahmat-Mu yang luas. Engkaulah penolong kami, maka tolonglah kami, ya Tuhan--untuk menegakkan dan menyebarkan agamamu--terhadap kaum yang kafir."” (Quraish Shihab).

Ayat ini mengajarkan pentingnya menempatkan developer pada proyek yang sesuai dengan keahlian mereka untuk memastikan mereka dapat bekerja dengan efektif dan efisien. Dengan demikian, perusahaan tidak hanya meningkatkan produktivitas dan kualitas hasil kerja, tetapi juga mempraktikkan prinsip-prinsip Islami dalam interaksi sehari-hari. Penelitian ini, yang menggunakan algoritma genetika untuk mengoptimalkan penempatan developer, berusaha untuk menciptakan sistem yang adil dan efisien sesuai dengan ajaran Islam, dimana setiap developer diberi tugas yang sesuai dengan kemampuan mereka, sehingga dapat membantu mencapai tujuan bersama dengan cara yang paling efektif dan bermartabat.

## **BAB V**

### **KESIMPULAN DAN SARAN**

Pada bab ini akan membahas tentang kesimpulan dari kegiatan yang dilakukan selama pengerjaan penelitian ini. Saran untuk pengembangan lebih lanjut berdasarkan penelitian ini yang kemudian hari akan diserahkan.

#### **5.1 Kesimpulan**

Hasil uji coba menunjukkan bahwa kombinasi nilai crossover 0,4 dan mutasi 0,2 mampu menghasilkan solusi terbaik dengan recognition rate sebesar 83% dibandingkan kombinasi lainnya. Temuan ini mengindikasikan bahwa algoritma genetika dengan konfigurasi parameter yang tepat dapat menjadi solusi yang baik untuk optimasi penempatan developer pada proyek yang sesuai dengan keahlian mereka.

Nilai crossover 0,4 yang berada di tengah-tengah menunjukkan keseimbangan yang baik antara eksplorasi dan eksploitasi, memungkinkan variasi genetik yang cukup tanpa kehilangan stabilitas. Nilai mutasi 0,2 yang rendah menunjukkan hasil terbaik karena memberikan ruang solusi baru tanpa merusak solusi yang sudah ada. Secara keseluruhan, konfigurasi parameter ini adalah yang paling efektif dalam mengelola sumber daya developer dengan algoritma genetika, memberikan keseimbangan yang tepat antara variasi dan kestabilan. Temuan ini dapat dijadikan panduan untuk penerapan algoritma genetika dalam manajemen sumber daya manusia di PT Sekawan Media Informatika maupun perusahaan lain yang menghadapi tantangan serupa.

## 5.2 Saran

Berdasarkan penelitian yang telah dilakukan, berikut adalah saran-saran untuk penelitian selanjutnya guna meningkatkan penelitian ini:

1. Menggunakan Dataset yang Lebih Beragam:

Dengan memperluas cakupan dataset untuk mencakup berbagai jenis proyek dan skenario pengelolaan sumber daya developer, hasil yang diperoleh akan lebih representatif dan dapat memberikan gambaran yang lebih menyeluruh tentang performa algoritma.

2. Evaluasi yang Lebih Komprehensif:

Selain itu, penggunaan data yang bervariasi juga memungkinkan evaluasi yang lebih komprehensif terhadap keakuratan dan generalisasi algoritma genetika yang digunakan. Hal ini penting untuk memastikan bahwa algoritma tidak hanya efektif dalam kondisi tertentu tetapi juga dapat diandalkan dalam berbagai situasi yang berbeda.

## DAFTAR PUSTAKA

- Amini, Nurlatifah, Triando Hamonangan Saragih, Mohammad Reza Faisal, Andi Farmadi, Friska Abadi, Ilmu Komputer, Matematika Dan Ilmu, et al. n.d. "JIP (Jurnal Informatika Polinema) Implementasi Algoritma Genetika Untuk Seleksi Fitur Pada Klasifikasi Genre Musik Menggunakan Metode Random Forest."
- Arifudin, Riza. n.d. "Optimasi Penjadwalan Proyek Dengan Penyeimbangan Biaya Menggunakan Kombinasi Cpm Dan Algoritma Genetika." *Riza Arifudin Jurnal Masyarakat Informatika*. Vol. 2.
- Fatkur, Muhammad, Lili Andraini, and Teknik Komputer. n.d. "Prediksi Inflasi Indonesia Menggunakan Model Regresi Berdasarkan Algoritma Genetika." *Portaldata.Org*. Vol. 2.
- Hafiz, Muhammad, Dio Rahman Alfateh, Alya Azzahrah Muhjah, Tiara Oktaviani, M Syahdan Gintana, Kevin Tri Anggoro, and Perani Rosyani. n.d. "Algoritma Genetika Penjadwalan Mesin Bertipe Sistem Produksi Untuk Meminimalkan Waktu Proses Produksi." *Jurnal Ilmu Komputer Dan Pendidikan*. Vol. 1. <https://journal.mediapublikasi.id/index.php/logic>.
- Hammad, Rifqi, Ahmad Zuli Amrullah, and Pahrul Irfan. 2022. "Optimasi Neural Network Dengan Menggunakan Algoritma Genetika Untuk Prediksi Jumlah Kunjungan Wisatawan (Neural Network Optimization Using Genetic Algorithm For Prediction Number of Tourist Visits)" 3 (4): 227–35.
- Li, Zhi, and Qi Zhu. 2020. "Genetic Algorithm-Based Optimization of Offloading and Resource Allocation in Mobile-Edge Computing." *Information (Switzerland)* 11 (2). <https://doi.org/10.3390/info11020083>.
- Oktarina, Dwi, and Alyauma Hajjah. 2019. "Perancangan Sistem Penjadwalan Seminar Proposal Dan Sidang Skripsi Dengan Metode Algoritma Genetika." *JOISIE Journal Of Information System And Informatics Engineering* 3 (1): 32–40.
- Pane, Syafrial Fachri, Rolly Maulana Awangga, Esi Vidia Rahmadani, and Seta Permana. 2019. "Implementasi Algoritma Genetika Untuk Optimalisasi Pelayanan Kependudukan." *Jurnal Tekno Insentif* 13 (2): 36–43. <https://doi.org/10.36787/jti.v13i2.130>.
- Ramadhan, Gatra Cahya, Pratama Bagus W, and Yesy Diah Rosita. 2023. "Penentuan Rute Optimal Untuk Jasa Pengiriman Barang Menggunakan Algoritma Genetika." *JTIM: Jurnal Teknologi Informasi Dan Multimedia* 5 (1): 48–55. <https://doi.org/10.35746/jtim.v5i1.322>.
- Septiani, Aida, Siti Julaeha, Jalan Raya Tengah No, Kelurahan Gedong, Pasar Rebo, and Jakarta Timur. 2023. "Penerapan Algoritma Genetika Untuk Memprediksi Penjadwalan Jasa Pasang Pada PT. Reka Graha Semesta." *Jurnal Sains Student Research* 1 (1): 628–39. <https://doi.org/10.61722/jssr.v1i1.200>.
- Setiawan, Debi, Diki Arisandi, and Liza Trisnawati. 2022. "Aplikasi Prediksi Penyakit Sirosis Hati Menggunakan Algoritma Genetika." Vol. 2.

- Sitanggang, Lusya Octoria, and Nurdin Bahtiar. 2019. "Aplikasi Data Mining Untuk Mendeteksi Penyakit Hepatitis Menggunakan Metode Support Vector Machine (SVM) Polynomial (Studi Kasus : Data Pasien Hati India)." *Jurnal Masyarakat Informatika* 10 (1): 20–27. <https://doi.org/10.14710/jmasif.10.1.31490>.
- Tarek Hegazy, By. n.d. "Optimization Of Resource Allocation And Leveling Using Genetic Algorithms."
- Teknika, Jurnal, and Anggi Andriyadi. n.d. "Teknika 16 (01): 133-140 Optimasi Algoritma Genetika Dalam Perancangan Sistem Informasi Penjadwalan Seminar Dan Sidang Skripsi Mahasiswa Institut Informatika Dan Bisnis (IIB) Darmajaya."
- Utami, Amalia Rizqi, and Barokatun Hasanah. 2022. "Analisis Performa Sistem High Altitude Platforms Menggunakan Algoritma Genetika Untuk Pengalokasian Subcarrier." *InComTech : Jurnal Telekomunikasi Dan Komputer* 12 (1): 33. <https://doi.org/10.22441/incomtech.v12i1.14211>.
- Xie, Linlin, Yajiao Chen, and Ruidong Chang. 2021. "Scheduling Optimization of Prefabricated Construction Projects by Genetic Algorithm." *Applied Sciences (Switzerland)* 11 (12). <https://doi.org/10.3390/app11125531>.
- tafsirweb.com. 7 Juni 2024. Surah Al-Qashash Ayat 26 (28:26 Quran) With Tafsir. Diakses pada 7 Juni 2024, dari <https://tafsirweb.com/7076-surat-al-qashash-ayat-26.html>.
- tafsirweb.com. 12 Juni 2024. Surah Al-Jatsiyah Ayat 13 (45:13 Quran) With Tafsir. Diakses pada 12 Juni 2024, dari <https://tafsirweb.com/9505-surat-al-jatsiyah-ayat-13.html>.
- Quraish Shihab. 2001. "Tafsir Al-Mishbah", Diakses pada 18 Juni 2024, dari <https://tafsirq.com/16-an-nahl/ayat-90>

## LAMPIRAN

### A. Data Developer

No	Nama	Jabatan	Tim	Spesialis
1	adi nugraha	Senior Developer	Office	Crossplafom
2	wildan setyo budi	Developer	Office	Mobile
3	melia diana	Developer	Bissiness	Frontend
4	imamah adilah	Proyek Admin	Office	-
5	dinda asri ilmiasih	Proyek Admin	Bissiness	-
6	yudanti dwi setya	Proyek Admin	Office	-
7	eko mursidi	Senior Developer	Office	Crossplatform
8	muh thoha	Senior Developer	Bissiness	Crossplatform
9	moh. Zainul adib	Senior Developer	Custom	Crossplatform
10	maulana mahateri	Developer	Office	Web
11	mohammad wahyu arifin	Senior Developer	Office	Web
12	mukhamad rizky alamsyah	Developer	Bissiness	Mobile
13	tommy nur chomsah	Developer	Bissiness	Backend
14	rosa kurnuasari	Developer	Custom	Backend
15	much. In'am awaludin	Developer	Custom	Web
16	achmad uwais alqorny	Developer	Bissiness	Web
17	yayak zabara	Developer	Office	Crossplatform
18	annisa angraini kusuma	Developer	Office	Mobile
19	masfuul aji eka putra	Developer	Office	Web
20	rizky sena yudha	Developer	Office	Web
21	ta'jilah iibnu fridyatma	Developer	Bissiness	Web
22	edwin yordan	Developer	Custom	Backend
23	machmud efendi	Developer	Office	Web
24	bima renaldi	Developer	Custom	Web
25	dimas fajar ramadan	Developer	Custom	Web

**B. Data Proyek**

<b>No</b>	<b>Nama</b>	<b>Kompleksitas</b>	<b>Tipe</b>	<b>Platform</b>
1	apm	Sedang	Bussiness	Web
2	his-vms	Sedang	Custom	Web
3	apw-erp	Sedang	Bussiness	Web
4	wvi	Sedang	Office	Web
5	itc-vms	Rendah	Custom	Web
6	mld-st	Tinggi	Office	Web
7	mld-hkp	Tinggi	Sipas	Mobile
8	yd-lms	Sedang	Custom	Web
9	jiép-simp	Sedang	Office	Web
10	unmer-sch	Sedang	Office	Web
11	syn-dm	Tinggi	Production	Mobile
12	jsa-mtb	Sedang	Office	Web
13	kna-ksp	Sedang	Bussiness	Web
14	ipi-qa	Sedang	Office	Mobile
15	wm-am	Sedang	Production	Web
16	ttn-ukt	Sedang	Office	Web
17	soho	Tinggi	Production	Mobile
18	ina-ncr	Tinggi	Production	Web
19	mld-simr	Sedang	Sipas	Mobile
20	kom-bjr	Sedang	Office	Web
21	tcb-pos	Sedang	Bussiness	Web
22	twc-trp	Sedang	Production	Web
23	riz-mrpj	Sedang	Bussiness	Web
24	sp-acm	Sedang	Bussiness	Web
25	bpdb-mppos	Sedang	Custom	Web
26	jtmr-iams	Sedang	Custom	Web
27	mind-ai	Tinggi	Production	Web
28	ksa-ak	Sedang	Office	Web
29	uptp3-pkc	Sedang	Office	Web
30	hris	Sedang	Office	Web