

**IMPLEMENTASI ALGORITMA GENETIKA UNTUK OPTIMASI
PENGUNAAN LAHAN PERTANIAN**

SKRIPSI

Laboratorium Komputasi Cerdas dan Visualisasi

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



**Disusun oleh
HARIM ADI SAPUTRO
NIM. 115060813111003**

**PROGRAM STUDI INFORMATIKA/ ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA**

MALANG

2015

LEMBAR PERSETUJUAN

IMPLEMENTASI ALGORITMA GENETIKA UNTUK OPTIMASI PENGGUNAAN LAHAN PERTANIAN

SKRIPSI

LABORATORIUM KOMPUTASI CERDAS DAN VISUALISASI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun oleh:

HARIM ADI SAPUTRO

NIM. 115060813111003

Skripsi ini telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Wayan Firdaus Mahmudy, S.Si., MT, Ph.D
NIP. 197209191997021001

Candra Dewi, S.Kom., MSc.
NIP. 19771114 200312 2 001

LEMBAR PENGESAHAN

IMPLEMENTASI ALGORITMA GENETIKA UNTUK OPTIMASI PENGUNAAN LAHAN PERTANIAN

SKRIPSI

LABORATORIUM KOMPUTASI CERDAS DAN VISUALISASI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun Oleh :

HARIM ADI SAPUTRO

NIM. 115060813111003

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 04 Mei 2015

Penguji I

Ir. Sutrisno, M.T.

NIP. 19570325 198701 1 001

Penguji II

Arief Andy Soebroto, ST., M.Kom.

NIP. 19720425 199903 1 002

Penguji III

Suprpto, ST., M.T

NIP. 19710727 199603 1 001

Mengetahui,

Ketua Program Studi Informatika / Ilmu Komputer

Drs. Marji, M.T.

NIP. 19670801 199203 1 001

ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 10 Mei 2015

Mahasiwa

Harim Adi Saputro

NIM.115060813111003

KATA PENGANTAR

Dengan nama Allah SWT Yang Maha Pengasih dan Penyayang. Segala puji bagi Allah SWT karena atas rahmat dan hidayahNya-lah penulis dapat menyelesaikan Tugas Akhir yang berjudul “Implementasi Algoritma Genetika untuk Optimasi Penggunaan Lahan Pertanian”. Shalawat serta salam atas junjungan besar kita Nabi Muhammad S.A.W. beserta keluarga dan para sahabat sekalian. Tugas Akhir ini disusun untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer di Fakultas Ilmu Komputer Universitas Brawijaya Malang. Melalui kesempatan ini, penulis menyampaikan rasa hormat dan terima kasih penulis yang sebesar-besarnya kepada pihak yang telah memberikan bantuan lahir maupun batin selama penulisan tugas akhir ini. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih penulis kepada :

1. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D. dan Ibu Candra Dewi, S.Kom., MSc. selaku dosen pembimbing tugas akhir penulis
2. Kepada orang tua penulis dan seluruh keluarga yang tiada henti hentinya memberikan do'a demi terselesainya tugas akhir ini.
3. Bapak Ir. Sutrisno, M.T, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Himawat Aryadita, S.T, M.Sc, dan Bapak Eddy Santoso, S.Kom selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Fakultas Ilmu Komputer Universitas Brawijaya
4. Bapak Drs. Marji, MT dan Bapak Issa Arwani, S.Kom., M.Sc selaku Ketua dan Sekretaris Program Studi Informatika.
5. Bapak / Ibu Dosen, Staff Administrasi dan Perpustakaan Fakultas Ilmu Komputer Universitas Brawijaya.
6. Seluruh teman-teman mahasiswa Fakultas Ilmu Komputer Universitas brawijaya khususnya angkatan 2011.
7. Seluruh pihak yang telah membantu kelancaran penulisan tugas akhir yang tidak dapat penulisan sebutkan satu per satu.

Semoga penulisan skripsi ini bermanfaat bagi pembaca dan untuk pengembangan penelitian selanjutnya. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna serta banyak kekurangan yang disebabkan oleh keterbatasan

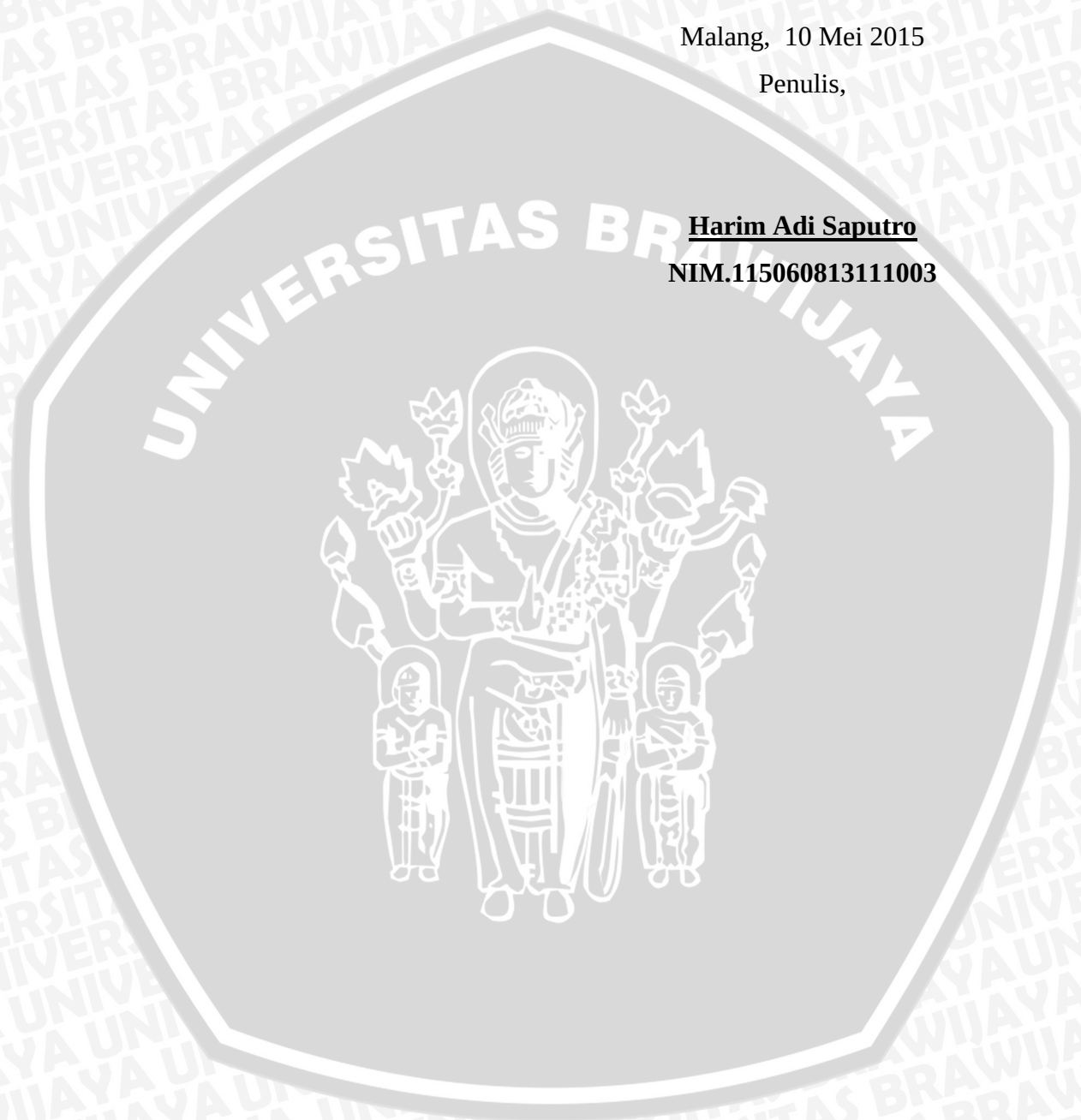
kemampuan dan pengalaman yang dimiliki oleh penulis. Maka dengan kerendahan hati penulis mengharapkan kritik dan saran yang bersifat membangun dari pembaca.

Malang, 10 Mei 2015

Penulis,

Harim Adi Saputro

NIM.115060813111003



ABSTRAK

Harim Adi Saputro, 2015. Implementasi Algoritma Genetika untuk Optimasi Penggunaan Lahan Pertanian.

Dosen Pembimbing : Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D dan Candra Dewi, S.Kom., MSc.

Indonesia memiliki dua musim, yaitu musim penghujan dan musim kemarau. Karena pengaruh dua musim yang dimiliki wilayah Indonesia, petani umumnya mengolah lahan pertaniannya menjadi dua musim tanam, yaitu musim tanam padi saat penghujan dan musim tanam polikultur (menanam lebih dari satu tanaman) saat kemarau. Sistem polikultur ini bertujuan untuk meminimalisasi gagal panen dan mengurangi resiko kerugian apabila salah satu komoditi harganya menurun. Pada saat menerapkan sistem pertanian polikultur, petani sering dihadapkan pada permasalahan mendasar, yaitu bagaimana menentukan proporsi dalam pemanfaatan lahan pertanian sehingga menghasilkan keuntungan yang maksimal dengan mengalokasikan luas lahan dan modal yang dimiliki.

Algoritma genetika merupakan salah satu algoritma yang bersifat *heuristic* dan dapat menyelesaikan permasalahan multi *objectiv*, sehingga dapat diterapkan untuk mencari solusi optimal dari permasalahan optimasi penggunaan lahan pertanian ini. Pencarian solusi dilakukan dengan mengkombinasikan kromosom kemudian diproses dengan operator genetika (seleksi, *crossover*, dan mutasi) dengan menginisialisasi parameter genetika (ukuran populasi, *crossover rate*, *mutation rate*, dan jumlah generasi). Pada penelitian ini, menggunakan representasi kromosom *real coded* dengan reproduksi *extended intermediate crossover* dan *random mutation*. Dari hasil pengujian, didapatkan hasil terbaik dengan nilai *fitness* tertinggi yang mendekati solusi optimal pada ukuran populasi 125, jumlah generasi 400, nilai probabilitas *crossover* 0,4 dan probabilitas mutasi 0,6. Seleksi yang lebih baik adalah seleksi elitism.

Kata Kunci : Algoritma Genetika, Optimasi, Penggunaan Lahan Pertanian

ABSTRACT

Harim Adi Saputro, 2015. Implementation of Genetic Algorithm for Optimization of Agricultural Land Use.

Advisor : Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D dan Candra Dewi, S.Kom., MSc.

Indonesia have two season which are rainy and dry season. Because of Indonesia's season, farmers generally cultivate their farm into two planting season which are rice planting season during rainy season and polyculture (planting more than one kind of crop) during dry season. The aims of polyculture system is to minimize crop failure and reduce the risk of loss if one commodity prices dropped. At the time of applying polyculture system, farmers are often have fundamental problem which is how to determine the proportion of the agricultural land to get maximum profit by allocating limited land and capital.

Genetic algorithm is one of the heuristic algorithm and can solve multi-objective problem. Genetic algorithm can be applied to find optimum solution of the optimization problem of agricultural land use. Search for solution will done by combining the chromosomes then processed by the genetic operators (crossover, mutation and selection) to initialize the genetic parameters (population size, crossover rate, mutation rate and the number of generations). In this research, used real coded chromosomes representation extended intermediate crossover and random mutation to reproduction. This research result are highest fitness value approaching the optimal solution in the population size of 125, 400 generations, crossover rate value 0.4 and mutation rate 0.6. The best selection is elitism selection.

Keywords : Genetic Algorithm, Optimation, Agricultural Land Use

DAFTAR ISI

LEMBAR KEMAJUAN SKRIPSI	Error! Bookmark not defined.
LEMBAR PERSETUJUAN.....	i
LEMBAR PENGESAHAN	ii
ORISINALITAS SKRIPSI	iii
KATA PENGANTAR	iv
ABSTRAK	vi
ABSTRACT.....	vii
DAFTAR ISI.....	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
DAFTAR PERSAMAAN	xiii
DAFTAR SOURCE CODE.....	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan	4
1.5 Manfaat	4
1.6 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	6
2.1 Kajian Pustaka.....	6
2.2 Tanaman Palawija	9
2.3 Tanaman Hortikultura	10

2.4 Pertanian Polikultur.....	11
2.4.1 Jenis-jenis Polikultur.....	12
2.5 Optimasi	14
2.6 Algoritma Genetika.....	14
2.6.1 Struktur Algoritma Genetika.....	16
2.6.2 Parameter Algoritma Genetika.....	17
2.6.3 Proses Algoritma Genetika	18
2.7 Penerapan Algoritma Genetika	19
2.7.1 Generate populasi awal	19
2.7.2 Representasi Kromosom	19
2.7.3 Crossover	20
2.7.4 Mutasi.....	21
2.7.5 <i>Fitness</i>	22
2.7.6 Seleksi	22
BAB III METODOLOGI PENELITIAN.....	27
3.1 Tahapan Penelitian.....	27
3.1.1 Studi Literatur	28
3.1.2 Analisis Kebutuhan.....	28
3.1.3 Perancangan	28
3.1.4 Implementasi	29
3.1.5 Pengujian.....	29
3.1.6 Pengambilan Kesimpulan.....	29
3.2 Kebutuhan Sistem	29
3.2.1 Kebutuhan Perangkat Keras.....	29
3.2.2 Kebutuhan Perangkat Lunak.....	30
3.3 Formulasi Permasalahan	30

3.3.1 Deskripsi Umum Sistem	30
3.3.2 Data yang Digunakan	30
3.3.3 Fungsi Obyektif Permasalahan	32
3.4 Siklus Penyelesaian Masalah Menggunakan Algoritma Genetika.....	34
3.3.1 Representasi Kromosom dan Perhitungan Fitness.....	36
3.3.2 Inisialisasi Populasi Awal	39
3.3.3 Reproduksi	40
3.3.4 Evaluasi.....	44
3.3.5 Seleksi	44
3.3.6 Memilih Kromosom Terbaik	48
BAB IV PERANCANGAN	49
4.1 Perancangan <i>Database</i>	49
4.1.1 Struktur Tabel.....	49
4.2 Perancangan <i>User Interface</i>	49
4.2.1 Rancangan Halaman Utama.....	49
4.2.2 Rancangan Halaman Setup Data.....	50
4.2.2 Rancangan Halaman Hitung Optimasi.....	51
4.2.3 Rancangan Halaman Hasil Proses Algoritma Genetika.....	52
4.3 Perancangan Uji Coba.....	52
4.3.1 Uji Coba Banyaknya Generasi	53
4.3.2 Uji Coba Ukuran Populasi	53
4.3.2 Uji Coba Kombinasi <i>Crossover Rate</i> dan <i>Mutasi Rate</i>	54
4.3.2 Uji Coba Seleksi.....	55
BAB V IMPLEMENTASI.....	55
5.1 Implementasi Program	55
5.1.1 Pembangkitan Populasi Awal	55

5.1.2 Perhitungan Nilai <i>Fitness</i>	56
5.1.3 Proses <i>Crossover</i>	57
5.1.4 Proses Mutasi	58
5.1.5 Proses Seleksi <i>Elitism</i>	60
5.1.6 Proses Seleksi <i>Binary Tournament</i>	60
5.1.7 Pemilihan Kromosom Terbaik	61
5.2 Implementasi Antarmuka Pengguna	62
5.2.1 Implementasi Halaman Utama	62
5.2.2 Implementasi Halaman <i>Entry Data Tanaman</i>	63
5.2.3 Halaman Proses Algoritma Genetika	63
5.2.4 Implementasi Halaman Daftar Data Tanaman	64
BAB VI PENGUJIAN DAN ANALISA	66
6.1 Hasil Pengujian Sistem	66
6.2 Hasil dan Analisa Pengujian Parameter Algoritma Genetika	67
6.2.1 Hasil dan Analisa Uji Coba Banyaknya Generasi	67
6.2.2 Hasil Dan Uji Coba Ukuran Populasi	70
6.2.3 Hasil dan Analisa Uji Coba Kombinasi <i>Crossover Rate</i> dan <i>Mutation Rate</i>	73
6.2.4 Hasil dan Analisa Perbandingan Seleksi Metode <i>Elitism</i> dan <i>Binary Tournament</i>	75
6.3 Analisa Hasil Pengujian Parameter Optimal Algoritma Genetika	77
BAB VII PENUTUP	79
7.1 Kesimpulan	79
7.2 Saran	79
DAFTAR PUSTAKA	81

DAFTAR GAMBAR

Gambar 2.1 Mencari Solusi Dengan Algoritma Genetika	16
Gambar 2.2 Siklus Umum Algoritma Genetika	19
Gambar 2.3 Contoh <i>Crossover</i>	21
Gambar 2.4 Contoh Proses Mutasi	22
Gambar 2.5 Pseudocode Seleksi Replacement	24
Gambar 2.6 Pseudocode Seleksi Elitism	25
Gambar 2.7 Pseudocode Seleksi Binary Tournament	26
Gambar 3.1 Diagram Alir Tahapan Penelitian	27
Gambar 3.2 Diagram Algoritma Genetika	35
Gambar 3.3 Contoh Representasi Kromosom	37
Gambar 3.4 Diagram Alir Proses Extended Intermediate Crossover	40
Gambar 3.5 Hasil Perhitungan Proses Crossover	41
Gambar 3.6 Diagram Alir Proses <i>Random Mutation</i>	42
Gambar 3.7 Hasil Perhitungan Proses Mutasi	43
Gambar 3.8 Seleksi <i>Elitism</i>	45
Gambar 3.9 Diagram Alir Seleksi Binary tournament	47
Gambar 4.1 Rancangan Tampilan Halaman Utama	50
Gambar 4.2 Rancangan Tampilan Halaman Setup Data	50
Gambar 4.3 Tampilan Halaman Input Data Optimasi	51
Gambar 4.4 Tampilan Halaman Hasil Algoritma Genetika	52
Gambar 5.1 Implementasi Halaman Utama	63
Gambar 5.2 Implementasi Halaman <i>Entry</i> Data Tanaman	63
Gambar 5.3 Implementasi Halaman Proses Algoritma Genetika	64
Gambar 5.4 Implementasi Halaman Daftar Data Tanaman	65
Gambar 5.5 Implementasi Submenu <i>Edit</i> Data Tanaman	65
Gambar 6.1 Grafik Hasil Uji Coba Banyaknya Generasi	69
Gambar 6.2 Grafik Hasil Uji Coba Banyaknya Populasi	72
Gambar 6.3 Hasil Uji Coba Kombinasi <i>Crossover rate</i> dan <i>Mutation rate</i>	75
Gambar 6.4 Perbedaan Nilai Fitness Uji Coba Seleksi Elitism dan Binary Tournament	76

DAFTAR TABEL

Tabel 2.1 Kajian Pustaka.....	8
Tabel 2.2 Representasi Kromosom	20
Tabel 3.1 Analisis kebutuhan data penelitian	31
Tabel 3.2 Data Tanaman	32
Tabel 3.3 Data Tanaman	36
Tabel 3.4 Contoh Populasi Awal	39
Tabel 3.5 Nilai Fitness dari Populasi Awal.....	39
Tabel 3.6 Hasil <i>Repair Offspring Crossover</i>	41
Tabel 3.7 Hasil <i>Repair Offspring</i> Mutasi.....	43
Tabel 3.8 Hasil Evaluasi Individu.....	44
Tabel 3.9 Hasil Seleksi <i>Elitism</i>	46
Tabel 3.10 Hasil Seleksi <i>Binary Tournament</i>	47
Tabel 3.11 Individu Terbaik.....	48
Tabel 3.12 Detail Kromosom Terbaik	48
Tabel 4.1 Stuktur Tabel Tanaman.....	49
Tabel 4.2 Rancangan Uji Coba Banyaknya Generasi	53
Tabel 4.3 Rancangan Uji Coba Ukuran Populasi	54
Tabel 4.4 Rancangan Uji Coba Kombinasi cr dan mr	54
Tabel 4.5 Uji Coba Perbandingan Seleksi.....	55
Tabel 6.1 Hasil Pengujian Sistem	66
Tabel 6.2 Hasil Percobaan Banyaknya Generasi	68
Tabel 6.3 Hasil Uji Coba Populasi.....	71
Tabel 6.4 Hasil Uji Coba Kombinasi <i>Crossover rate</i> dan <i>Mutation rate</i>	74
Tabel 6.5 Hasil Uji Coba Metode Seleksi.....	76
Tabel 6.6 Rekomendasi Proporsi Luas Optimal	77
Tabel 6.7 Rekomendasi Proporsi Oleh Pakar	77

DAFTAR PERSAMAAN

Persamaan (2-1) Pembangkitan <i>Offspring Extended Intermediate Exchange Crossover</i>	19
Persamaan (2-2) Pembangkitan <i>Offspring Extended Intermediate Exchange Crossover</i>	19
Persamaan (2-3) Pembangkitan <i>Offspring Random Mutation</i>	20
Persamaan (2-4) Persamaan Total Nilai Fitness	23
Persamaan (2-5) Persamaan Probabilitas Nilai Fitness.....	23
Persamaan (2-6) Persamaan Probabilitas Kumulatif Nilai Fitness	23
Persamaan (3-1) Fungsi Obyektif	33
Persamaan (3-2) Persamaan Penalti	33
Persamaan (3-3) Penalti1	34
Persamaan (3-4) Total Biaya Produksi	34
Persamaan (3-5) Penalti2 Setiap Tanaman	34
Persamaan (3-6) Total Penalti2.....	34
Persamaan (3-7) Repair Offspring	37

DAFTAR SOURCE CODE

Source Code 5.1 Pembangkitan Populasi Awal	55
Source Code 5.2 Perhitungan Nilai Fitness	57
Source Code 5.3 Proses Crossover	57
Source Code 5.4 Proses Repair Offspring crossover	58
Source Code 5.5 Proses Mutasi.....	59
Source Code 5.6 Proses Repair Offspring Mutasi	60
Source Code 5.7 Proses Seleksi Elitism.....	60
Source Code 5.8 Proses Seleksi Binary Tournament.....	61
Source Code 5.9 Proses Pemilihan Kromosom Terbaik	62



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan negara agraris yang sebagian besar penduduknya bermata pencaharian di bidang pertanian. Negara Indonesia sangat diuntungkan dengan dikaruniai kondisi alam yang mendukung, hamparan lahan yang luas, keragaman hayati yang melimpah, serta beriklim tropis, dimana sinar matahari terjadi sepanjang tahun sehingga bisa menanam sepanjang tahun. Dengan adanya sumber daya alam seperti ini, seharusnya mampu membangkitkan Indonesia menjadi negara yang makmur dan tercukupi kebutuhan pangan seluruh warganya. Meskipun belum terpenuhi, pertanian menjadi salah satu sektor riil yang memiliki peran sangat nyata dalam membantu penghasilan devisa negara (Warsani, 2013).

Berdasarkan data BPS (Badan Pusat Statistik) tahun 2014, jumlah petani di Indonesia mencapai 44% dari total angkatan kerja di Indonesia, atau sekitar 46,7 juta jiwa. Sebagai negara agraris, hingga kini mayoritas penduduk Indonesia telah memanfaatkan sumber daya alam untuk menunjang kebutuhan hidupnya dan salah satunya ialah dengan menggantungkan hidup pada sektor pertanian. Oleh karena itu sektor pertanian memiliki peranan yang sangat penting sebagai penghasil pangan bagi penduduk yang jumlahnya tiap tahun semakin bertambah. Berdasarkan data Dinas Pertanian Pangan pada tahun 2008, penggunaan optimalisasi lahan pertanian masih rendah. Hal ini diakibatkan oleh kurang kreatifnya petani untuk melakukan usaha tani karena masih melakukan *monocrop* di lahan sawah, sedangkan untuk komoditi lainnya seperti palawija masih dipandang sebagai tanaman *secondary crop*, kecuali untuk tanaman hortikultura yang memiliki nilai ekonomi tinggi.

Letak geografis Indonesia yang berada di wilayah tropis secara tidak langsung mengakibatkan sebagian besar wilayah Indonesia memiliki dua musim yaitu musim penghujan dan musim kemarau. Karena pengaruh dua musim yang dimiliki wilayah Indonesia tersebut, petani umumnya mengolah lahan sawahnya menjadi dua musim tanam, yaitu musim tanam padi ketika musim hujan dan musim tanam palawija saat musim kemarau (Sutanta, 2005). Pada saat musim kemarau

dengan curah hujan dan ketersediaan pengairan yang sedikit, tetapi mengharuskan petani untuk tetap menanam komoditas pada lahan pertaniannya sebagai mata pencaharian utama. Umumnya, pada daerah pertanian yang curah hujan tidak merata sepanjang tahun dan irigasi teknis tidak tersedia, pola yang digunakan adalah pola tanam polikultur, dimana petani menanam lebih dari satu tanaman pada lahan yang sama untuk mendapatkan hasil yang beragam dari komoditas yang ditanam (Anindita, 2013). Sistem pertanian polikultur ini mempunyai kecenderungan bahwa yang paling penting adalah tetap memperoleh hasil panen dan keuntungan secara ekonomi. Pada saat menanam secara polikultur ini, petani sering dihadapkan pada permasalahan yang mendasar, yaitu bagaimana menentukan pembagian proporsi luas pada setiap jenis tanaman yang akan ditanam, dengan memanfaatkan luas lahan dan modal yang dimiliki sehingga menghasilkan keuntungan yang maksimal (Sutanta, 2005).

Dari permasalahan yang telah diuraikan di atas, penulis berinisiatif untuk membuat sebuah sistem yang dapat merekomendasikan proporsi luas lahan yang tepat untuk setiap jenis tanaman yang akan ditanam, dengan mengalokasikan luas lahan dan modal yang dimiliki. Sehingga pemanfaatan luas lahan dan prediksi keuntungan panen bisa optimal. Sistem ini akan dibuat dengan mengimplementasikan algoritma genetika untuk mendapatkan luas lahan yang optimal untuk masing-masing jenis tanaman, agar menghasilkan keuntungan maksimal dengan mengalokasikan luas lahan dan modal yang dimiliki.

Metode untuk optimasi penggunaan luas lahan pertanian ini menggunakan algoritma genetika karena hasil akurasi terbaik pada metode algoritma genetika dibandingkan metode lain. Algoritma ini sering digunakan dalam permasalahan optimasi yang kompleks karena selain memungkinkan untuk keluar dari optimum lokal, algoritma ini juga memiliki ruang pencarian yang sangat luas. Algoritma genetika banyak diterapkan dalam permasalahan bidang fisika, biologi, ekonomi, sosiologi, maupun bidang ilmu lainnya (Mahmudy, 2013).

Dalam penelitian sebelumnya yang dilakukan Sari, Mahmudy dan Dewi (2014) dengan menerapkan Algoritma Genetika untuk optimasi pemenuhan kebutuhan gizi pada ibu hamil. Dari hasil pengujian yang telah dilakukan, algoritma genetika mampu memberikan solusi optimal yang mendekati kebutuhan gizi ibu

hamil dengan harga yang minimal. Penelitian selanjutnya dilakukan oleh Wardhani, Safrizal dan Chairi (2011) yang menerapkan implementasi algoritma genetika untuk optimasi komposisi bahan pakan ikan air tawar terbukti keberhasilan perhitungan pemenuhan kebutuhan nutrisi 100% dan tingkat efisiensi biaya pakan sekitar 46.5%. Penelitian dengan kasus yang berbeda dilakukan oleh Fiarni dan Bastiyan (2013) dengan menerapkan algoritma genetika untuk merekomendasikan portofolio investasi yang optimal, dengan hasil tingkat kesesuaian 67%. Ketiga penelitian tersebut menyimpulkan bahwa algoritma genetika dapat digunakan dalam penyelesaian masalah optimasi. Pendekatan metode yang sama akan dilakukan pada sistem untuk rekomendasi kombinasi jenis tanaman dan proporsi luas lahan yang tepat. Optimalisasi dibatasi dengan kendala berupa luas lahan dan modal yang dimiliki.

Dengan adanya sistem ini, diharapkan dapat membantu para penyuluh dan petani dalam menentukan proporsi luas lahan yang tepat pada setiap jenis tanaman yang akan ditanam. Dengan demikian, akan didapat pemanfaatan luas lahan yang optimal, untuk mengalokasikan luas lahan dan modal yang dimiliki agar diperoleh keuntungan maksimum.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan, maka bisa dirumuskan masalah sebagai berikut:

1. Bagaimana menerapkan metode algoritma genetika untuk menyelesaikan masalah optimasi penggunaan lahan pertanian?
2. Bagaiman menentukan representasi kromosom yang paling efisien untuk permasalahan optimasi penggunaan lahan pertanian?
3. Bagaimana parameter algoritma genetika yang paling optimal untuk menyelesaikan masalah optimasi penggunaan lahan pertanian?

1.3 Batasan Masalah

Batasan masalah yang akan dijadikan sebagai pedoman dalam pelaksanaan penelitian ini adalah sebagai berikut :

1. Jenis tanaman hanya terbatas pada tanaman palawija dan hortikultura kelompok sayuran.

2. Parameter masukan berupa luas lahan, modal dan jenis tanaman yang dipilih untuk diproses menggunakan algoritma genetika sehingga dihasilkan keluaran berupa komposisi jenis tanaman beserta proporsi luas optimumnya dan prediksi keuntungan.
3. Data yang dijadikan perhitungan dalam optimasi adalah data kuantitatif meliputi biaya produksi, luas lahan minimal dan hasil produksi pada masing-masing tanaman.
4. Pola tanam polikultur yang digunakan dalam penelitian ini adalah jenis polikultur campuran berkelompok, yaitu penanaman beberapa jenis tanaman pada satu lahan dengan segmentasi secara teratur.
5. Biaya produksi dan hasil produksi dianggap sama untuk semua karakteristik dan jenis tanah.
6. Semua jenis komoditi yang ditanam diasumsikan berhasil panen semua.

1.4 Tujuan

1. Mengimplementasikan algoritma genetika untuk merekomendasikan proporsi luas lahan yang optimal untuk penggunaan lahan pertanian.
2. Mengetahui representasi kromosom yang efisien untuk permasalahan optimasi penggunaan lahan pertanian?
3. Menganalisa dan menguji penerapan algoritma genetika untuk optimasi penggunaan lahan pertanian.

1.5 Manfaat

Penelitian ini diharapkan mempunyai manfaat sebagai berikut :

1. Bagi penyuluh, hasil penelitian ini dapat digunakan sebagai media untuk memberikan rekomendasi bagi petani dalam penentuan proporsi luas lahan yang tepat untuk setiap jenis tanaman yang akan ditanam, sehingga mengoptimalkan kinerja tenaga penyuluh karena menyediakan keluaran perhitungan dan rekomendasi yang siap pakai.
2. Bagi petani, hasil penelitian ini dapat digunakan untuk membantu menentukan proporsi luas lahan yang tepat untuk masing-masing tanaman, untuk menghasilkan keuntungan yang maksimal dengan mengalokasikan lahan dan modal yang dimiliki.

1.6 Sistematika Penulisan

Sistematika penulisan penelitian ditunjukkan untuk memberikan gambaran dan uraian dari penyusun tugas akhir secara garis besar yang meliputi beberapa bab sebagai berikut :

BAB I : PENDAHULUAN

Menguraikan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Menguraikan tentang tinjauan pustaka, dasar teori dan referensi yang mendasari pembuatan sistem Optimasi Penggunaan Lahan Pertanian Menggunakan Algoritma Genetika.

BAB III : METODE PENELITIAN

Menguraikan tentang metode dan langkah kerja yang dilakukan dalam penelitian yang terdiri dari studi literatur, metode pengambilan data, analisis kebutuhan. Dalam bab ini juga dijelaskan proses algoritma genetika beserta contoh perhitungan manualnya

BAB IV : PERANCANGAN

Membahas perancangan yang terdiri dari perancangan database, perancangan *user interface* dan perancangan pengujian

BAB V : IMPLEMENTASI

Membahas implementasi dari sistem Optimasi Penggunaan Lahan Pertanian Menggunakan Algoritma Genetika yang sesuai dengan perancangan sistem yang telah dibuat

BAB VI : PENGUJIAN DAN ANALISIS

Memuat hasil pengujian dan analisis terhadap sistem yang telah direalisasikan

BAB VII : PENUTUP

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam tugas akhir ini serta saran-saran untuk pengembangan lebih lanjut.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

Pada bab ini akan diuraikan tinjauan pustaka dan dasar teori yang mendukung penelitian yang diperoleh dari beberapa referensi diantaranya kajian pustaka, teori tentang tanaman palawija, tanaman hortikultura, polikultur, optimasi dan algoritma genetika.

2.1 Kajian Pustaka

Penelitian tentang rekomendasi dan optimasi telah banyak dilakukan dengan berbagai obyek permasalahan yang berbeda. Pada penelitian ini, digunakan beberapa referensi yang relevan sebagai rujukan dalam proses penenelitian nantinya.

Salah satu referensi yang menjadi rujukan pada penelitian ini adalah penelitian yang dilakukan oleh Sari, Mahmudy dan Dewi (2014) dengan menerapkan algoritma genetika untuk optimasi kebutuhan gizi pada ibu hamil. Penelitian ini bertujuan untuk mendapatkan pemecahan dalam optimasi pemenuhan asupan gizi pada ibu hamil dengan macam bahan yang memiliki gizi seimbang, sesuai kebutuhan ibu hamil serta memiliki harga seminimal mungkin. Proses algoritma pada penelitian ini diawali dengan *generate* populasi awal sebanyak jumlah populasi yang ditentukan. Representasi kromosom dilakukan dengan representasi *integer encoding* yang mewakili jenis makanannya. Pada penelitian ini terdapat penalti yang diberlakukan untuk individu yang melanggar aturan dalam hal ini adalah jika total gizi nya kurang dari total kebutuhan gizi yang seharusnya. Proses yang terakhir adalah proses seleksi dengan cara menghitung total harga pada masing-masing individu tersebut dan individu yang terpilih sebagai solusi adalah individu dengan komposisi gizi yang memenuhi dan harga minimal (Sari, Mahmudy & Dewi, 2014). Dari hasil pengujian yang telah dilakukan, algoritma genetika mampu memberikan solusi optimal yang mendekati kebutuhan gizi ibu hamil dengan harga yang minimal.

Penelitian selanjutnya adalah penelitian yang dilakukan oleh Wardhani dengan menerapkan algoritma genetika pada permasalahan optimasi komposisi pakan ikan air tawar. Penelitian tersebut bertujuan untuk membuat suatu sistem komputerisasi menggunakan metode algoritma genetika multi-obyektif untuk menentukan komposisi bahan pakan ikan air tawar dengan mudah dan menghasilkan kombinasi terbaik (Wardhani, Safrizal & Chairi, 2011). Proses algoritma genetika pada penelitian ini yaitu dengan melakukan inisialisasi populasi kromosom dengan solusi bilangan integer secara acak. Kemudian melakukan evaluasi setiap kromosom dalam populasi. Proses reproduksi dilakukan dengan cara *crossover* dan mutasi untuk menghasilkan keturunan baru. Proses terakhir adalah dengan melakukan seleksi individu yang ada dengan metode *roulette wheel*. Fungsi obyektif yang digunakan pada penelitian ini terdiri dari dua fungsi yaitu $f_1(x)$ yang merupakan nilai gizi, dan fungsi $f_2(x)$ yang merupakan fungsi efisiensi biaya. Dari perhitungan fungsi tersebut, akan diperoleh solusi optimum nilai $f_1(x)$ yang menghasilkan nilai selisih sama dengan 0 (Wardhani, Safrizal & Chairi, 2011).

Penelitian selanjutnya adalah penelitian yang dilakukan oleh Fiarni dan Bastiyan dengan menerapkan algoritma genetika untuk solusi permasalahan portofolio investasi. Penelitian ini bertujuan untuk merekomendasikan proporsi portofolio saham sehingga mendapatkan hasil yang optimal. Proses algoritma genetika pada penelitian ini yaitu dengan melakukan inisialisasi populasi kromosom dengan solusi bilangan biner. Kemudian melakukan evaluasi setiap kromosom dalam populasi. Proses reproduksi dilakukan dengan cara *crossover* dan mutasi untuk menghasilkan keturunan baru (Fiarni dan Bastiyan, 2013). Dari hasil pengujian yang telah dilakukan, algoritma genetika mampu memberikan solusi optimal untuk rekomendasi portofolio saham.

Referensi keempat adalah penelitian dengan obyek serupa namun menggunakan metode lain yang dilakukan Sutanta 2005 yang menerapkan metode *linier programming* untuk optimasi penggunaan lahan pertanian. Penelitian ini bertujuan untuk merekomendasikan proporsi penggunaan luas lahan yang tepat untuk setiap jenis tanaman yang akan ditanam dengan batasan modal yang dimiliki. Masukan dalam penelitian ini adalah modal, luas lahan yang dimiliki dan tanaman yang akan ditanam. Sementara keluaran dari penelitian ini adalah jenis tanaman

yang akan ditanam, proporsi luas masing-masing tanaman, total biaya produksi dan keuntungan yang didapatkan.

Secara garis besar, rangkuman dari penelitian di atas ditampilkan pada tabel

2.1.

Tabel 2.1 Kajian Pustaka

No	Judul	Obyek	Metode	Hasil
1.	Optimasi Asupan Gizi Pada Ibu Hamil Dengan Menggunakan Algoritma Genetika	Pemenuhan asupan gizi pada ibu hamil dengan macam bahan yang memiliki gizi seimbang dan sesuai dengan kebutuhan ibu hamil serta memiliki harga seminimal mungkin	Metode yang digunakan yaitu algoritma genetika dengan permutasi integer yang mewakili item makanan. Reproduksi menggunakan <i>one cut point crossover</i> dan <i>exchange mutation</i>	Hasilnya berupa rekomendasi komposisi makanan dengan total kandungan gizi seimbang yang mendekati kebutuhan ibu hamil dengan optimasi harga minimal.
2	Optimasi Komposisi Bahan Pakan Ikan Air Tawar Menggunakan Metode <i>Multi-Objective Genetic Algorithm</i>	Komposisi bahan pakan ikan air tawar yang memenuhi kebutuhan nutrisi dengan biaya yang efisien.	Metode yang digunakan yaitu <i>Multi-Objective Genetic Algorithm</i> dengan representasi kromosom <i>integer encoding</i>	Penerapan metode yang digunakan berhasil untuk optimasi komposisi bahan pakan dengan tingkat keberhasilan pemenuhan nutrisi 100% dan efisiensi biaya 46,5%.
3.	Sistem Rekomendasi Portofolio Investasi Berbasis Algoritma Genetika	Obyek yang diteliti adalah proporsi portofolio setiap saham untuk mendapatkan hasil yang optimal	Metode yang digunakan yaitu algoritma genetika dengan representasi kromosom bilangan biner. Reproduksi menggunakan <i>one cut point crossover</i> dan <i>exchange mutation</i>	Algoritma genetika mampu memberikan rekomendasi proporsi portofolio saham yang efisien dengan <i>return value</i> besar dan resiko yang minimal.
4.	Aplikasi Sistem Pendukung Pengambilan Keputusan	Pembagian proporsi yang tepat untuk setiap jenis tanaman	Metode yang digunakan adalah metode simpleks yang sesuai	Penentuan proporsi luas lahan yang tepat untuk setiap jenis

	Untuk Optimalisasi Penggunaan Lahan Pertanian	yang akan ditanam dengan mengalokasikan lahan dan modal yang dimiliki	dengan <i>linear programming</i> .	tanaman yang akan ditanam dengan mengalokasikan modal dan lahan yang dimiliki.
--	---	---	------------------------------------	--

Sumber : (Sari, Mahmudy & Dewi, 2013), (Wardhani, Safrizal & Chairi, 2011), (Fiarni & Bastiyan, 2011), (Sutanta, 2005).

Tabel 2.1 merupakan tabel kajian pustaka yang berisi penelitian-penelitian sebelumnya terkait algoritma genetika yang terbukti berhasil diterapkan dalam permasalahan optimasi. Sehingga, pada penelitian ini, metode yang digunakan adalah metode algoritma genetika dengan representasi kromosom menggunakan *realcode*. Representasi kromosom tersebut dibuat berdasarkan proporsi luas lahan pada setiap jenis tanaman yang akan ditanam.

2.2 Tanaman Palawija

Menurut Kamus Besar Bahasa Indonesia, palawija adalah tanaman selain padi yang biasanya ditanam pada sawah maupun ladang. Palawija merupakan tanaman pangan yang biasanya ditanam di lahan sawah pada saat musim kemarau. Namun seiring berkembangnya jaman, dengan memanfaatkan lahan pertanian yang bisa dioptimalkan, sekarang tanaman palawija tidak hanya ditanam pada musim kemarau, tetapi pada musim hujan di lahan kering tanaman palawija juga dibudidayakan (Soekartawi dalam Megawati & Sulistyanto, 2014).

Palawija merupakan istilah pada bidang pertanian, yang secara makna adalah tanaman kedua. Masyarakat Indonesia biasa menyebut palawija sebagai tanaman kedua karena palawija ini adalah tanaman penambah selain tanaman padi. Umumnya para petani di pulau Jawa mengutamakan menanam padi pada sebidang lahannya pada saat musim penghujan. Palawija juga merupakan tanaman yang tidak memerlukan pengairan banyak seperti padi, sehingga dapat ditanam pada lahan kering yang ketersediaan air nya terbatas. Di Indonesia ada banyak jenis tanaman yang memerlukan sedikit kebutuhan air dan dapat tahan dengan serangan hama, sehingga dapat ditanam pada saat kondisi kemarau. Beberapa contoh tanaman palawija yang sering ditanam di Indonesia adalah jagung, kedelai, kacang tanah, kacang hijau dan lain-lain.

2.3 Tanaman Hortikultura

Hortikultura berasal dari kata *hortus* yang artinya kebun dan kata *colere* yang artinya budidaya. Secara harfiah istilah hortikultura adalah usaha budidaya tanaman kebun seperti buah-buahan, sayur-sayuran dan tanaman hias (Zulkarnain, 2010). Hortikultura merupakan komoditas tanaman yang memiliki keunggulan komparatif dan kompetitif yang dimilikinya dalam pemulihan perekonomian di Indonesia mendatang. Perkembangan tanaman hortikultura di Indonesia pada saat ini masih dalam kategori perkebunan rakyat yang tumbuh dan dipelihara secara tradisional, serta komoditas hortikultura yang diusahakan juga masih terbatas (Sunu dan Wartoyo, 2006).

Tanaman yang termasuk kedalam tanaman hortikultura sangat luas dan beragam. Namun secara umum, tanaman hortikultura mempunyai ciri-ciri sebagai berikut (Zulkarnain, 2010) :

1. Produk tanaman hortikultura mudah rusak bila disimpan tanpa perlakuan khusus, misalnya dengan perlakuan suhu rendah yaitu (4°C) atau dengan pelapisan lilin, karena dipanen dalam bentuk segar.
2. Komponen utama mutu produk ditentukan oleh banyaknya kandungan air (*water content*), bukan oleh kandungan bahan kering (*dry content*) karena konsumsinya dalam keadaan segar.
3. Ketersediaan produk, terutama kelompok buah-buahan bersifat musiman dan murah pada saat panen raya, sehingga mengakibatkan jatuhnya harga pada saat panen, namun harga akan kembali mahal pada saat di luar musim.
4. Harga pokok ditentukan oleh kualitas, bukan kuantitas.
5. Produk hortikultura merupakan sumber vitamin dan mineral, bukan diutamakan sumber karbohidrat dan protein.
6. Selain untuk pemenuhan kebutuhan jasmani, tanaman hortikultura juga digunakan untuk pemenuhan kebutuhan rohani seperti tanaman hias.

Berdasarkan ciri-ciri tanaman hortikultura di atas, terlihat bahwa dalam pembudidayaan tanaman hortikultura harus secara insentif serta bersifat padat modal dan padat karya, artinya adalah tanaman hortikultura adalah tanaman yang dalam budidayanya membutuhkan masukan yang tinggi, namun menghasilkan keuntungan yang tinggi juga per satuan luas per satuan waktu (Zulkarnain, 2010).

Berdasarkan kegunaannya, tanaman hortikultura dikelompokkan menjadi tanaman yang dapat dikonsumsi yaitu buah-buahan, sayur-sayuran dan tanaman hortikultura yang tidak dikonsumsi yaitu tanaman hias. Dalam penelitian ini, peneliti menggunakan tanaman hortikultura kelompok sayuran karena relatif banyak ditanam oleh petani di Indonesia. Selain itu, proses penanaman dan waktu panennya yang relatif singkat dibanding kelompok buah-buahan. Tanaman hortikultura kelompok sayuran antara lain wortel, kentang, bawang merah, bawang putih, tomat dan cabai.

2.4 Pertanian Polikultur

Polikultur berasal dari kata poli yang artinya banyak dan kultur yang artinya tanaman. Secara harfiah polikultur berarti model pertanian yang menanam banyak jenis tanaman pada lahan yang sama. Polikultur adalah model pertanian yang menerapkan aspek lingkungan yang lebih baik dan melestarikan keanekaragaman hayati lokal. Sistem pertanian polikultur menerapkan model pertanian yang ekonomis, ekologis, berbudaya dan manusiawi. Model pertanian ini disebut juga dengan model pertanian yang berkelanjutan dan koreksi total terhadap model pertanian monokultur (Sabirin, et al., 2010).

Penerapan teknik budidaya secara polikultur diharapkan dapat meningkatkan *craying capacity* atau daya dukung lahan pada keadaan tertentu, dimana pertumbuhan produksi akan tetap stabil. Hasil produksi dengan sistem monokultur, petani hanya dapat memanen satu produk dalam satu periode. Namun dengan polikultur, hasil panen dalam satu periode akan bertambah dengan pemanfaatan lahan, hal ini sangat membantu peningkatan penghasilan petani (Kholifah, Trisyani & Yuniar, 2008).

Dengan pemilihan yang tepat, sistem pertanian polikultur dapat memberikan beberapa keuntungan antara lain sebagai berikut (Anindita, 2013) :

1. Mengurangi serangan organisme pengganggu tanaman (OPT)
2. Menambah kesuburan tanah
3. Siklus hidup hama atau penyakit dapat terputuskan karena sistem ini dibarengi dengan rotasi tanaman dapat memutus siklus OPT

4. Memproleh hasil panen yang beragam. Penanaman lebih dari satu jenis tanaman akan menghasilkan panen yang beragam. Ini menguntungkan karena bila harga salah satu komoditas rendah, dapat ditutup oleh harga komoditas lainnya.

Dalam sistem polikultur, pemilihan jenis tanaman menjadi sangat penting karena tanaman yang tidak sesuai dapat menyebabkan kerugian, misalnya tanaman akan berebut unsur hara, adanya tanaman lain akan mendatangkan hama dan penyakit baru maupun pertumbuhan tanaman saling terhambat. Beberapa hal yang perlu diperhatikan dalam pemilihan jenis tanaman antara lain (Anindita, 2013) :

1. Sosok tanaman dan kebutuhan sinar matahari

Pemilihan jenis tanaman yang tinggi, rindang, berdaun lebat dan membutuhkan sinar matahari lama dengan jenis tanaman yang pendek tidak membutuhkan sinar matahari lama atau perlu naungan. Sehingga kebutuhan sinar matahari dapat tercukupi dengan baik.

2. Kebutuhan unsur hara

Adanya jenis tanaman membutuhkan sedikit unsur Nitrogen (N) dan jenis tanaman yang membutuhkan banyak unsur N dan ada jenis tanaman yang mampu mengikat unsur N dari udara seperti kacang-kacangan

3. Sistem perakaran

Adanya jenis tanaman yang memiliki perakaran di dalam tanah yang dalam, dangkal, melebar dan lainnya.

2.4.1 Jenis-jenis Polikultur

Jenis jenis pola tanam polikultur terbagi menjadi ke dalam beberapa pola tanam, di antaranya sebagai berikut (Anindita, 2013) :

1. Tumpang Sari

Pengertian Tumpang sari adalah pola penanaman lebih dari satu jenis tanaman pada waktu yang bersamaan atau selama periode tanam pada satu tempat yang sama. Beberapa keuntungan dari sistem tumpangsari antara lain pemanfaatan lahan kosong disela-sela tanaman pokok, peningkatan produksi total persatuan luas karena lebih efektif dalam penggunaan cahaya, air serta unsur hara, disamping dapat mengurangi resiko kegagalan panen dan menekan pertumbuhan gulma.

2. Penanaman Bergilir

Perngertian penanaman bergilir adalah pola penanaman secara bergantian antara tanaman satu dengan tanaman lain. Pola tanam ini dapat dilakukan secara beruntun sepanjang tahun dengan mempertimbangkan faktor-faktor lain untuk mendapat keuntungan maksimum. Biasanya masyarakat melakukan pola tanam bergilir untuk menjaga kesuburan dan kandungan unsur hara pada lahan. Sebagai contohnya adalah tanaman jagung kemudian kedelai dilanjutkan padi gogo.

3. Tanaman Bersisipan

Pola tanam bersisipan adalah dengan cara menyisipkan satu atau beberapa tanaman selain tanaman pokok yang sedang ditanam. Pada umumnya tipe ini dikembangkan untuk mengintensifikan lahan. Dengan demikian kemampuan lahan untuk menghasilkan sesuatu produk pangan semakin tergali. Oleh karena itu pengelola dituntut untuk semakin jeli menentukan tanaman apa yang perlu disisipkan agar waktu dan nilai ekonomisnya dapat membantu dalam usaha meningkatkan pendapatan. Sebagai contoh adalah tanaman jagung yang disisipkan di antara tanaman kacang tanah, waktu jagung menjelang panen disisipkan kacang panjang.

4. Tanaman Campuran

Pola tanam campuran adalah dengan cara penanaman yang terdiri dari beberapa tanaman yang bercampur jadi satu, tanpa diatur jarak dan lariknya. Jadi pada pola tanam campuran ini, petani dibebaskan menyampur berbagai tanaman dalam satu lahan. Penggunaan pola tanam seperti ini dapat membuat efisiensi penggunaan lahan pertanian, tetapi riskan terhadap hama dan penyakit tanaman. Contohnya adalah menanam jagung dan kedelai secara campuran tanpa peduli jarak dan lariknya.

5. Penanaman Campuran Berkelompok

Pola tanam campuran berkelompok adalah pola tanam dengan menanam dua atau lebih tanaman pada satu lahan dan satu waktu yang sama. Pola penanaman ini biasanya digunakan untuk membagi petak lahan yang tersedia untuk ditanami beberapa jenis tanaman, sehingga hasil panen bisa beragam.

Contohnya adalah menanam jagung pada sebagian lahannya dan menanam kedelai pada lahan sisanya.

6. Penanaman Lorong

Penanaman Lorong yaitu menanam lebih dari satu jenis tanaman pada suatu lahan dengan penanaman tanaman berumur pendek diantara larikan atau lorong tanaman berumur panjang atau tanaman tahunan.

2.5 Optimasi

Optimasi adalah suatu permasalahan komputasional yang bertujuan untuk menemukan solusi terbaik dari beberapa solusi yang mungkin. Secara lebih formal, optimasi adalah menemukan solusi yang berada dalam daerah yang mungkin (*feasible region*) yang memiliki nilai minimum atau maksimum dari fungsi objektif (Mahmudy, 2013).

Optimasi didefinisikan sebagai proses pemilihan sebuah solusi dari sejumlah alternatif solusi dengan memenuhi sejumlah batasan (*constraints*). Misalkan pada permasalahan pencarian rute untuk mengunjungi sejumlah kota. Pada kasus ini tentu saja terdapat banyak alternatif pilihan rute untuk dijadikan solusi. Solusi yang dipilih disesuaikan dengan tujuan dari permasalahan ini, misalkan memilih rute terpendek atau rute dengan waktu tempuh tercepat. Batasan yang ada misalkan setiap kota harus dikunjungi tepat satu kali (Mahmudy, 2013).

Untuk menyelesaikan kasus optimasi, dapat digunakan metode heuristik, yaitu metode pencarian yang didasarkan atas intuisi atau aturan-aturan empiris untuk memperoleh solusi yang lebih baik daripada solusi yang telah dicapai sebelumnya. Metode optimasi tidak selalu menghasilkan solusi terbaik, namun jika dirancang dengan baik, dapat menghasilkan solusi yang mendekati optimum dalam waktu yang relatif cepat. Metode heuristik yang bisa diterapkan dalam masalah optimasi misalnya algoritma koloni semut, algoritma hill climbing, algoritma simulated annealing dan algoritma evolusi (Mahmudy, 2013)

2.6 Algoritma Genetika

Algoritma Genetika atau *Genetic Algorithm* (GA) dikenalkan oleh John Holland dalam menyelesaikan masalah optimasi. Algoritma Genetika mensimulasikan proses yang terjadi pada populasi alamiah yang merupakan hal

yang penting dalam proses evolusi. Algoritma Genetika adalah metode pencarian yang meniru perumpamaan evolusi biologis alami untuk menentukan kromosom atau individu berkualitas tinggi dalam suatu kawasan berbingka potensial yang disebut populasi. Proses pemilihan individu dari suatu populasi dievaluasi berdasarkan fungsi *fitness*. Kromosom berwujud string tersebut merupakan calon pada setiap siklus operasi yang disebut generasi.

Algoritma Genetik (*Genetic Algorithms, Gas*) merupakan tipe *evolutionary algorithm* yang paling banyak diterapkan. Algoritma genetika berkembang seiring dengan perkembangan teknologi informasi yang sangat pesat dan diilhami oleh ilmu genetika, karena itu istilah yang digunakan dalam algoritma genetika banyak diadopsi dari ilmu tersebut. Apabila dibandingkan dengan prosedur pencarian dan optimasi biasa, algoritma genetika berbeda dalam beberapa hal sebagai berikut (Mahmudy, 2013) :

- Manipulasi dilakukan terhadap kode dari himpunan parameter (kromosom), tidak secara langsung terhadap parameternya sendiri.
- Proses pencarian dilakukan dari beberapa titik dalam satu populasi, tidak dari satu titik saja.
- Proses pencarian menggunakan informasi dari fungsi tujuan.
- Proses pencarian algoritma genetika menggunakan *stochastic operations* yang bersifat probabilistik, tidak menggunakan aturan deterministik.

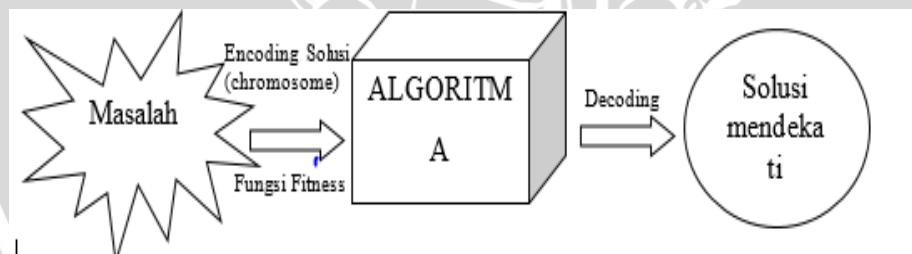
Sebagai metode optimasi, algoritma genetika memiliki beberapa kelebihan sebagai berikut (Mahmudy, 2013) :

1. Algoritma genetika merupakan algoritma yang berbasis populasi yang memungkinkan digunakan pada optimasi masalah dengan ruang pencarian (*search space*) yang sangat luas dan kompleks. Properti ini juga memungkinkan algoritma genetika untuk melompat keluar dari daerah optimum lokal.
2. Individu yang ada pada populasi bisa diletakkan pada beberapa sub-populasi yang diproses pada sejumlah komputer secara paralel. Hal ini dapat mengurangi waktu komputasi pada masalah yang sangat kompleks. Sehingga penyelesaian permasalahan dapat dilakukan dalam waktu yang relatif singkat.

3. Algoritma genetika dapat digunakan untuk menyelesaikan masalah yang kompleks dengan banyak variable. Variable tersebut bisa kontinyu, diskrit atau campuran keduanya.
4. Algoritma genetika menggunakan kromosom untuk mengkodekan solusi sehingga bisa melakukan pencarian tanpa memperhatikan informasi derivatif yang spesifik dari masalah yang diselesaikan.
5. Algoritma genetika bisa diimplementasikan pada berbagai macam data seperti data yang dibangkitkan secara numerik atau menggunakan fungsi analitis.
6. Algoritma Genetika bersifat *ergodic*, sembarang solusi dapat diperoleh dari solusi yang lain dengan hanya beberapa langkah. Hal ini memungkinkan eksplorasi pada daerah pencarian yang sangat luas dilakukan dengan lebih cepat dan mudah.
7. Algoritma genetika cukup fleksibel untuk dihibridisasikan dengan algoritma lainnya untuk mendapatkan solusi yang lebih baik.

2.6.1 Struktur Algoritma Genetika

Penggunaan Algoritma Genetika untuk memecahkan suatu masalah ditunjukkan pada Gambar 2.1.



Gambar 2.1 Mencari Solusi Dengan Algoritma Genetika
Sumber : (Mahmudy, 2013)

Proses yang terdapat pada algoritma genetika ini mengadopsi dari teori evolusi. Pada algoritma genetika diawali dengan inialisasi sebuah populasi, yaitu menciptakan individu-individu secara acak yang memiliki susunan gen (kromosom) tertentu. Kromosom ini mewakili solusi dari permasalahan yang dipecahkan. Proses reproduksi perlu dilakukan untuk menghasilkan keturunan (*offspring*) dari individu-individu yang ada pada suatu populasi. Proses evaluasi digunakan untuk menghitung kebugaran (*fitness*) setiap kromosom. Semakin besar *fitness* maka semakin baik kromosom tersebut untuk dijadikan calon solusi.

Proses seleksi dilakukan untuk memilih individu dari himpunan populasi dan *offspring* yang akan dipertahankan hidup untuk menjadi sekumpulan populasi pada generasi berikutnya. Fungsi *probabilistic* digunakan untuk memilih individu yang dipertahankan hidup. Individu yang lebih baik (mempunyai nilai *fitness* lebih besar) mempunyai peluang lebih besar untuk terpilih (Mahmudy, 2013).

Setelah melewati sekian iterasi (generasi) akan didapatkan sebuah individu terbaik. Individu terbaik ini mempunyai susunan kromosom yang bisa dikonversi menjadi solusi yang terbaik (paling tidak mendekati optimum). Dari sini bisa disimpulkan bahwa algoritma genetika menghasilkan suatu solusi yang mendekati optimum dengan melakukan pencarian solusi yang terbaik dari sejumlah alternatif titik optimum berdasarkan fungsi *probabilistic* (Mahmudy, 2013).

2.6.2 Parameter Algoritma Genetika

Algoritma genetika bekerja berdasarkan parameter – parameter tertentu yang akan mempengaruhi kinerja dan perilaku dari algoritma dalam mencapai solusi optimal dari permasalahan yang diselesaikan. Parameter – parameter penting yang mempengaruhi performa algoritma genetika diantaranya adalah (Mahmudy, 2013):

1. Ukuran Populasi

Ukuran populasi merupakan jumlah kromosom yang ada pada populasi dalam satu generasi. Kromosom sebagai alternatif solusi pada algoritma genetika. Sehingga semakin banyak suatu ukuran populasi, proses pencarian solusi dalam algoritma genetika akan semakin optimal. Namun, algoritma genetika akan bergerak lambat jika terlalu banyak jumlah kromosomnya, karena akan semakin banyak jumlah individu yang akan diproses dalam algoritma genetika.

2. Jumlah Generasi

Generasi merupakan jumlah iterasi yang dilakukan terhadap proses evaluasi tiap-tiap populasi. Generasi digunakan untuk mempertahankan individu (kromosom) baik yang mampu bertahan untuk proses pada generasi selanjutnya. Jumlah generasi yang tepat harus diperhitungkan dalam melakukan proses optimasi menggunakan algoritma genetika. Untuk mengetahui jumlah generasi yang tepat pada suatu penyelesaian masalah, perlu dilakukan proses pengujian terhadap ukuran generasi.

3. Crossover Rate

Crossover rate (cr) merupakan seberapa persen dari total kromosom yang akan melalui proses perkawinan silang (*crossover*). Bila tidak terjadi *crossover*, *offspring* (kromosom anak hasil *crossover*) merupakan salinan yang serupa dari induk (*parent*). Bila terjadi *crossover*, *offspring* disusun dari bagian-bagian kromosom induk (*parent*). *Crossover* dilakukan dengan harapan kromosom-kromosom baru akan memiliki bagian-bagian yang baik dari kromosom-kromosom terdahulu. Untuk mengetahui nilai parameter *crossover rate* yang tepat pada penyelesaian masalah menggunakan algoritma genetika, perlu dilakukan proses pengujian terhadap nilai *crossover rate* yang optimal.

4. Mutation Rate

Mutation rate (mr) menunjukkan seberapa sering bagian-bagian kromosom akan bermutasi. Nilai *mutation rate* akan menjadi penentu jumlah anak hasil mutasi. Untuk mengetahui nilai parameter yang optimal dalam algoritma genetika, perlu juga dilakukan pengujian terhadap nilai *mutation rate*. Rekomendasi untuk menentukan nilai parameter adalah sebagai berikut:

Tidak ada metode yang pasti untuk menentukan nilai parameter dalam algoritma genetika. Kombinasi yang tepat untuk nilai parameter tersebut sangat dipengaruhi oleh permasalahan yang akan diselesaikan, sehingga perlu dilakukan serangkaian pengujian pedahuluan untuk mendapatkan kombinasi nilai parameter yang sesuai. Kombinasi parameter yang sesuai dengan ukuran populasi (*popSize*) antara 30 sampai 50, cr antara 0.3 sampai 0.8, dan mr antara 0.1 sampai 0.3 (Mahmudy, 2013).

2.6.3 Proses Algoritma Genetika

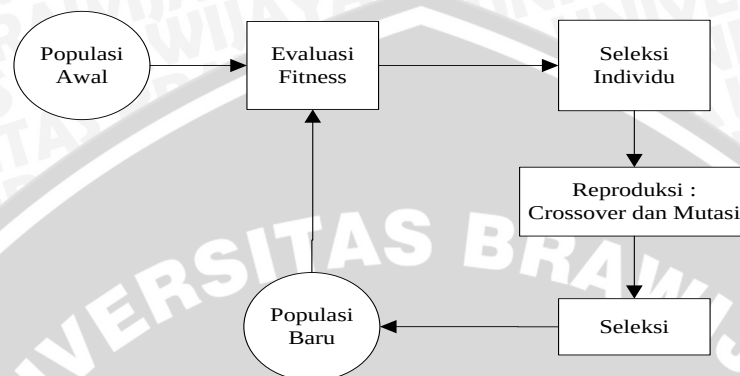
Langkah umum dalam pemecahan masalah dengan menggunakan algoritma genetika adalah sebagai berikut (Wardhani, 2011) :

- Melakukan inisialisasi populasi kromosom dengan solusi secara acak (*random*).
- Melakukan evaluasi setiap kromosom dalam populasi menggunakan persamaan fungsi evaluasi (*fitness function*).
- Memilih sebagian anggota populasi sebagai solusi yang sesuai untuk generasi selanjutnya.

d. Menciptakan solusi (keturunan) baru dengan mengawinkan solusi dari induknya dengan cara *crossover* dan mutasi.

e. Membuang anggota populasi lama yang memiliki nilai *fitness* rendah.

Siklus umum proses Algoritma Genetika ditunjukkan pada Gambar 2.2.



Gambar 2.2 Siklus Umum Algoritma Genetika
Sumber : (Wardhani, 2011)

2.7 Penerapan Algoritma Genetika

Ada beberapa langkah yang harus dilakukan dalam penerapan algoritma genetika. Berikut langkah dalam penerapan algoritma genetika tersebut (Mahmudy, 2013) :

2.7.1 Generate populasi awal

Membangkitkan populasi awal adalah proses membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu untuk mencari penyelesaian optimal. Populasi awal yang dibangun pada tugas akhir ini menggunakan bilangan dengan range yang telah ditentukan.

2.7.2 Representasi Kromosom

Representasi kromosom merupakan proses pengkodean dari penyelesaian asli suatu permasalahan. Solusi dari suatu masalah harus dipetakan (*encoding*) menjadi string kromosom. String kromosom ini tersusun atas sejumlah gen yang menggambarkan variabel-variabel keputusan yang digunakan dalam solusi (Mahmudy, 2013). Dalam berbagai macam kasus, representasi sebuah solusi menjadi kromosom sangat menentukan kualitas dari solusi yang dihasilkan (Mahmudy, 2013).

Ada berbagai macam representasi kromosom, diantaranya adalah representasi biner, representasi *real coded*, representasi integer, representasi matriks, dan representasi permutasi (Mahmudy, 2013). Representasi biner calon solusi dinyatakan dalam bilangan biner yaitu 1 dan 0. Representasi integer dinyatakan dengan angka integer. Pada skripsi ini digunakan representasi *real coded* atau *Real Coded Genetic Algoritim*..

Contoh representasi kromosom dalam penelitian ini, misal pengguna memilih 3 jenis tanaman yang diinginkan yaitu tanaman 1, tanaman 2 dan tanaman 3. Maka inisialisasi populasi akan dibangkitkan bilangan random dimana setiap kromosom terdiri dari 3 gen. Contoh pembangkitan populasi awal ditunjukan pada Tabel 2.2.

Tabel 2.2 Representasi Kromosom

<i>Parent</i>	1	2	3
	0.1403	0.6428	0.2170

Tabel 2.2 menunjukan kromosom yang memiliki 3 gen yang terdiri atas 3 jenis tanaman dan masing-masing gen mewakili proporsi luas lahan yang direkomendasikan antara lain tanaman 1 dengan proporsi 0.1403 dari luas lahan yang dimiliki, tanaman 2 dengan proporsi 0.6428 dari luas lahan yang dimiliki serta tanaman 3 dengan proporsi 0.2170 dari luas lahan yang dimiliki.

2.7.3 Crossover

Crossover dilakukan dengan memilih dua induk (*parent*) secara acak dari populasi. Dalam tahap *crossover* harus ditentukan *crossover rate* (cr). Nilai cr menyatakan rasio *offspring* yang dihasilkan proses *crossover* terhadap ukuran populasi sehingga akan dihasilkan *offspring* sebanyak $cr \times popSize$ (jumlah populasi). Misal , jika ditentukan $cr = 0.5$ dan *popSize* 10, maka hasil dari proses *crossover* adalah $0.5 \times 10 = 5$ *offspring*. Secara umum, mekanisme *crossover* adalah sebagai berikut (Mahmudy, 2013) :

1. Memilih dua buah kromosom sebagai *parent*.
2. Memilih secara acak populasi dalam kromosom, sehingga masing – masing kromosom *parent* terpisah menjadi dua segmen.
3. Menukar antar segmen kromosom induk untuk menghasilkan kromosom anak.

Pada penelitian ini, metode *crossover* yang digunakan adalah metode *extended intermediate crossover*. *Extended intermediate crossover* menghasilkan *offspring* dari kombinasi nilai dua induk. Misalkan P1 dan P2 adalah dua kromosom *Parent*, maka *offspring* C1 dan C2 dapat dibangkitkan dengan persamaan (2-1) dan (2-2) (Mahmudy, 2013) :

$$C1 = P1 + \alpha (P2 - P1) \dots\dots\dots(2-1)$$

$$C2 = P2 + \alpha (P1 - P2) \dots\dots\dots(2-2)$$

Contoh *extended intermediate exchange* ditunjukkan pada Gambar 2.3. Nilai α dibangkitkan secara acak pada interval 0 sampai 1. Dengan perhitungan pada persamaan (2-1) dan (2-2) didapatkan hasil *offspring crossover* seperti pada Gambar 2.3.

α	0.6289	0.7148	0.3165
Parent 1	0.1403	0.6428	0.2170
Parent 2	0.2841	0.3448	0.3711
Offspring 1	0.2307	0.4298	0.2657
Offspring 2	0.1936	0.5578	0.3223

Gambar 2.3 Contoh *Crossover*

2.7.4 Mutasi

Mutasi menciptakan individu baru dengan melakukan modifikasi satu atau lebih gen dalam individu yang sama. Mutasi berfungsi untuk menggantikan gen yang hilang dari populasi selama proses seleksi serta menyediakan gen yang tidak ada dalam populasi awal. Mutasi akan meningkatkan variasi populasi. Misal, jika ditentukan $p_m = 0.2$ dan jumlah populasi 10 maka ada $0.2 \times 10 = 2$ *offspring* yang akan dihasilkan dari proses mutasi (Mahmudy, 2013).

Mutasi dilakukan setelah proses *crossover*. Fungsi dari mutasi adalah untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Sehingga mutasi akan meningkatkan variasi populasi.

Pada penelitian ini metode mutasi yang digunakan adalah *random mutation*. Metode mutasi ini memilih satu induk secara acak dari populasi. Kemudian membangkitkan nilai gen terpilih dengan persamaan yang sudah ditentukan.

Misalkan domain variabel x_j adalah $[min_j, max_j]$ dan *offspring* yang dihasilkan adalah $C=[x'_1...x'_n]$, maka nilai gen *offspring* bisa dibangkitkan sebagai berikut (Mahmudy, 2013) :

$$x'_i = x_i + r (max_i - min_i) \dots\dots\dots(2-3)$$

Nilai *range* (r) didapatkan dari membangkitkan nilai acak pada range yang telah ditentukan, misalkan [0; 1]. Sementara nilai max dan min didapat dari nilai maksimal dan minimal gen hasil pembangkitan populasi awal.

Contoh mutasi pada penelitian ini ditunjukkan pada Gambar 2.4 :

$r = 0.0471$ $max = 0.7201$ $min = 0.2110$

<i>Parent</i>	0.1803	0.5195	0.3002
<i>Offspring</i>	0.1803	0.5435	0.3002

Gambar 2.4 Contoh Proses Mutasi

2.7.5 *Fitness*

Sebuah fungsi *fitness* digunakan untuk mengukur seberapa baik suatu individu. Individu terbaik diakhir generasi bisa dikodekan sebagai solusi terbaik yang bisa diperoleh. Semakin besar nilai *fitness*, maka semakin baik individu tersebut untuk dijadikan calon solusi (Mahmudy, 2013). Fungsi *fitness* pada penelitian ini dihitung dengan memaksimalkan keuntungan yang diperoleh, sehingga penentuan persamaan *fitness* menggunakan fungsi obyektif dari permasalahan yang akan dimaksimalkan. Sehingga *fitness* yang terbaik akan menjadi solusi optimasi.

2.7.6 Seleksi

Seleksi dilakukan untuk memilih individu dari himpunan populasi dan *offspring* yang dipertahankan pada generasi berikutnya (Mahmudy, 2013). Fungsi probabilitas digunakan untuk memilih individu yang dipertahankan hidup. Individu yang mempunyai *fitness* lebih besar mempunyai peluang besar untuk terpilih. Metode seleksi yang sering digunakan adalah *roulette wheel*, *replacement*, *elitism* dan *binary tournament*.

2.7.6.1 Seleksi *Roulette Wheel*

Seleksi *roulette wheel* adalah metode seleksi dengan melakukan perhitungan nilai probabilitas seleksi (prob) setiap individu berdasarkan nilai *fitness* nya. Dari nilai probabilitas setiap individu ini dapat dihitung nilai probabilitas kumulatif (propCom) yang digunakan pada proses seleksi setiap individu.

Langkah-langkah dalam seleksi *roulette wheel* berdasarkan probabilitas kumulatif adalah (Mahmudy, 2013) :

- Misalnya *fitness* (P_k) adalah nilai *fitness* individu ke- k . Maka bisa dihitung total *fitness* dalam sebuah populasi dengan Persamaan (2-3) (Mahmudy, 2013) :

$$totalFitness = \sum_{k=1}^{populasi} Fitness (P_k) \dots \dots \dots (2-4)$$

- Nilai probabilitas seleksi (prob) setiap individu dapat dihitung dengan persamaan (2-4).

$$Prob(k) = \frac{Fitness P(k)}{totalFitness} \dots \dots \dots (2-5)$$

- Nilai probabilitas kumulatif (probCum) setiap individu dapat dihitung dengan persamaan (2-5).

$$propCum(k) = \sum_{j=1}^k prob (j) \dots \dots \dots (2-6)$$

2.7.6.2 Seleksi *Replacement*

Replacement selection adalah seleksi yang menjamin individu terbaik selalu lolos. Tetapi pada seleksi ini tidak menutup kemungkinan untuk peluang individu dengan nilai *fitness* rendah untuk lolos ke generasi berikutnya, karena solusi optimum mungkin didapatkan dari hasil reproduksi individu-individu dengan hasil *fitness* rendah. Metode seleksi *Replacement selection* mempunyai dua aturan (Mahmudy, 2013) :

1. *Offspring* yang diproduksi melalui proses mutasi menggantikan induknya jika mempunyai nilai *fitness* yang lebih baik
2. *Offspring* yang diproduksi melalui proses *crossover* (menggunakan dua induk dan menghasilkan satu anak), akan menggantikan induk yang terlemah jika mempunyai nilai *fitness* yang lebih baik daripada induk yang terlemah tersebut.

Pseudo-code seleksi *elitism* dideskripsikan sebagai berikut (Mahmudy, 2013) :

PROCEDURE ReplacementSelection

Input:

POP: himpunan individu pada populasi
 OS: himpunan individu anak (*offspring*) hasil reproduksi menggunakan *crossover* dan mutasi
 ns: banyaknya *offspring*
 P: list dari indeks parent individu dalam *offspring*

Output :

POP: himpunan individu pada populasi setelah proses seleksi selesai
 FOR i=1 TO ns DO
 //get index of parent
 p = P1
 IF Fitness (OS_i) > Fitness (POP_p) THEN
 POP_p ← OS_i
 End IF

END PROCEDURE

Gambar 2.5 Pseudocode Seleksi Replacement

2.7.6.3 Seleksi Elitism

Seleksi *elitism* adalah seleksi dimana individu – individu yang terpilih untuk menjadi generasi selanjutnya berdasarkan pada nilai *fitness* tertinggi. Metode seleksi *elitism* bekerja dengan mengumpulkan semua individu dalam populasi (*popSize*) dan *offspring* dalam satu penampungan. Individu terbaik dalam penampungan akan lolos untuk masuk dalam generasi berikutnya. Metode seleksi *elitism* menjamin individu yang terbaik akan selalu lolos (Mahmudy, 2013).

Pseudo-code seleksi *elitism* dideskripsikan sebagai berikut (Mahmudy, 2013):

PROCEDURE ElitismSelection

Input:

POP: himpunan individu pada populasi
 pop_size: ukuran populasi
 OS: himpunan individu anak (*offspring*) hasil reproduksi menggunakan *crossover* dan mutasi

Output :

POP: himpunan individu pada populasi setelah proses seleksi selesai

```

TEMP <- Merge (POP, OS)
OrderAscending (Temp)
POP <- CopyBest (Temp, pop_size)
END PROCEDURE

```

Gambar 2.6 Pseudocode Seleksi Elitism

Kelemahan metode seleksi *elitism* adalah metode ini tidak mengijinkan individu dengan *fitness* rendah untuk masuk ke populasi berikutnya. Dalam beberapa kasus solusi optimum dicapai pada individu dengan *fitness* rendah (Mahmudy, 2013).

2.7.6.2 Seleksi *Binary Tournament*

Binary tournament selection merupakan salah satu metode seleksi terpopuler dalam algoritma genetika karena efisiensi dan implementasi yang sederhana. Dalam seleksi *tournament*, n individu dipilih secara acak sebanyak *popSize*. Banyaknya perbandingan dalam turnamen terhadap individu disebut *tournament size*. Satu individu akan bersaing dengan individu lain untuk menentukan nilai *fitness* tertinggi yang akan menjadi pemenang dan individu sebagai pemenang akan terpilih sebagai generasi berikutnya. Seleksi turnamen juga memberikan kesempatan pada semua individu terpilih untuk mempertahankan keragamannya.

Pseudo-code seleksi *Binary Tournament* dideskripsikan sebagai berikut (Mahmudy, 2013).:

```

PROCEDURE BinaryTournamentSelection
Input:
POP: himpunan individu pada populasi
pop_size: ukuran populasi
OS: himpunan individu anak (offspring) hasil reproduksi
menggunakan crossover and mutasi
ns: banyaknya offspring
Output:
POP: himpunan individu pada populasi setelah proses seleksi
selesai
FOR i=1 TO ns DO
p = Random (1, pop_size)
IF Fitness(OSi) > Fitness(POPp) THEN

```


POp □ Osi
 END IF
 END FOR
END PROCEDURE

Gambar 2.7 *Pseudocode Seleksi Binary Tournament*



BAB III

METODOLOGI PENELITIAN

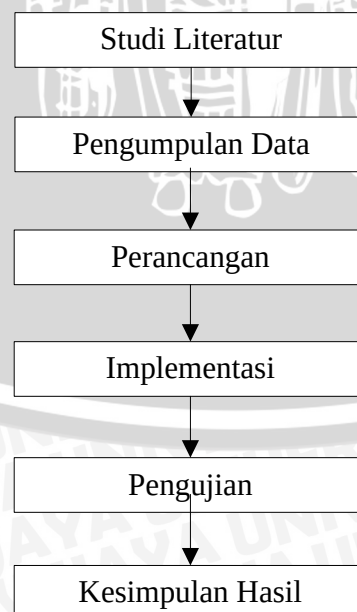
Pada bab metodologi penelitian ini akan dijelaskan tahapan dalam pembuatan sistem optimasi penggunaan lahan pertanian dengan algoritma genetika. Pada bab ini, juga akan dibahas mengenai tahapan penelitian, kebutuhan sistem, formulasi permasalahan dan siklus penyelesaian algoritma genetika beserta contoh perhitungan manualnya.

3.1 Tahapan Penelitian

Pada sub bab tahapan penelitian ini, akan dibahas langkah-lagkah yang digunakan dalam pembuatan sistem optimasi penggunaan lahan pertanian dengan algoritma genetika. Tahapan pembuatannya adalah sebagai berikut :

1. Melakukan studi literatur mengenai algoritma dalam permasalahan optimasi
2. Menganalisa dan merancang sistem
3. Membuat sistem berdasarkan analisa dan perancangan yang dilakukan
4. Melakukan ujicoba terhadap sistem
5. Melakukan evaluasi hasil yang diperoleh dari uji coba tersebut dengan cara membandingkan generasi tiap populasi.

Diagram alir tahapan penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Tahapan Penelitian

Berdasarkan Gambar 3.1, maka tahap-tahap yang harus dilakukan pada penelitian ini adalah :

3.1.1 Studi Literatur

Pada tahap ini dilakukan pengumpulan referensi yang berhubungan dengan penelitian ini. Tujuan dari pengumpulan referensi adalah sebagai dasar teori atau sumber penunjang dalam penelitian. Teori-teori yang berhubungan dengan penelitian meliputi :

1. Optimasi
2. Algoritma genetika
3. Polikultur
4. Palawija dan hortikultura

Referensi yang digunakan pada penelitian ini berasal dari jurnal nasional dan internasional, skripsi, buku, dan artikel-artikel yang terdapat di internet. Selain itu juga dilakukan wawancara dengan pihak yang terkait dengan permasalahan yang akan diselesaikan, yaitu dosen Fakultas Pertanian Universitas Brawijaya dan Dinas Pertanian Kabupaten Tulungagung.

3.1.2 Analisis Kebutuhan

Pada tahapan ini, akan dilakukan analisis kebutuhan sistem atau bagaimana aplikasi ini akan berjalan.

1. Sistem dapat menerima masukan berupa nilai parameter algoritma genetika, luas lahan dan modal yang dimiliki.
2. Sistem mampu menampilkan masukan dari pengguna.
3. Sistem mampu menampilkan hasil kromosom terbaik pada setiap generasi
4. Sistem mampu menampilkan kromosom terbaik dari generasi terakhir
5. Sistem mampu menampilkan rekomendasi proporsi setiap jenis tanaman yang akan ditanam dan informasi keuntungan yang didapat.

3.1.3 Perancangan

Pada tahap ini akan dilakukan perancangan dari implementasi algoritma genetika untuk optimasi penggunaan lahan pertanian. Perancangan meliputi perancangan basis data, perancangan antar muka (*user interface*) dan perancangan

pengujian. Penjelasan lebih detail tentang perancangan akan dijelaskan pada bab berikutnya.

3.1.4 Implementasi

Pada tahap ini akan dilakukan proses implementasi algoritma genetika untuk optimasi penggunaan lahan pertanian berdasarkan perancangan yang telah dilakukan sebelumnya ke dalam kode program.

3.1.5 Pengujian

Pada tahap ini akan dilakukan pengujian dari sistem yang telah dibuat pada proses implementasi. Pengujian dilakukan dengan menguji parameter-parameter dari algoritma genetika yang meliputi :

1. Pengujian generasi
2. Pengujian populasi
3. Pengujian *crossover rate* dan *mutation rate*
4. Pengujian seleksi

Dari pengujian tersebut akan didapatkan parameter optimal yang dapat digunakankan untuk menyelesaikan permasalahan dalam penelitian ini.

3.1.6 Pengambilan Kesimpulan

Pengambilan kesimpulan merupakan metode yang dilakukan setelah tahapan perancangan, implementasi dan pengujian selesai dilakukan. Kesimpulan yang diambil dilakukan berdasarkan kesesuaian teori dan praktik. Kesimpulan dibuat untuk menjawab rumusan masalah yang telah disusun.

3.2 Kebutuhan Sistem

Analisa kebutuhan sistem merupakan tahapan untuk menganalisa semua kebutuhan yang diperlukan dalam penyelesaian penelitian ini agar sistem yang akan dibuat dapat berjalan dengan optimal. Analisa kebutuhan sistem ada dua macam yaitu kebutuhan perangkat keras dan kebutuhan perangkat lunak.

3.2.1 Kebutuhan Perangkat Keras

Perangkat keras yang digunakan untuk pengembangan perangkat lunak dalam penelitian ini memiliki spesifikasi sebagai berikut :

1. Processor Intel(R) Core i3 CPU @210 GHz

2. Memori 2GB
3. *Harddisk* dengan kapasitas 520 GB

3.2.2 Kebutuhan Perangkat Lunak

Kebutuhan Perangkat Lunak yang digunakan dalam penelitian ini adalah :

1. Sistem Operasi Microsoft Windows 8 Ultimate *edition* 32 bit.
2. Bahasa pemrograman Vb.Net dengan *editor* Microsof Visual Studio 2010.
3. SQL server sebagai *Database Management System* (DBMS).
4. Microsoft Office Visio 2007 untuk pembuatan diagram alir.

3.3 Formulasi Permasalahan

Formulasi permasalahan merupakan tahapan untuk identifikasi permasalahan dan kebutuhan data yang digunakan untuk menyelesaikan permasalahan pada penelitian ini.

3.3.1 Deskripsi Umum Sistem

Sistem yang akan dibangun dalam penelitian ini adalah sebuah sistem yang mengimplementasikan algoritma genetika untuk permasalahan optimasi penggunaan lahan pertanian agar mendapatkan kombinasi jenis tanaman dan proporsi luas yang optimal, sehingga mendapatkan hasil atau keuntungan yang maksimal dengan mengalokasikan luas lahan dan modal yang dimiliki. Pengguna akan diminta untuk memasukkan luas lahan, modal yang dimiliki dan jenis tanaman yang akan ditanam. Selanjutnya akan diproses dengan algoritma genetika untuk mendapatkan rekomendasi proporsi luas lahan pada masing-masing jenis tanaman yang sudah dipilih tersebut, sehingga hasil produksi atau keuntungan yang didapat nantinya akan maksimal dengan mengalokasikan luas lahan dan modal yang dimiliki.

3.3.2 Data yang Digunakan

Dalam penyelesaian masalah optimasi penggunaan luas lahan pertanian ini, peneliti menggunakan data primer, yaitu data yang didapat langsung dari responden penelitian. Metode pengumpulan data primer bersifat kuantitatif dengan cara wawancara. Selain itu juga digunakan metode pengumpulan data sekunder dengan

mengumpulkan literatur yang terkait dengan penelitian. Data yang dibutuhkan dalam penelitian ini yaitu :

1. Data biaya produksi pada masing-masing jenis tanaman per meter persegi.
2. Data keuntungan hasil produksi pada masing-masing jenis tanaman per meter persegi.
3. Data luas lahan minimal pada masing-masing jenis tanaman.
4. Data jarak tanam dan umur panen optimal pada masing-masing jenis tanaman.

Analisa kebutuhan data beserta sumber, metode, dan kegunaannya ditunjukkan pada Tabel 3.1.

Tabel 3.1 Analisis kebutuhan data penelitian

No	Kebutuhan Data	Sumber Data	Metode	Kegunaan Data
1.	Data total biaya produksi pada masing-masing jenis tanaman per meter persegi.	Jurusan Sosial Ekonomi Fakultas Pertanian Universitas Brawijaya	Literatur, Wawancara	Sebagai salah satu dasar penentuan jumlah biaya yang dikeluarkan pada tanaman tertentu dengan satuan luas tertentu.
2.	Data keuntungan hasil produksi pada masing-masing jenis tanaman per meter persegi.	Jurusan Sosial Ekonomi Fakultas Pertanian Universitas Brawijaya.	Literatur, Wawancara	Digunakan untuk analisis hasil keuntungan setiap jenis tanaman yang akan ditanam dengan satuan luas tertentu.
3.	Data luas lahan minimal pada masing-masing jenis tanaman.	Dinas Pertanian Kabupaten Tulungagung.	Wawancara	Digunakan sebagai acuan <i>constrain</i> proporsi luas lahan optimal
4.	Data jarak tanam dan usia panen optimal pada masing-masing jenis tanaman	Jurusan Sosial Ekonomi Fakultas Pertanian Universitas Brawijaya	Literatur, Wawancara	Digunakan sebagai informasi bagi pengguna.

Hasil data pada setiap jenis tanaman palawija dan hortikultura kelompok sayur-sayuran yang diperoleh dari studi literatur dan wawancara ditunjukan pada Tabel 3.2. Data yang digunakan merupakan data hasil wawancara dan survei yang

dilaksanakan pada tahun 2014. Sehingga pada penelitian ini disediakan fitur untuk *update data*.

Tabel 3.2 Data Tanaman

Kode	Nama Tanaman	Jarak Tanam (cm)	Usia Panen (Hari)	Biaya ProduksiI (Rp/M ²)	Keuntungan produksi (Rp/M ²)	Lahan Minimal (M ²)
T001	JAGUNG	40x90	85-95	926.00	764.00	1900
T002	KACANG TANAH	30x20	100-110	1931.00	2909.00	1200
T003	KEDELAI	40x40	85-90	1100.00	818.00	1200
T004	KACANG HIJAU	20X20	58-85	600.00	527.00	800
T005	UBI	75X30	120-130	414.00	1688. 00	1600
T006	SINGKONG	100X100	210-215	365.00	1485. 00	1000
T007	BAWANG MERAH	20x20	95-115	5401.00	4791.69	2000
T008	BAWANG PUTIH	20x20	95-115	5120. 00	4080. 00	2000
T009	KACANG PANJANG	40X60	>50	1759. 00	2401. 00	1180
T010	KENTANG	70x30	90-180	4381. 00	4069. 00	4200
T011	CABAI BESAR	60x70	75-85	3240. 00	3040. 00	1000
T012	CABAI KECIL	60x70	80-85	3720. 00	3505. 00	1000
T013	WORTEL	45x50	90-110	725. 00	2014. 00	1500
T014	TERONG	60x60	75-100	1989. 00	1520. 00	1600
T015	KUBIS	50x60	65	1046. 46	1303. 49	3000

Sumber : (Wawancara)

Dari data pada tabel 3.2 yang akan dijadikan parameter dalam proses perhitungan dengan algoritma genetika adalah data kuantitatif biaya produksi, keuntungan produksi dan lahan minimal pada setiap jenis tanaman. Sementara data jarak tanam dan usia panen adalah data informasi bagi pengguna untuk penentuan dan pengaturan jarak tanam serta usia panen yang optimal pada masing-masing tanaman yang akan ditanam. Sehingga dengan informasi ini, pengguna dapat melakukan proses penanaman sesuai dengan jarak tanam terbaik yang direkomendasikan oleh pakar sesuai data tersebut.

3.3.3 Fungsi Obyektif Permasalahan

Pada penelitian ini, akan dilakukan optimasi untuk memaksimalkan keuntungan dengan kendala yang ada. Sehingga, dimunculkan sebuah persamaan untuk menghitung fungsi obyektif ($f(x)$) dari permasalahan yang akan diselesaikan dengan persamaan (3-1). Nilai alfa adalah suatu nilai yang digunakan untuk

preferensi dari sebuah penalti. Karena dalam permasalahan optimasi ini adalah pencarian nilai maksimal, maka nilai *fitness* sama dengan nilai fungsi obyektif (Mahmudy, 2013) :

$$Fitness = (L * \sum_{i=1}^n (Xi * Keuntungan [i])) - \alpha_1 * Penalti_1 - \alpha_2 * Penalti_2 \dots (3-1)$$

Keterangan :

Fitness = fungsi obyektif yang dimaksimalkan

L = luas lahan

X_i = proporsi luas tanaman ke- i

α_1 = nilai konstanta 1

α_2 = nilai konstanta 2

Dari fungsi obyektif yang dibangkitkan pada persamaan (3-1) terdapat dua kendala atau penalti pada suatu individu. Penalty adalah suatu bobot yang digunakan ketika individu melakukan pelanggaran terhadap aturan (Tyas, 2013). Nilai sebuah penalti adalah selisih dari pelanggaran yang dilakukan terhadap aturan (Sari, 2014). Aturan yang digunakan dalam penelitian ini adalah total biaya produksi pada satu kali penanaman tidak boleh lebih dari modal yang dimiliki dan lahan minimal setiap tanaman yang dipilih, tidak boleh kurang dari lahan minimal yang ditetapkan pakar sesuai dengan data yang tertera pada tabel 3.2. Bagi suatu individu atau kromosom yang melanggar aturan tersebut, akan dikenakan suatu penalti. Secara umum, perhitungan penalti ditunjukkan pada persamaan (3-2) berikut ini (Smith & Coit, 1996) :

$$Penalti = \sum_{i=1}^n C_i \delta_i \dots \dots \dots (3-2)$$

Keterangan :

C_i = Sebuah konstanta yang dikenakan untuk pelanggaran kendala atau penalti ke- i

δ_i = Kendala atau penalti ke- i yang dilanggar

n = Jumlah gen dalam kromosom

Penalti₁ terjadi jika total biaya produksi pada sebuah kromosom nilainya lebih besar dari modal yang dialokasikan pada sekali masa tanam. Nilai penalti₁ adalah selisih total biaya produksi dan modal yang dimiliki. Sehingga, untuk mendapatkan nilai penalti₁ dapat dihitung dengan persamaan (3-3) berikut :

$$\delta_1 = \begin{cases} 0, & \text{Modal} > \text{Total biaya produksi} \\ \text{Total biaya produksi} - \text{modal}, & \text{Modal} < \text{Total biaya produksi} \end{cases} \dots (3-3)$$

Keterangan :

δ_1 = Penalti₁

Modal = Jumlah modal yang dialokasikan dalam proses penanaman

Total biaya produksi = Jumlah biaya produksi dalam sekali masa tanam

Nilai total biaya produksi pada sekali proses tanam merupakan penjumlahan dari setiap biaya produksi pada masing-masing tanaman yang akan ditanam. Sehingga, total biaya produksi dapat diperoleh dengan persamaan (3-4) berikut :

$$\text{Total Biaya Produksi} = L * \sum_{i=1}^n (X_i * \text{biaya produksi } [i]) \dots (3-4)$$

Penalti₂ terjadi ketika proporsi luas lahan yang direkomendasikan pada sebuah kromosom kurang dari lahan minimal yang telah ditentukan pakar. Nilai penalti₂ didapat dari selisih lahan minimal yang telah ditentukan pakar dengan proporsi pada masing-masing tanaman yang akan ditanam. Sehingga penalti₂ untuk setiap proporsi jenis tanaman dapat dihitung dengan persamaan (3-5) berikut :

$$\delta_2[i] = \begin{cases} 0, & \text{Proporsi } [i] > \text{luas minimal}[i] \\ \text{luas minimal } [i] - \text{Proporsi } [i], & \text{Proporsi } [i] < \text{luas minimal}[i] \end{cases} \dots (3-5)$$

Keterangan :

$\delta_2[i]$ = Penalti₂ tanaman ke-i

Luas minimal [i] = Luas minimal tanaman ke-i dari ketentuan pakar

Proporsi [i] = Proporsi luas tanaman ke-i dari rekomendasi sistem

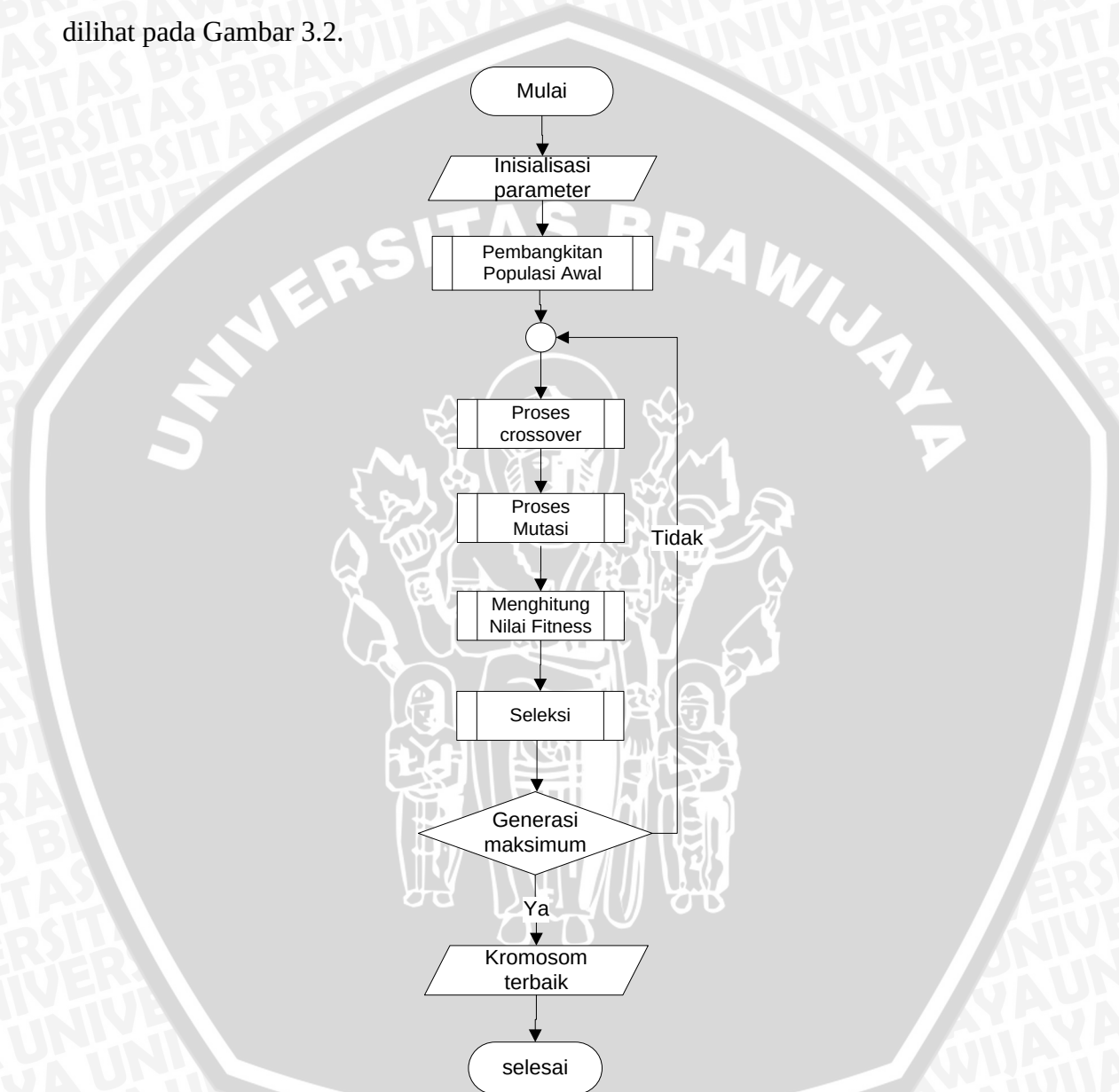
Dari hasil perhitungan penalti₂ untuk proporsi setiap jenis tanaman pada persamaan (3-5) kemudian dilakukan penjumlahan untuk setiap nilai penalti₂ pada masing-masing tanaman, sehingga didapat nilai penalti₂ seperti persamaan (3-6) berikut :

$$\text{Penalti}_2 = \sum_{i=1}^n \delta_2 \dots (3-6)$$

3.4 Siklus Penyelesaian Masalah Menggunakan Algoritma Genetika

Pada subbab ini dibahas proses pada algoritma genetika. Langkah pertama yaitu dengan membangkitkan populasi awal secara random. Proses reproduksi dilakukan dengan *crossover* dan mutasi. Selanjutnya individu dari populasi awal

dan *offspring* hasil *crossover* dan mutasi digabungkan untuk dilakukan proses seleksi. Seleksi dilakukan dengan menghitung *fitness* masing – masing individu. Individu terbaik adalah individu yang memiliki nilai *fitness* paling tinggi setelah *n* generasi. Diagram alir untuk proses algoritma genetika yang digunakan dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram Algoritma Genetika

Gambar 3.2 menggambarkan diagram alir algoritma genetika untuk penyelesaian masalah optimasi. Langkah pertama yaitu membuat inisialisasi parameter awal dengan menentukan jumlah individu dalam populasi (*popSize*), jumlah generasi yang akan dilakukan, nilai *crossover rate* (*cr*), dan nilai *mutation*

rate (mr). Langkah kedua yaitu membangkitkan populasi awal sebanyak individu yang telah diinisialisasikan. Langkah selanjutnya yaitu membuat populasi baru sebanyak jumlah generasi yang telah ditentukan dengan melakukan proses *crossover* pada kromosom induk yang telah dipilih sesuai dengan nilai *crossover rate* yang telah ditentukan sebelumnya. Kemudian melakukan proses mutasi pada kromosom induk yang dipilih berdasarkan nilai *mutation rate* yang telah ditentukan sebelumnya. Langkah selanjutnya menghitung nilai *fitness* pada masing – masing kromosom yang telah didapatkan. Langkah terakhir yaitu melakukan proses seleksi untuk menentukan individu yang akan dibangkitkan pada generasi selanjutnya. Jika terpenuhi kondisi berhenti, maka proses selesai.

Pada penyelesaian kasus ini, dimisalkan parameter yang digunakan yaitu :

- *population size* = 10
- *crossover rate* = 0.4
- *mutation rate* = 0.2

Dalam kasus ini, dimisalkan seorang petani memiliki luas lahan 14000M² dan modal awal Rp 10.000.000, hendak menanam 3 jenis tanaman yaitu jagung, kedelai dan kacang hijau. Data masing masing tanaman dapat dilihat pada Tabel 3.3

Tabel 3.3 Data Tanaman

Kode	Nama	Biaya Produksi (Rp,-/M ²)	Keuntungan (Rp,-/M ²)	Lahan Minimal (M ²)
T001	Jagung	926.10	764.00	1900
T003	Kedelai	1100.00	818.00	1200
T004	Kacang Hijau	600.00	527.00	600

Dari kasus di atas, dapat diselesaikan dengan algoritma genetika untuk penentuan proporsi luas lahan yang optimal pada setiap jenis tanaman yang akan ditanam dengan batasan modal yang dimiliki petani tersebut. Proses perhitungan manual penyelesaian algoritma genetika untuk mendapatkan proporsi luas lahan yang optimal pada masing masing jenis tanaman dapat dilihat pada sub bab 3.3.1 sampai 3.3.5 berikut :

3.3.1 Representasi Kromosom dan Perhitungan Fitness

Representasi kromosom dalam penelitian ini menggunakan representasi pengkodean real (*real coded*). Pembentukan kromosom yaitu dengan

membangkitkan nilai *random* pada *interval* 0 sampai 1, yang nantinya hasil nilai *random* tersebut menyatakan proporsi dari luas lahan. *Constraint* pembentukan kromosom adalah jumlah proporsi luas lahan dari index ke-1 hingga ke-n jika dijumlahkan tidak boleh lebih dari 1. Contoh representasi kromosom dari penelitian ini ditampilkan pada Gambar 3.3.

0.1106	0.2658	0.4684
--------	--------	--------

Gambar 3.3 Contoh Representasi Kromosom

Pada Gambar 3.3 menunjukkan kromosom yang memiliki 3 gen yang terdiri atas 3 jenis tanaman yang akan ditanam. Masing-masing kolom menyatakan proporsi luas lahan pada masing-masing jenis tanaman yang dipilih. Kolom ke-1 menyatakan tanaman jagung (T001) dengan proporsi 0.1106, kolom ke-2 menyatakan tanaman kedelai (T003) dengan proporsi 0.2658 dan kolom ke-3 menyatakan tanaman kacang hijau (T004) dengan proporsi 0.4684.

Sementara proses perhitungan *fitness* dari representasi kromosom pada Gambar 3.3 dapat dihitung dengan persamaan (3-1) sampai (3-6). Hasil perhitungan *fitness* ditunjukkan sebagai berikut:

$$\begin{aligned}
 \text{Total Biaya Produksi} &= L * \sum_{i=1}^n (X_i * \text{biaya produksi } [i]) \\
 &= 14000 * (0,1106 * 926,00 + 0,2658 * 1100,00 + 0,4684 * 600,00) \\
 &= 14000 * 785,82 \\
 &= 11001526,34
 \end{aligned}$$

Karena total biaya produksi dari hasil rekomendasi pada kromosom tersebut lebih besar daripada modal yang dimiliki yaitu 10.000.000, maka terjadi pelanggaran yang disebut penalti_1 . Sehingga dihitung nilai dari penalti_1 sesuai dengan persamaan (3-3). Berikut hasil perhitungan penalti_1 :

$$\begin{aligned}
 \text{Penalti}_1 &= \text{Total biaya produksi} - \text{Modal} \\
 &= 11001526,34 - 10.000.000 \\
 &= 1001526,34
 \end{aligned}$$

Selanjutnya menghitung proporsi dari masing-masing luas tanaman untuk kromosom pada Gambar 3.3, untuk mengetahui apakah terjadi pelanggaran

terhadap luas lahan minimal atau tidak. Berikut hasil perhitungan proporsi masing-masing tanaman.

$$\begin{aligned}\text{Proporsi jagung} &= L * X_{\text{jagung}} \\ &= 14000 * 0,1106 \\ &= 1548,07\end{aligned}$$

$$\begin{aligned}\text{Proporsi kedelai} &= L * X_{\text{kedelai}} \\ &= 14000 * 0,2658 \\ &= 15121,41\end{aligned}$$

$$\begin{aligned}\text{Proporsi kacang hijau} &= L * X_{\text{kacang hijau}} \\ &= 14000 * 0,4684 \\ &= 6557,18\end{aligned}$$

Dari hasil perhitungan proporsi T001, T003 dan T004, kemudian dibandingkan dengan luas minimal yang sudah menjadi ketentuan. Apabila luas proporsi kurang dari luas minimal, maka terjadi pelanggaran dan ada nilai penalti dengan perhitungan seperti pada persamaan (3-5). Dari hasil perhitungan, didapatkan nilai penalti₂ masing-masing jenis tanaman sebagai berikut :

$$\text{Penalti2}_{\text{jagung}} = 1900 - 1548,07 = 351,93 ; \text{ karena luas proporsi} < \text{luas minimal}$$

$$\text{Penalti2}_{\text{kedelai}} = 0 ; \text{ karena luas proporsi} > \text{luas minimal}$$

$$\text{Penalti2}_{\text{kacang hijau}} = 0 ; \text{ karena luas proporsi} > \text{luas minimal}$$

Setelah mendapat nilai penalti₂ untuk setiap jenis tanaman, selanjutnya dilakukan perhitungan dengan persamaan (3-6) untuk mendapatkan nilai total penalti₂. Berikut hasil perhitungan penalti₂ dengan persamaan (3-6):

$$\begin{aligned}\text{Penalti2} &= \sum_{i=1}^n \text{Penalti2} [i] \\ &= \text{Penalti2}_{\text{jagung}} + \text{Penalti2}_{\text{kedelai}} + \text{Penalti2}_{\text{kacang hijau}} \\ &= 351,93 + 0 + 0 \\ &= 351,93\end{aligned}$$

Sehingga nilai fitness untuk kromosom pada Gambar 3.3 adalah:

$$\begin{aligned}\text{Fitness} &= (L * \sum_{i=1}^n (X_i * \text{Keuntungan} [i])) - \alpha_1 * \text{Penalti1} - \alpha_2 * \text{Penalti2} \\ &= 14000 * (0,1106 * 764,00 + 0,3658 * 818,00 + 0,4684 * 527,00 - 5 * 1001526,34 - \\ &\quad 100 * 351,93)\end{aligned}$$

$$= 8827672.356 - 5007631.68 - 35193$$

$$= 3784847.655$$

3.3.2 Inisialisasi Populasi Awal

Inisialisasi populasi awal adalah proses membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu untuk mencari penyelesaian optimal. Populasi awal dibuat sebanyak sesuai jumlah *popSize* yang telah diinisialisasi di awal. Populasi awal yang dibangun pada penelitian ini menggunakan bilangan real dengan *range* yang telah ditentukan yaitu antara 0 sampai 1. Contoh populasi awal dapat dilihat pada Tabel 3.4.

Tabel 3.4 Contoh Populasi Awal

Parent	Kromosom		
	Jagung	Kedelai	Kacang Hijau
P1	0.1106	0.2658	0.4684
P2	0.0139	0.3610	0.1354
P3	0.1879	0.4366	0.3755
P4	0.5337	0.2216	0.2447
P5	0.2018	0.5353	0.0910
P6	0.2600	0.3025	0.4375
P7	0.5705	0.2710	0.1585
P8	0.1240	0.6169	0.0341
P9	0.2687	0.2515	0.4798
P10	0.6221	0.0472	0.2259

Nilai *fitness* dari representasi kromosom pada Tabel 3.4 dapat dihitung dengan cara yang sama pada sub bab 3.3.1. Hasil perhitungan *fitness* dari representasi kromosom Tabel 3.4 dapat dilihat pada Tabel 3.5.

Tabel 3.5 Nilai Fitness dari Populasi Awal

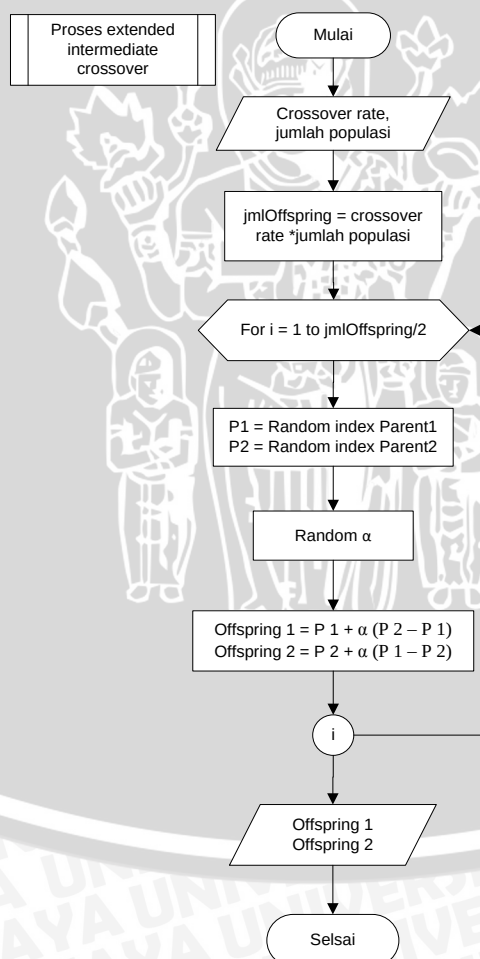
Parent	Kromosom			Penalti 1	Penalti 2			Total	Fitness
	Jagung	Kedelai	Kacang Hijau		Jagung	Kedelai	Kacang Hijau	Penalti 2	
Parent1	0.1106	0.2658	0.4684	1001526.34	351.93	0	0	351.93	3784847.65
Parent2	0.0139	0.3610	0.1354	0.00	1705.27	0	0	1705.27	5111231.89
Parent3	0.1879	0.4366	0.3755	2313975.44	0	0	0	0	-1789776.6
Parent4	0.5337	0.2216	0.2447	473204.18	0	0	365.76	365.76	5967365.44
Parent5	0.2018	0.5353	0.0910	1625119.56	0	0	0	0	835040.27
Parent6	0.2600	0.3025	0.4375	1704364.91	0	0	0	0	951173.24
Parent7	0.5705	0.2710	0.1585	2061425.20	0	0	0	0	-670045.52
Parent8	0.1900	0.5169	0.0641	1394089.80	163.95	0	123.09	287.04	1643205.74
Parent9	0.2687	0.2515	0.4798	1386927.08	0	0	0	0	2359357.69
Parent10	0.6221	0.0472	0.2259	10689527.15	0	539.539	0	539.54	5359064.042

3.3.3 Reproduksi

Proses reproduksi dilakukan untuk memperoleh variasi individu. Terdapat dua proses dalam melakukan reproduksi dalam algoritma genetika, yaitu proses *crossover* dan proses mutasi. Pada penelitian ini, proses *crossover* dilakukan dengan *extended intermediate crossover* sedangkan proses mutasi menggunakan *random mutation*, karena metode reproduksi ini yang paling umum digunakan pada algoritma genetika dengan representasi secara *realcode* (Mahmudy, 2009).

3.3.3.1 Perhitungan Crossover

Pada proses *crossover*, akan dihasilkan anakan (*offspring*) sebanyak hasil perkalian antara jumlah populasi dengan *crossover rate* (cr). Proses *crossover* ditunjukkan diagram alir pada Gambar 3.4.



Gambar 3.4 Diagram Alir Proses *Extended Intermediate Crossover*

Proses *crossover* yang digunakan dalam penelitian ini adalah *extended intermediate crossover*, karena pada proses *crossover* ini mampu membangkitkan

individu baru hasil persilangan antara dua induk dengan masih mewariskan keturunan dari gen induk. Dalam *extended intermediate crossover* menghasilkan *offspring* dengan variasi nilai gen dari kombinasi dua induk (Muhlenbein & Schlierkamp dalam Mahmudy, 2013).

Misalkan P_1 dan P_2 adalah dua kromosom parent, maka *offspring* C_1 dan C_2 dapat dibangkitkan dengan persamaan (2-1) dan (2-2). Banyaknya *offspring* yang dihasilkan dari kasus di atas adalah hasil perkalian *crossover rate* dengan jumlah populasi = $0,4 \times 10 = 4$ *offspring*. Maka dipilih 2 pasang *parent* untuk proses *crossover* yang akan menghasilkan 4 *offspring* (Mahmudy, 2013). Proses *extended intermediate exchange* ditunjukkan pada Gambar 3.5. Nilai α dibangkitkan pada interval $[0,1]$.

α	0.6289	0.7148	0.3165
Parent 1	0.1106	0.3658	0.4684
Parent 3	0.1879	0.4366	0.3755
Offspring 1	0.1592	0.4164	0.4390
Offspring 2	0.1393	0.3860	0.4049
Parent 4	0.5337	0.2216	0.0167
Parent 10	0.6221	0.0472	0.2259
Offspring 3	0.5893	0.0969	0.1462
Offspring 4	0.5665	0.1719	0.2964

Gambar 3.5 Hasil Perhitungan Proses *Crossover*

Pada proses *crossover* di atas, hasil *offspring* yang dihasilkan ada yang tidak sesuai dengan *constraint* yang ditetapkan. Jumlah seluruh isi gen pada beberapa kromosom *offspring* yang dihasilkan jumlahnya lebih dari satu. Untuk mengatasi hal ini, kromosom *offspring* yang mempunyai total isi gen lebih dari 1, maka harus *direpair* dengan persamaan (3-7) berikut :

$$\text{Repair Offspring} = \text{newGen} \frac{\text{isiGen}}{\sum_{i=1}^n \text{isiGen}} \dots\dots\dots(3-7)$$

Dari persamaan (3-6) hasil *repair offspring* dari proses *crossover* ditampilkan pada Tabel 3.6.

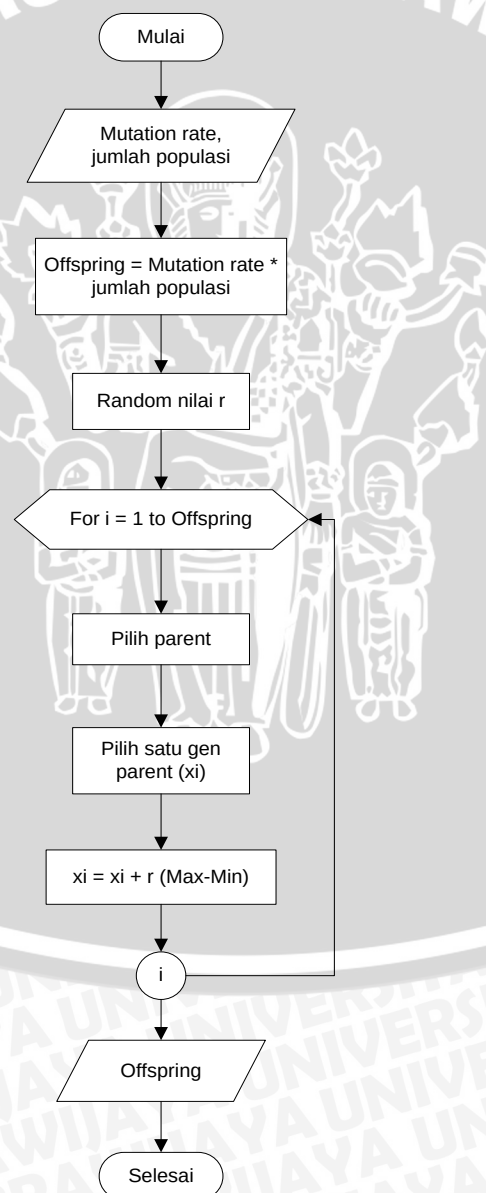
Tabel 3.6 Hasil *Repair Offspring Crossover*

Offspring 1	0.1569	0.4104	0.4327
Offspring 2	0.1393	0.3860	0.4049

<i>Offspring</i> 3	0.5893	0.0969	0.1462
<i>Offspring</i> 4	0.5665	0.1719	0.1597

3.3.3.2 Perhitungan Mutasi

Pada proses mutasi, seperti halnya pada proses *crossover*, akan dihasilkan *offspring*. Pada tiap kromosom yang gennya terpilih, akan dilakukan mutasi dengan menukarkan nilai gen tersebut secara acak. Jumlah *offspring* yang dihasilkan dari proses mutasi ini diperoleh dari hasil perkalian antara jumlah populasi dengan *mutation rate* (mr) yang telah ditentukan sebelumnya. Proses mutasi ditunjukkan oleh diagram alir pada Gambar 3.6.



Gambar 3.6 Diagram Alir Proses *Random Mutation*

Metode mutasi yang digunakan pada permasalahan ini adalah *random mutation*. Metode ini adalah metode mutasi yang sering digunakan untuk representasi kromosom secara *real code*. Pada *random mutation* ini hasil gen terpilih dibangkitkan dengan persamaan baru sehingga akan menghasilkan variasi *offspring* dengan kromosom baru yang belum ada pada inisialisasi populasi awal (Mahmudy, 2013).

Proses *random mutation* dengan memilih parent secara acak, kemudian melakukan proses mutasi terhadap nilai gen terpilih dengan persamaan (2-3) dengan membangkitkan nilai *r* (*random*) sebelumnya. Nilai *r* (*random*) dibangkitkan secara acak pada interval 0 sampai 1. Banyaknya *offspring* yang dihasilkan dari proses mutasi pada kasus di atas adalah hasil perkalian *mutation rate* dengan jumlah populasi = $0.2 \times 10 = 2$ *offspring*. Maka dipilih 2 *parent* secara random untuk proses mutasi. Nilai gen yang terpilih akan dibangkitkan sesuai dengan persamaan (2-3) (Mahmudy, 2013). Proses mutasi ditunjukkan pada Gambar 3.7. Nilai $r = 0,2414$; $\max = 0.8997$ dan $\min = 0.0547$.

Parent 3	0.1879	0.4366	0.3755
Offspring 5	0.1879	0.4366	0.5795
Parent 8	0.1240	0.6169	0.0341
Offspring 6	0.1240	0.8209	0.0341

Gambar 3.7 Hasil Perhitungan Proses Mutasi

Pada proses *mutasi* di atas, hasil *offspring* yang dihasilkan juga tidak sesuai dengan *constraint* yang ditetapkan. Jumlah seluruh isi gen pada beberapa *offspring* yang dihasilkan jumlahnya lebih dari satu. Sehingga perlu dilakukan proses *repair* kromosom hasil *offspring* dengan persamaan (3-6) agar hasil *offspring* dari proses mutasi menghasilkan kromosom yang sesuai dengan *constraint* yang ditetapkan.. Hasil *repair offspring* mutasi ditunjukkan pada Tabel 3.7.

Tabel 3.7 Hasil *Repair Offspring* Mutasi

Offspring 5	0.1560	0.3626	0.4813
Offspring 6	0.1240	0.2414	0.0341

3.3.4 Evaluasi

Evaluasi digunakan untuk menghitung nilai *fitness* dari *parent* dan seluruh *offspring* dari hasil proses *crossover* maupun mutasi untuk dilakukan proses seleksi pada tahap selanjutnya. Hasil evaluasi dapat dilihat pada Tabel 3.8.

Tabel 3.8 Hasil Evaluasi Individu

Individu	Kromosom			Fitness
	Jagung	Kedelai	Kacang Hijau	
Parent1	0.1106	0.3658	0.4684	3784847.655
Parent2	0.0139	0.3610	0.1354	5111231.886
Parent3	0.1879	0.4366	0.3755	-1789776.673
Parent4	0.5337	0.2216	0.0167	5967365.435
Parent5	0.2018	0.5353	0.0910	835040.2742
Parent6	0.2600	0.3025	0.4375	951173.2446
Parent7	0.5705	0.2710	0.0585	-670045.5205
Parent8	0.1240	0.6169	0.0341	1643205.736
Parent9	0.2687	0.2515	0.4798	2359357.689
Parent10	0.6221	0.0472	0.2259	5359064.042
Offspring1	0.1569	0.4104	0.4327	-375946.3567
Offspring2	0.1393	0.3860	0.4049	3141118.142
Offspring3	0.5893	0.0969	0.0829	8876488.712
Offspring4	0.5665	0.1719	0.1597	2540470.563
Offspring5	0.1560	0.3626	0.4813	1119340.389
Offspring6	0.1240	0.8209	0.0341	-11730197.54

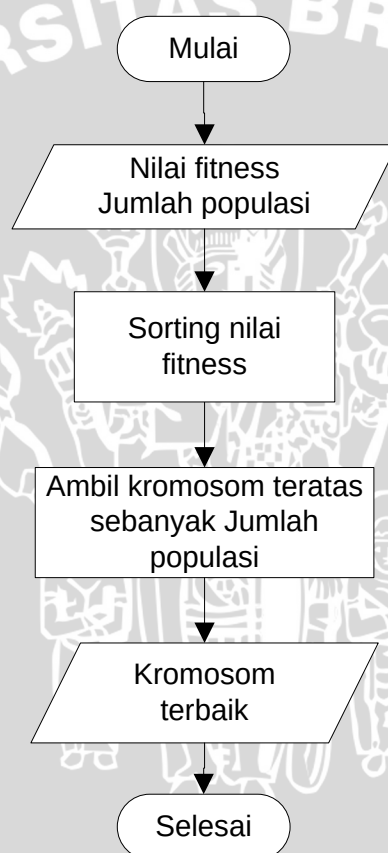
3.3.5 Seleksi

Proses seleksi dilakukan untuk memperoleh hasil kromosom terbaik yang akan dijadikan populasi pada generasi berikutnya. Proses seleksi pada penelitian ini menggunakan metode *elitism* dan *binary tournament* untuk dibandingkan hasil yang optimal dalam penyelesaian permasalahan optimasi penggunaan lahan pertanian ini. Metode seleksi *elitism* selalu memilih individu dengan nilai *fitness* tertinggi untuk menjadi individu pada generasi selanjutnya, namun pada metode seleksi *elitism* memungkinkan terjadinya individu yang homogen dan menyebabkan konvergensi dini (Putri, 2014). Metode seleksi *binary tournament* dilakukan dengan memilih dua induk dan dibandingkan nilai *fitness* nya. Nilai *fitness* yang lebih bagus akan menjadi individu pada generasi selanjutnya, sehingga dapat menjaga keragaman individu dalam suatu populasi (Putri, 2014). Oleh

karena itu, pada penelitian ini mencoba membandingkan kedua metode tersebut untuk mengetahui metode yang paling optimal pada penyelesaian permasalahan optimasi penggunaan lahan pertanian.

3.3.4.1 Seleksi Elitism

Tahapan seleksi *elitism* adalah mengurutkan kromosom berdasarkan nilai *fitness parent* dan *offspring* dari yang tertinggi hingga terendah kemudian mengambil kromosom sebanyak jumlah populasi awal. Diagram alir proses seleksi dengan metode *elitism* dapat dilihat pada Gambar 3.8.



Gambar 3.8 Seleksi *Elitism*

Tahapan dalam proses seleksi menggunakan metode *elitism* adalah sebagai berikut:

1. Mengurutkan semua kromosom berdasarkan nilai *fitness* tertinggi hingga terendah.
2. Mengambil kromosom dengan nilai *fitness* tertinggi sejumlah populasi awal.

Hasil seleksi *elitism* dapat dilihat pada Tabel 3.9.

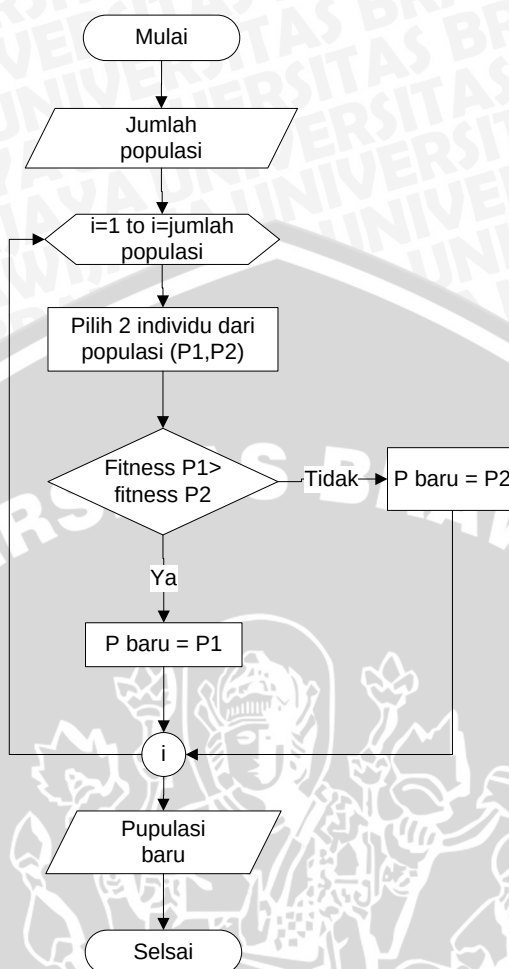
Tabel 3.9 Hasil Seleksi *Elitism*

Individu	Kromosom			Fitness
	Jagung	Kedelai	Kacang Hijau	
<i>Offspring 3</i>	0.589297	0.09693	0.082922674	8876488.712
<i>Parent 4</i>	0.533711	0.22162	0.016731499	5967365.435
<i>Parent 10</i>	0.622097	0.04718	0.225866333	5359064.042
<i>Parent 2</i>	0.013909	0.36101	0.135362811	5111231.886
<i>Parent 1</i>	0.110576	0.36582	0.468369885	3784847.655
<i>Offspring 2</i>	0.139258	0.38601	0.404907866	3141118.142
<i>Offspring 4</i>	0.566512	0.17187	0.159675158	2540470.563
<i>Parent 9</i>	0.268712	0.25145	0.479838216	2359357.689
<i>Parent 8</i>	0.124004	0.6169	0.034064785	1643205.736
<i>Offspring 5</i>	0.156031	0.36263	0.481340559	1119340.389

3.3.4.2 Seleksi Binary Tournament

Proses seleksi dengan menggunakan metode *binary tournament selection* adalah dengan cara mengambil secara acak dua individu yang ada pada populasi *parent* dan *offspring* kemudian dibandingkan nilainya. Satu individu dengan nilai *fitness* lebih besar akan terpilih untuk masuk populasi pada generasi berikutnya. Proses ini diulangi sampai terpenuhi sebanyak jumlah populasi yang telah ditentukan di awal (Mahmudy, 2013).

Proses seleksi dengan menggunakan metode *binary tournament selection* adalah dengan cara mengambil secara acak dua individu yang ada pada populasi *parent* dan *offspring* kemudian dibandingkan nilainya. Satu individu dengan nilai *fitness* lebih besar akan terpilih untuk masuk populasi pada generasi berikutnya. Diagram alir proses seleksi dengan metode *Binary tournament* dapat dilihat pada Gambar 3.9.



Gambar 3.9 Diagram Alir Seleksi *Binary tournament*

Hasil seleksi dengan menggunakan *binary tournament* dapat dilihat pada Tabel 3.10.

Tabel 3.10 Hasil Seleksi *Binary Tournament*

P(t+1)	Individu Pertama	Individu Kedua	Individu Terpilih	<i>Fitness</i>
Parent 1	P2	P4	P4	5967365.4353
Parent 2	P1	C3	C3	8876488.7116
Parent 3	P8	P5	P8	1643205.7358
Parent 4	C1	P4	P4	5967365.4353
Parent 5	P6	P10	P10	5359064.0425
Parent 6	P2	P3	P2	5111231.8860
Parent 7	P1	C4	P1	3784847.6550
Parent 8	P8	C1	P8	1643205.7358
Parent 9	C6	P5	P5	835040.2742
Parent 10	C3	C1	C3	8876488.7116

3.3.6 Memilih Kromosom Terbaik

Setelah dilakukan proses seleksi, maka dilanjutkan dengan pemilihan individu terbaik berdasarkan nilai *fitness* yang paling besar. Hasil individu terbaik pada generasi pertama dapat dilihat pada Tabel 3.11 dan detail kromosom terbaik dapat dilihat pada Tabel 3.12.

Tabel 3.11 Individu Terbaik

Individu Terbaik	Kromosom			Fitness
Offspring 3	0.5893	0.0969	0.1462	6684916.11

Tabel 3.12 Detail Kromosom Terbaik

Individu Terbaik	Proporsi (M ²)			Total Biaya Produksi (Rp)	Keuntungan (Rp)
	Jagung	Kedelai	Kacang Hijau		
Offspring 3	8250.1641	1356.9635	1160.9174	9829687.253	8024924.978



BAB IV

PERANCANGAN

4.1 Perancangan Database

Perancangan *database* merupakan bagian dari perancangan relasi *database* pada sistem yang akan dibuat. Pemodelan yang digunakan adalah struktur tabel *database*.

4.1.1 Struktur Tabel

Struktur tabel dari perancangan *database* ditunjukkan pada Tabel 4.1

Tabel 4.1 Stuktur Tabel Tanaman

Nama Atribut	Tipe	Keterangan
Kode	Varchar	Menyimpan kode taaman, bersifat primary key
Nama	Varchar	Menyimpan nama tanaman
Umur	Int	Menyimpan usia panen tanaman
Jarak_tanam	Varchar	Menyimpan jarak tanam tanaman
Biaya_produksi	Varchar	Menyimpan biaya produksi tanaman
Harga_Jual	Varchar	Menyimpan harga jual tanaman
Hasil_produksi	Varchar	Menyimpan hasil produksi tanaman
Keuntungan	Varchar	Menyimpan hasik keuntungan tanaman

4.2 Perancangan User Interface

Perancangan *User Interface* (UI) dari aplikasi ini terdiri dari empat halaman yaitu halaman utama (halaman yang berisi pilihan menu pada sistem), halaman *setup* data, halaman input data untuk perhitungan dan halaman hasil algoritma genetika.

4.2.1 Rancangan Halaman Utama

Rancangan tampilan halaman utama dari sistem yang akan dibuat ditampilkan pada Gambar 4.1.

MENU UTAMA		
OPTIMASI PENGGUNAAN LAHAN PERTANIAN DENGAN ALGORITMA GENETIKA		
Entry Data	Hitung Optimasi	Setup Data

Gambar 4.1 Rancangan Tampilan Halaman Utama

Keterangan :

1. Button untuk memasukkan data tanaman
2. Button untuk menghitung optimasi dan mendapatkan rekomendasi
3. Button untuk menu pencarian
4. Button untuk keluar dari sistem

4.2.2 Rancangan Halaman Setup Data

Rancangan untuk halaman setup data dari optimasi penggunaan luas lahan ini dapat dilihat pada Gambar 4.2.

HALAMAN SETUP DATA		
Kode_Tanaman	1	Biaya_Tanaman
Nama_Tanaman	2	Luas_Tanaman
Umur_Tanaman	3	Keuntungan
	4	5
	6	7
	8	9
Simpan	Reset	Keluar

Gambar 4.2 Rancangan Tampilan Halaman Setup Data

Keterangan :

1. TextBox untuk memasukkan kode tanaman
2. TextBox untuk memasukkan nama tanaman
3. TextBox untuk memasukkan umur tanaman
4. TextBox untuk memasukkan biaya produksi tanaman

5. TextBox untuk memasukkan luas minimal tanaman
6. TextBox untuk memasukkan keuntungan tanaman
7. Button untuk menyimpan data
8. Button untuk reset
9. Button untuk keluar dari halaman setup data

4.2.2 Rancangan Halaman Hitung Optimasi

Rancangan untuk halaman perhitungan dari optimasi penggunaan luas lahan ini dapat dilihat pada Gambar 4.3.

HALAMAN PERHITUNGAN OPTIMASI						
Luas Lahan	1	Meter Persegi	popSize	3		
Modal yang dimiliki	2	Rupiah	Generasi	4		
			Crossover rate	5		
			Mutation rate	6		
7						
Kode	Nama	Usia Panen	Jarak Tanam	Biaya Produksi	Luas Minimal	Pilih
						☐
						☐
						☐
						☐
						☐
						☐

8
Hitung

9
Reset

10
Keluar

Gambar 4.3 Tampilan Halaman Input Data Optimasi

Keterangan :

1. TextBox untuk memasukan total luas lahan yang dimiliki
2. TextBox untuk memasukan modal yang dimiliki
3. TextBox untuk memasukan PopSize
4. TextBox untuk memasukan jumlah generasi
5. TextBox untuk memasukan nilai *crossover rate*
6. TextBox untuk memasukan nilai *mutation rate*
7. DataGridView untuk menampilkan data tanaman yang akan dipilih
8. Button untuk melakukan proses algoritma genetika
9. Button untuk reset masukan
10. Button untuk keluar

4.2.3 Rancangan Halaman Hasil Proses Algoritma Genetika

Setelah input data dan di proses, hasil dari proses algoritma genetika untuk optimasi penggunaan lahan pertanian dapat dilihat pada Gambar 4.4.

HASIL PERHITUNGAN																																																																																												
1 Generasi Terakhir <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th>Individu</th> <th>Kromosom</th> <th>Fitness</th> </tr> </thead> <tbody> <tr><td>*</td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </tbody> </table> 2 Individu Terbaik <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th>Individu</th> <th>Kromosom</th> <th>Fitness</th> </tr> </thead> <tbody> <tr><td>*</td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </tbody> </table>						Individu	Kromosom	Fitness	*																																									Individu	Kromosom	Fitness	*																																							
	Individu	Kromosom	Fitness																																																																																									
*																																																																																												
	Individu	Kromosom	Fitness																																																																																									
*																																																																																												
3 Detail Hasil Rekomendasi <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th>Tanaman a</th> <th>Tanaman b</th> <th>Tanaman n</th> <th>Keuntungan</th> </tr> </thead> <tbody> <tr><td>*</td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> </tbody> </table> Ulangi 4 Keluar 5 Cetak 6						Tanaman a	Tanaman b	Tanaman n	Keuntungan	*																																																																																		
	Tanaman a	Tanaman b	Tanaman n	Keuntungan																																																																																								
*																																																																																												

Gambar 4.4 Tampilan Halaman Hasil Algoritma Genetika

Keterangan :

1. Datagridview hasil generasi akhir.
2. Datagridview hasil individu terbaik.
3. Datagridview detail rekomendasi luas lahan terbaik.
4. Button untuk mengulangi perhitungan
5. Button untuk keluar dari halaman
6. Button untuk mencetak hasil

4.3 Perancangan Uji Coba

Dikarenakan tidak adanya metode yang pasti untuk menentukan parameter algoritma genetika, maka untuk mengevaluasi program dilakukan beberapa uji coba antara lain :

1. Uji coba untuk menentukan ukuran populasi yang optimal.
2. Uji coba untuk menentukan banyaknya generasi yang optimal.
3. Uji coba untuk mencari kombinasi *crossover rate* dan *mutasi rate* yang terbaik.
4. Uji coba perbedaan hasil seleksi *elitism* dan *binary tournament*

4.3.1 Uji Coba Banyaknya Generasi

Uji coba banyaknya generasi yaitu uji coba untuk mengetahui banyaknya generasi yang tepat untuk menghasilkan komposisi luas tanaman yang terbaik. Ukuran banyaknya generasi mempengaruhi kemampuan algoritma genetika dalam menemukan solusi optimal. Banyaknya generasi yang digunakan adalah kelipatan 50. Rancangan uji coba ukuran populasi dapat dilihat pada Tabel 4.2. Parameter yang digunakan pada uji coba populasi adalah :

- a. Ukuran populasi = 50
- b. Ukuran generasi = 50-550
- c. *Crossover rate* dan *mutation rate* = 0,5 dan 0,5

Tabel 4.2 Rancangan Uji Coba Banyaknya Generasi

Banyak Generasi	Nilai <i>Fitness</i>										Rata- rata <i>Fitness</i>
	Percobaan Generasi Ke-										
	1	2	3	4	5	6	7	8	9	10	
50											
100											
150											
200											
250											
300											
350											
400											
450											
500											
550											

4.3.2 Uji Coba Ukuran Populasi

Uji coba ukuran populasi yaitu untuk mengetahui ukuran populasi yang tepat untuk menghasilkan solusi yang terbaik. Ukuran populasi mempengaruhi kemampuan algoritma genetika dalam menemukan solusi optimal. Banyaknya populasi yang digunakan adalah kelipatan 25. Rancangan uji coba ukuran populasi dapat dilihat pada Tabel 4.3. Parameter yang digunakan pada uji coba populasi adalah :

- a. Ukuran populasi = 25-250
- b. Ukuran generasi = generasi terbaik pada uji coba generasi

c. *Crossover rate* = 0.5

d. *Mutation rate* = 0.5

Tabel 4.3 Rancangan Uji Coba Ukuran Populasi

Banyak Populasi	Nilai <i>fitness</i>										Rata-rata <i>fitness</i>
	Percobaan populasi ke -										
	1	2	3	4	5	6	7	8	9	10	
25											
50											
75											
100											
125											
150											
175											
200											
225											
250											

4.3.2 Uji Coba Kombinasi *Crossover Rate* dan Mutasi *Rate*

Uji coba berdasarkan *crossover rate* (*cr*) dan mutasi *rate* (*mr*) dilakukan untuk mengetahui kombinasi *cr* dan *mr* yang paling baik untuk menghasilkan solusi yang optimal. Pada uji coba ini digunakan nilai yang berbeda pada parameter algoritma genetika, yaitu nilai *crossover rate* (*cr*) dan mutasi *rate* (*mr*). Nilai *cr* dan *mr* yang digunakan antara 0 dan 1. Rancangan uji coba kombinasi *cr* dan *mr* ditampilkan pada Tabel 4.4. Parameter yang digunakan dalam uji coba adalah :

- Jumlah populasi = Hasil populasi terbaik pada uji coba populasi
- Jumlah generasi = Hasil generasi terbaik pada uji coba generasi

Tabel 4.4 Rancangan Uji Coba Kombinasi *cr* dan *mr*

Kombinasi		Nilai <i>Fitness</i>										Rata-rata <i>fitness</i>
		Percobaan kombinasi <i>cr</i> dan <i>mr</i> ke -										
<i>cr</i>	<i>mr</i>	1	2	3	4	5	6	7	8	9	10	
1	0											
0,9	0,1											
0,8	0,2											
0,7	0,3											
0,6	0,4											
0,5	0,5											
0,4	0,6											
0,3	0,7											

0,2	0,8											
0,1	0,9											
0	1											

4.3.2 Uji Coba Seleksi

Pengujian seleksi dilakukan untuk membandingkan metode seleksi mana yang terbaik pada permasalahan optimasi penggunaan lahan pertanian ini. Pengujian setiap metode seleksi dilakukan sebanyak 10 kali dan dibandingkan setiap hasilnya. Rancangan uji coba perbedaan seleksi dapat dilihat pada Tabel 4.5. Parameter yang digunakan dalam pengujian seleksi adalah parameter terbaik pada pengujian generasi, populasi dan kombinasi *crossover rate* serta *mutation rate*.

Tabel 4.5 Uji Coba Perbandingan Seleksi

Metode Seleksi	Nilai fitness percobaan ke-										Rata- rata Fitness
	1	2	3	4	5	6	7	8	9	10	
Elitism											
Binary Tournament											

BAB V

IMPLEMENTASI

5.1 Implementasi Program

Berdasarkan metodologi dan perancangan yang telah dijelaskan pada Bab 3 dan 4, maka selanjutnya akan dijelaskan implementasi proses-proses tersebut dalam bentuk potongan kode program.

5.1.1 Pembangkitan Populasi Awal

Proses pembangkitan populasi awal dilakukan dengan melakukan inisialisasi kromosom berdasarkan hasil pembangkitan jumlah populasi yang telah dimasukan oleh pengguna. Suatu individu diperoleh dengan cara membangkitkan bilangan *random* pada interval 0 sampai 1 yang menyatakan proporsi dari luas tanaman yang akan ditanam. *Constraint* pembentukan kromosom adalah jumlah proporsi luas lahan dari index ke-1 hingga ke-n, dimana jika dijumlahkan tidak boleh lebih dari satu. Proses *repair* kromosom dilakukan apabila dari kromosom yang telah dibangkitkan secara *random* melanggar dari *constrain* yang telah ditentukan. Proses pembangkitan populasi awal dapat dilihat pada *Source Code* 5.1.

```

For i As Integer = 0 To txtPopsSize.Value - 1
    Dim iTotal As Double = 0
    For j As Integer = 0 To iCheckedCount - 1
        iPopulation(i, j) = Math.Round(randomNumber.Next(0,
        10000) / 10000, 4)
        iTotal += iPopulation(i, j)
        If iPopulationMin > iPopulation(i, j) Then iPopulationMin
        = iPopulation(i, j)
        If iPopulationMax < iPopulation(i, j) Then iPopulationMax
        = iPopulation(i, j)
    Next
    If iTotal > 1 Then
        For j As Integer = 0 To iCheckedCount - 1
            iPopulation(i, j) = Math.Round(iPopulation(i, j) /
            iTotal, 4)
        Next
    End If
Next

```

Source Code 5.1 Pembangkitan Populasi Awal

5.1.2 Perhitungan Nilai *Fitness*

Perhitungan nilai *fitness* diperoleh dengan menghitung nilai keuntungan dari kromosom dikurangi penalti₁ dan penalti₂. Nilai keuntungan dari kromosom didapat dari mengalikan proporsi luas lahan dengan keuntungan pada tanaman per satuan luas. Penalti₁ terjadi ketika biaya produksi lebih besar dari modal yang dimiliki. Nilai penalti₁ di dapat dari hasil pengurangan biaya produksi dan modal. Sementara penalti₂ terjadi ketika proporsi lahan kurang dari lahan minimal yang telah ditentukan dari pakar. Nilai penalti₂ adalah jumlah dari penalti₂ pada setiap tanaman yang akan ditanam. Perhitungan *fitness* dilakukan pada setiap individu. Proses perhitungan *fitness* dapat dilihat pada *source code* 5.2.

```
//Hitung Total Biaya Produksi
Dim iTotalBiayaProduksi As Double = 0
For j As Integer = 0 To iCheckedCount - 1
    iTotalBiayaProduksi += iPopulation(i, j) *
        dbDataTable.Rows(sTanamanIndex(j))(6)
Next

iTotalBiayaProduksi = txtLuasLahan.Value * iTotalBiayaProduksi

//Hitung Penalti1
Dim iPenalti1 As Double = 0
If iTotalBiayaProduksi > txtModal.Value Then
    iPenalti1 = iTotalBiayaProduksi - txtModal.Value
End If

//Hitung Penalti2
Dim iPenalti2 As Double = 0
For j As Integer = 0 To iCheckedCount - 1
    If dbDataTable.Rows(sTanamanIndex(j))(7) >
        txtLuasLahan.Value * iPopulation(i, j) Then
        iPenalti2 += dbDataTable.Rows(sTanamanIndex(j))(7)
            - txtLuasLahan.Value * iPopulation(i, j)
    End If
Next

//Hitung Keuntungan
Dim iKeuntungan As Double = 0
For j As Integer = 0 To iCheckedCount - 1
    iKeuntungan += iPopulation(i, j) *
        dbDataTable.Rows(sTanamanIndex(j))(10)
Next
```

```
//Hitung Fitness
iPopulation(i, iCheckedCount) = Math.Round(txtLuasLahan.Value
* iKeuntungan - alpha1 * iPenalti1 - alpha2 * iPenalti2, 4)
```

Source Code 5.2 Perhitungan Nilai Fitness

5.1.3 Proses Crossover

Metode crossover yang digunakan adalah *extended intermediate crossover*. Langkah pertama yang dilakukan adalah pemilihan induk yang akan mengalami proses *crossover* dengan melakukan pemilihan secara acak dari sejumlah kromosom yang telah dibangkitkan pada proses inisialisasi populasi awal. Selanjutnya membangkitkan nilai alfa (α) secara acak antara 0 sampai 1. Individu yang terpilih yang akan dilakukan proses *crossover*. Anak dari proses *crossover* sejumlah perkalian dari jumlah individu dan nilai *crossover rate*. Proses *crossover* dapat dilihat pada *source code 5.3*.

```
For i As Integer = 0 To Math.Round(iPopulasiCrossover / 2) - 1
    Dim iParentIndex1 As Integer = 0
    Dim iParentIndex2 As Integer = 0
    Dim iRandomAlpha As Double = 0
    Dim iTotalOffspring1 As Double = 0
    Dim iTotalOffspring2 As Double = 0

    Do While iParentIndex1 = iParentIndex2
        iParentIndex1 = randomNumber.Next(0,txtPopsiize.Value-1)
        iParentIndex2 = randomNumber.Next(0,txtPopsiize.Value-1)
        iRandomAlpha = Math.Round(randomNumber.Next(0,
        10000)/10000,4)
    Loop

    For j As Integer = 0 To iCheckedCount - 1
        iPopulation(txtPopsiize.Value - 1 + (i * 2 + 1), j) =
        iPopulation(iParentIndex1, j) + iRandomAlpha *
        Math.Abs(iPopulation(iParentIndex2, j) -
        iPopulation(iParentIndex1, j))

        iPopulation(txtPopsiize.Value - 1 + (i * 2 + 2), j) =
        iPopulation(iParentIndex2, j) + iRandomAlpha *
        Math.Abs(iPopulation(iParentIndex1, j) -
        iPopulation(iParentIndex2, j))
    Next
Next
```

Source Code 5.3 Proses Crossover

Proses *repair* gen dilakukan apabila kromosom hasil reproduksi *crossover* melanggar *constrain* yang telah ditentukan, yaitu nilai dari gen pada kromosom jika dijumlahkan lebih dari satu. Apabila hasil penjumlahan gen pada setiap kromosom lebih dari satu, dilakukan proses *repair* dengan cara membagi setiap gen dengan jumlah dari keseluruhan nilai gen pada kromosom tersebut. *Source code* proses *repair* gen pada kromosom hasil *crossover* dapat dilihat pada *source code* 5.4.

```
//Repair offspring crossover
For j As Integer = 0 To iCheckedCount - 1
    iTotalOffspring1 += iPopulation(txtPopsSize.Value - 1 + (i
    * 2 + 1), j)
    iTotalOffspring2 += iPopulation(txtPopsSize.Value - 1 + (i
    * 2 + 2), j)
Next
If iTotalOffspring1 > 1 Then
    For j As Integer = 0 To iCheckedCount - 1
        iPopulation(txtPopsSize.Value - 1 + (i * 2 + 1), j) =
        Math.Round(iPopulation(txtPopsSize.Value - 1 + (i * 2 +
        1), j) / iTotalOffspring1, 4)
    Next
End If

If iTotalOffspring2 > 1 Then
    For j As Integer = 0 To iCheckedCount - 1
        iPopulation(txtPopsSize.Value - 1 + (i * 2 + 2), j) =
        Math.Round(iPopulation(txtPopsSize.Value - 1 + (i * 2 +
        2), j) / iTotalOffspring2, 4)
    Next
End If
```

Source Code 5.4 Proses Repair Offspring crossover

5.1.4 Proses Mutasi

Metode mutasi yang digunakan adalah *random mutation*. Hal pertama yang dilakukan adalah membangkitkan nilai *random* (r) dengan interval 0 sampai 1. Langkah selanjutnya yaitu melakukan pemilihan individu yang akan mengalami proses mutasi dengan cara memilih secara acak dari individu yang telah dibangkitkan pada proses inisialisasi populasi awal. Individu yang terpilih, akan mengalami proses mutasi. Hasil anak dari proses mutasi merupakan perkalian dari jumlah populasi dikali dengan nilai *mutation rate*. *Source code* proses mutasi dapat dilihat pada *source code* 5.5.

```
For i As Integer = 0 To iPopulasiMutasi - 1
    Dim iParentIndex As Integer = 0
    Dim iParentColumnIndex As Integer = 0
```

```

Dim iRandomAlpha As Double = 0
Dim iTotalOffspring As Double = 0

//random xi
iParentIndex = randomNumber.Next(0, txtPopsize.Value - 1)
iParentColumnIndex = randomNumber.Next(0, iCheckedCount - 1)
iRandomAlpha = Math.Round(randomNumber.Next(0, 10000) / 10000, 4)

//random mutation
For j As Integer = 0 To iCheckedCount - 1
    If j = iParentColumnIndex Then
        iPopulation(txtPopsize.Value - 1 + Math.Round(iPopulasiCrossover / 2) * 2 + (i + 1), j) = iPopulation(iParentIndex, j) + iRandomAlpha * (iPopulationMax - iPopulationMin)
    Else
        iPopulation(txtPopsize.Value - 1 + Math.Round(iPopulasiCrossover / 2) * 2 + (i + 1), j) = iPopulation(iParentIndex, j)
    End If
Next
Next

```

Source Code 5.5 Proses Mutasi

Apabila kromosom hasil proses mutasi ada yang melanggar *constrain* yang telah ditentukan, maka dilakukan *repair* gen. Proses *repair* gen dilakukan dengan cara membagi setiap gen dengan jumlah dari keseluruhan gen pada kromosom tersebut. *Source code* proses *repair* gen pada kromosom hasil mutasi dapat dilihat pada source code 5.6.

```

//total gen pda kromosom
For j As Integer = 0 To iCheckedCount - 1
    iTotalOffspring += iPopulation(txtPopsize.Value - 1 + Math.Round(iPopulasiCrossover / 2) * 2 + (i + 1), j)
Next

//Repair Offspring
If iTotalOffspring > 1 Then
    For j As Integer = 0 To iCheckedCount - 1
        iPopulation(txtPopsize.Value - 1 + Math.Round(iPopulasiCrossover / 2) * 2 + (i + 1), j) = Math.Round(iPopulation(txtPopsize.Value - 1 +

```

```

Math.Round(iPopulasiCrossover / 2) * 2 + (i + 1), j)
/ iTotalOffspring, 4)
Next
End If

```

Source Code 5.6 Proses Repair Offspring Mutasi

5.1.5 Proses Seleksi Elitism

Proses seleksi pemilihan induk untuk generasi berikutnya dilakukan dengan metode *elitism* dan *binary tournament*. Pada seleksi model *elitism* ini, semakin besar nilai *fitness* suatu individu, maka akan semakin besar kemungkinan individu tersebut terpilih untuk menjadi induk. Langkah pertama proses seleksi ini adalah dengan mengurutkan nilai *fitness* setiap individu dari nilai yang terbesar sampai terkecil, lalu diambil individu teratas sebanyak jumlah populasi awal untuk menjadi induk pada generasi berikutnya. *Source code* proses seleksi *elitism* dapat dilihat pada *source code* 5.7.

```

Dim dv As DataView = dt.DefaultView
dv.Sort = "Fitness DESC"
dt = dv.ToTable

ReDim iPopulation(txtPopsiize.Value - 1 +
Math.Round(iPopulasiCrossover / 2) * 2 + iPopulasiMutasi,
iCheckedCount)

//Inisialisasi Populasi untuk perhitungan berikutnya
For i As Integer = 0 To txtPopsiize.Value - 1
    For j As Integer = 0 To iCheckedCount
        iPopulation(i, j) = dt.Rows(i)(j)
    Next
Next
Next

```

Source Code 5.7 Proses Seleksi Elitism

5.1.6 Proses Seleksi Binary Tournament

Proses seleksi *binary tournament*, dilakukan dengan cara memilih secara acak dua individu. Individu yang terpilih, dibandingkan nilai *fitness* nya. Nilai *fitness* yang lebih besar akan terpilih menjadi induk untuk generasi berikutnya. Hal ini dilakukan sampai mendapatkan sejumlah populasi awal untuk menjadi induk pada generasi berikutnya. *Source code* proses seleksi *binary tournament* dapat dilihat pada *source code* 5.8.


```

ReDim iPopulation(txtPopsSize.Value - 1 +
Math.Round(iPopulasiCrossover / 2) * 2 + iPopulasiMutasi,
iCheckedCount)
//Inisialisasi Populasi untuk perhitungan berikutnya
For i As Integer = 0 To txtPopsSize.Value - 1
    Dim iParentIndex1 As Integer = 0
    Dim iParentIndex2 As Integer = 0

    Do While iParentIndex1 = iParentIndex2
        iParentIndex1 = randomNumber.Next(0,
UBound(iPopulation))
        iParentIndex2 = randomNumber.Next(0,
UBound(iPopulation))
    Loop
    For j As Integer = 0 To iCheckedCount
        If dt.Rows(iParentIndex1)(iCheckedCount) >
dt.Rows(iParentIndex2)(iCheckedCount) Then
            iPopulation(i, j) = dt.Rows(iParentIndex1)(j)
        Else
            iPopulation(i, j) = dt.Rows(iParentIndex2)(j)
        End If
    Next
Next

```

Source Code 5.8 Proses Seleksi Binary Tournament

5.1.7 Pemilihan Kromosom Terbaik

Kromosom terbaik adalah kromosom yang memiliki nilai *fitness* tertinggi dari semua kromosom. Kromosom terbaik dipilih dari semua kromosom hasil proses algoritma genetika dengan membandingkan nilai *fitness*. Kemudian dilakukan proses perhitungan proporsi luas lahan masing-masing tanaman yang dihasilkan dari kromosom terbaik dan dihitung analisa biaya produksi dan keuntungannya. Kromosom terbaik adalah yang akan menjadi solusi dari optimasi penggunaan luas lahan pertanian. Proses pemilihan kromosom terbaik dan hasil yang ditampilkan ke pengguna dapat dilihat pada source code 5.9.

```

//mengambil kromosom teratas setelah di sorting
For j As Integer = 0 To iCheckedCount
    iPopulation(0, j) = dt.Rows(0)(j)
Next

//Hitung Total Biaya Produksi Terbaik
Dim iTotalBiayaProduksiTerbaik As Double = 0
For j As Integer = 0 To iCheckedCount - 1
    iTotalBiayaProduksiTerbaik += iPopulation(0, j) *
txtLuasLahan.Value * dbDataTable.Rows(sTanamanIndex(j))(6)

```

```

Next
    //Hitung Keuntungan Terbaik
    Dim iKeuntunganTerbaik As Double = 0
    For j As Integer = 0 To iCheckedCount - 1
        iKeuntunganTerbaik += iPopulation(0, j) *
            txtLuasLahan.Value *
            dbDataTable.Rows(sTanamanIndex(j)) (10)
    Next

    //Hitung Total luas lahan yang dipakai
    Dim iTotalluasPakai As Double = 0
    For j As Integer = 0 To iCheckedCount - 1
        iTotalluasPakai += iPopulation(0, j) *
            txtLuasLahan.Value
    Next

    //Kesimpulan
    If iTotalBiayaProduksiTerbaik > txtModal.Value Then
        lblResult.Text = "Maaf, modal Anda tidak cukup untuk
            melakukan penanaman seperti ini. " & vbCrLf &
            "Kekurangan modal adalah " &
            Format$(iTotalBiayaProduksiTerbaik - txtModal.Value,
                "n0") & vbCrLf &
            "Sisa luas lahan adalah " & Format$(txtLuasLahan.Value -
                iTotalluasPakai + 1, "n0")
    Else
        lblResult.Text = "Modal Anda sudah cukup untuk melakukan
            penanaman seperti ini. " & vbCrLf & "Sisa modal Anda
            adalah " & Format$(txtModal.Value -
                iTotalBiayaProduksiTerbaik, "n0") & vbCrLf &
            "Sisa luas lahan adalah " & Format$(txtLuasLahan.Value -
                iTotalluasPakai + 1, "n0")
    End If

```

Source Code 5.9 Proses Pemilihan Kromosom Terbaik

5.2 Implementasi Antarmuka Pengguna

Antarmuka pengguna pada implementasi ini terdiri dari *form* untuk halaman utama, dimana pada halaman utama ini pengguna dapat memilih menu yang ada yaitu menu *entry* data tanaman, menu perhitungan algoritma genetika dan menu daftar data tanaman.

5.2.1 Implementasi Halaman Utama

Halaman utama adalah halaman yang pertama kali muncul saat program dijalankan. Pada halaman utama ini, terdapat tiga sub menu yang dapat dipilih pengguna, yaitu menu *entry* data tanaman, perhitungan algoritma genetika dan daftar data tanaman. Gambar 5.1 merupakan implementasi halaman utama sistem.



Gambar 5.1 Implementasi Halaman Utama

5.2.2 Implementasi Halaman *Entry Data Tanaman*

Halaman *entry* data tanaman adalah halaman yang dapat digunakan oleh pengguna untuk memasukkan data tanaman yang baru. Ketika button simpan di pilih, maka data yang dimasukan, akan disimpan pada *database* sistem. Ketika button tutup dipilih, pengguna akan keluar dari halaman *entry* data. Gambar 5.2 merupakan implementasi dari halaman *entry* data.

Gambar 5.2 Implementasi Halaman *Entry Data Tanaman*

5.2.3 Halaman Proses Algoritma Genetika

Halaman proses algoritma genetika adalah halaman yang digunakan untuk melakukan proses dan menampilkan hasil optimasi dengan algoritma genetika. Pada halaman ini, pengguna dapat memilih tanaman yang akan ditanam, memasukkan luas lahan dan modal, serta parameter algoritma genetika untuk

selanjutnya dilakukan proses perhitungan dengan algoritma genetika. Gambar 5.3 merupakan implementasi dari halaman proses algoritma genetika.

The screenshot shows the 'Proses Algoritma Genetika' application window. It contains a table of plant data, input fields for land area, budget, and population, and a table of the best chromosome.

	ID Tanaman	Nama	Jarak Tanam (cm)	Usia Panen (hari)	Biaya Produksi (Rp/M2)	Lahan Minimal (M2)	Hasil Produksi (Kg/M2)	Harga Jual (Rp/Kg)	Keuntungan (Rp/M2)
☒	T001	JAGUNG	40 X 90	85 - 95	1926.00	160.33	0.85	3200.00	784.40
☐	T002	KACANG TANAH	30 X 20	100 - 110	531.00	216.00	1.92	7000.00	4909.00
☒	T003	KEDELAI	40 X 40	85 - 90	910.00	280.00	0.24	7200.00	818.00
☒	T004	KACANG HIJAU	20 X 20	58 - 85	718.00	160.00	0.10	11000.00	327.00
☐	T005	UBI	40 X 90	120 - 130	114.00	360.00	3.59	2450.00	4688.85
☒	T006	SINGKONG	100 X 100	210 - 215	365.00	1000.00	5.50	700.00	3485.00
☐	T007	BAWANG MERAH	20 X 20	95 - 115	3324.00	300.00	0.72	18000.00	9672.83

Luas Lahan: 10,000.00 (Meter2)
 Modal: 5,000,000.00 (Rupiah)
 Popsize: 100
 Probabilitas Crossover: 0.40
 Probabilitas Mutasi: 0.20
 Iterasi: 10
 Seleksi Kromosom: ☒ Elitism ☐ Binary Tournament

Buttons: Hitung, Reset Grid, Tutup

	JAGUNG	KEDELAI	KACANG HIJAU	SINGKONG	Fitness
0.0359	0.1812	0.1851	0.3288	13827937.3996	
0.013	0.1586	0.1402	0.3116	12686704	
0.0135	0.1695	0.2807	0.3405	8023680.4149	
0.0136	0.1705	0.2938	0.3432	-3197097.1469	
0.0379	0.1904	0.1857	0.3856	-4362186.3547	
0.0356	0.191	0.2171	0.3538	-12452759.0452	
0.0389	0.1922	0.231	0.3966	-43984403.2889	
0.013	0.1586	0.3917	0.3116	-50914874.3329	
0.0523	0.0228	0.3244	0.6006	-51079556.8	

Kromosom Terbaik

	JAGUNG	KEDELAI	KACANG HIJAU	SINGKONG	Fitness
0.0359	0.1812	0.1851	0.3288	13827937.3996	

Hasil Optimasi

	JAGUNG	KEDELAI	KACANG HIJAU	SINGKONG	Fitness
359	1812	1851	3288	13827937.3996	

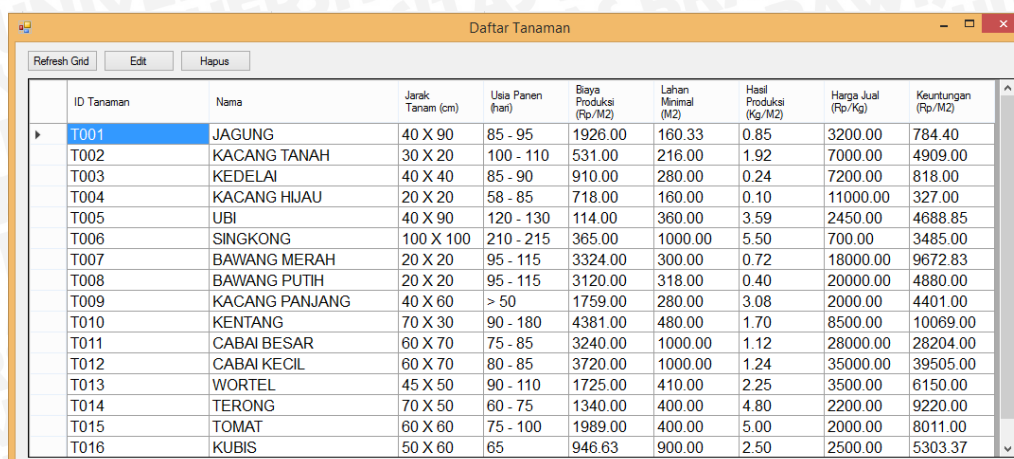
Modal Anda sudah cukup untuk melakukan penanaman seperti ini.
 Sisa modal Anda adalah 130,508
 Sisa luas lahan adalah 2,691

Gambar 5.3 Implementasi Halaman Proses Algoritma Genetika

Pada Gambar 5.3, ketika button hitung dipilih maka sistem akan melakukan proses dengan algoritma genetika untuk mendapatkan hasil optimasi. Ketika button reset gird dipilih, maka masukan yang sudah dimasukan pengguna akan direset ulang. Ketika button tutup dipilih, maka pengguna akan keluar dari halaman perhitungan algoritma genetika. Hasil yang ditampilkan kepada pengguna adalah hasil dari generasi terakhir dari proses algoritma genetika, hasil kromosom terbaik dan informasi hasil optimasi luas dari masing-masing tanaman serta total biaya produksi dan keuntungan yang didapat.

5.2.4 Implementasi Halaman Daftar Data Tanaman

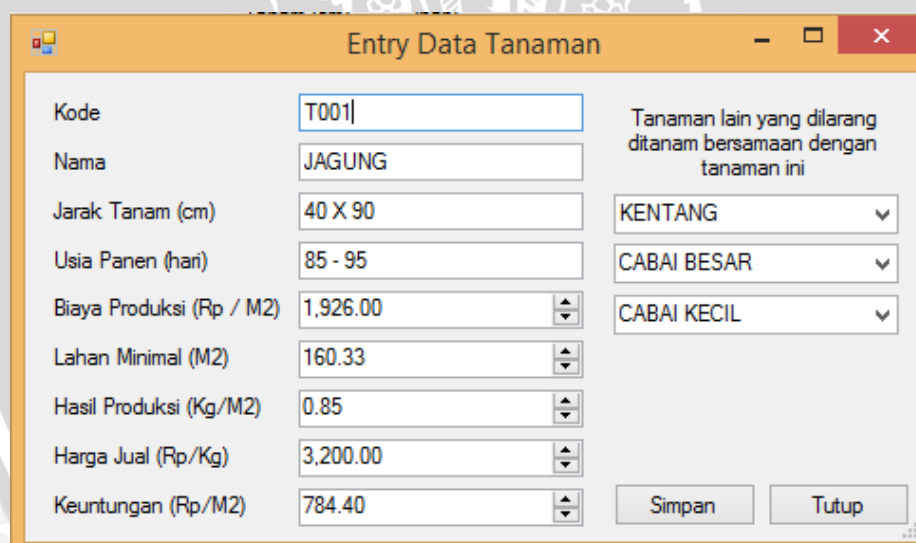
Halaman daftar data tanaman adalah halaman yang dapat digunakan untuk melihat daftar detail data tanaman yang sudah tersimpan pada *database* sistem. Pengguna juga dapat melakukan proses *edit* data atau hapus data. Gambar 5.4 merupakan implementasi halaman daftar data tananaman.



ID Tanaman	Nama	Jarak Tanam (cm)	Usia Panen (hari)	Biaya Produksi (Rp/M2)	Lahan Minimal (M2)	Hasil Produksi (Kg/M2)	Harga Jual (Rp/Kg)	Keuntungan (Rp/M2)
T001	JAGUNG	40 X 90	85 - 95	1926.00	160.33	0.85	3200.00	784.40
T002	KACANG TANAH	30 X 20	100 - 110	531.00	216.00	1.92	7000.00	4909.00
T003	KEDELAI	40 X 40	85 - 90	910.00	280.00	0.24	7200.00	818.00
T004	KACANG HIJAU	20 X 20	58 - 85	718.00	160.00	0.10	11000.00	327.00
T005	UBI	40 X 90	120 - 130	114.00	360.00	3.59	2450.00	4688.85
T006	SINGKONG	100 X 100	210 - 215	365.00	1000.00	5.50	700.00	3485.00
T007	BAWANG MERAH	20 X 20	95 - 115	3324.00	300.00	0.72	18000.00	9672.83
T008	BAWANG PUTIH	20 X 20	95 - 115	3120.00	318.00	0.40	20000.00	4880.00
T009	KACANG PANJANG	40 X 60	> 50	1759.00	280.00	3.08	2000.00	4401.00
T010	KENTANG	70 X 30	90 - 180	4381.00	480.00	1.70	8500.00	10069.00
T011	CABAI BESAR	60 X 70	75 - 85	3240.00	1000.00	1.12	28000.00	28204.00
T012	CABAI KECIL	60 X 70	80 - 85	3720.00	1000.00	1.24	35000.00	39505.00
T013	WORTEL	45 X 50	90 - 110	1725.00	410.00	2.25	3500.00	6150.00
T014	TERONG	70 X 50	60 - 75	1340.00	400.00	4.80	2200.00	9220.00
T015	TOMAT	60 X 60	75 - 100	1989.00	400.00	5.00	2000.00	8011.00
T016	KUBIS	50 X 60	65	946.63	900.00	2.50	2500.00	5303.37

Gambar 5.4 Implementasi Halaman Daftar Data Tanaman

Pada Gambar 5.4, ketika pengguna memilih button hapus, maka data yang terpilih akan terhapus dari database sistem. Ketika pengguna memilih sub menu *edit*, maka akan muncul *form* untuk melakukan proses *edit* data seperti pada Gambar 5.5.



Kode	T001	Tanaman lain yang dilarang ditanam bersamaan dengan tanaman ini	
Nama	JAGUNG		
Jarak Tanam (cm)	40 X 90		KENTANG
Usia Panen (hari)	85 - 95		CABAI BESAR
Biaya Produksi (Rp / M2)	1,926.00		CABAI KECIL
Lahan Minimal (M2)	160.33		
Hasil Produksi (Kg/M2)	0.85		
Harga Jual (Rp/Kg)	3,200.00		
Keuntungan (Rp/M2)	784.40		
		Simpan	Tutup

Gambar 5.5 Implementasi Submenu *Edit* Data Tanaman

BAB VI

PENGUJIAN DAN ANALISA

Pada bab ini akan dibahas mengenai hasil uji coba yang telah dilakukan dalam optimasi penggunaan lahan pertanian dengan menggunakan algoritma genetika. Pengujian dilakukan untuk mengetahui parameter algoritma genetika yang optimal agar dapat menghasilkan *fitness* terbaik. Sesuai dengan perancangan pada bab 4, terdapat 4 macam pengujian yang akan dilakukan yaitu pengujian ukuran generasi, pengujian ukuran populasi, dan pengujian kombinasi *crossover rate* dan *mutation rate* serta pengujian perbandingan seleksi, dimana masing-masing pengujian tersebut dibandingkan terhadap nilai *fitness*.

6.1 Hasil Pengujian Sistem

Hasil pengujian terhadap sistem dilakukan untuk mengetahui kromosom terbaik dengan nilai *fitness* tertinggi yang dihasilkan dari proses algoritma genetika. Dalam uji coba ini akan dilakukan pengujian terhadap masukan luas lahan sebesar 14.000 meter persegi, modal Rp 10.000.000, akan menanam jagung, kedelai dan kacang hijau. Kemudian dilakukan inisialisasi parameter algoritma genetika dengan banyak populasi 20 kromosom kemudian diproses sejumlah 1 generasi, nilai *crossover rate* 0,5 dan *mutation rate* 0,5. Hasil pengujian sistem yang menghasilkan 20 kromosom terbaik pada generasi satu dapat dilihat pada Tabel 6.1.

Tabel 6.1 Hasil Pengujian Sistem

Jagung	Kedelai	Kacang Hijau	Fitness
0.2566	0.2728	0.4706	8006388
0.1864	0.3048	0.5087	7917850.8
0.5299	0.1056	0.3645	7866990.8
0.2172	0.2607	0.5221	7852081.2
0.5913	0.1798	0.093	7774533.37
0.2565	0.2061	0.5374	7773187.2
0.5088	0.1029	0.3883	7765154
0.5717	0.1900	0.1033	7759909.31
0.2638	0.1958	0.5404	7757980.8
0.5372	0.0972	0.3656	7757619.2
0.2321	0.189	0.5789	7644080.4
0.1669	0.2009	0.6322	7500206.4

Jagung	Kedelai	Kacang Hijau	Fitness
0.3361	0.2487	0.4151	7433990.54
0.5671	0.1776	0.0554	7157644
0.6154	0.0276	0.279	6408504.8
0.2536	0.2382	0.2244	6082281.6
0.0274	0.1775	0.7951	5450555.6
0.1218	0.3812	0.117	5160029.6
0.1800	0.0838	0.4929	4995520.4
0.0735	0.115	0.3879	3017767.6

6.2 Hasil dan Analisa Pengujian Parameter Algoritma Genetika

Proses uji coba ini dilakukan untuk mengetahui berapa *fitness* tertinggi dari kromosom terbaik pada setiap percobaan parameter algoritma genetika. Parameter algoritma genetika yang terdiri dari banyak populasi, banyak generai, *crossover rate* dan *mutation rate*. Hasil pengujian ini kemudian dianalisa untuk diambil suatu kesimpulan dari penelitian ini.

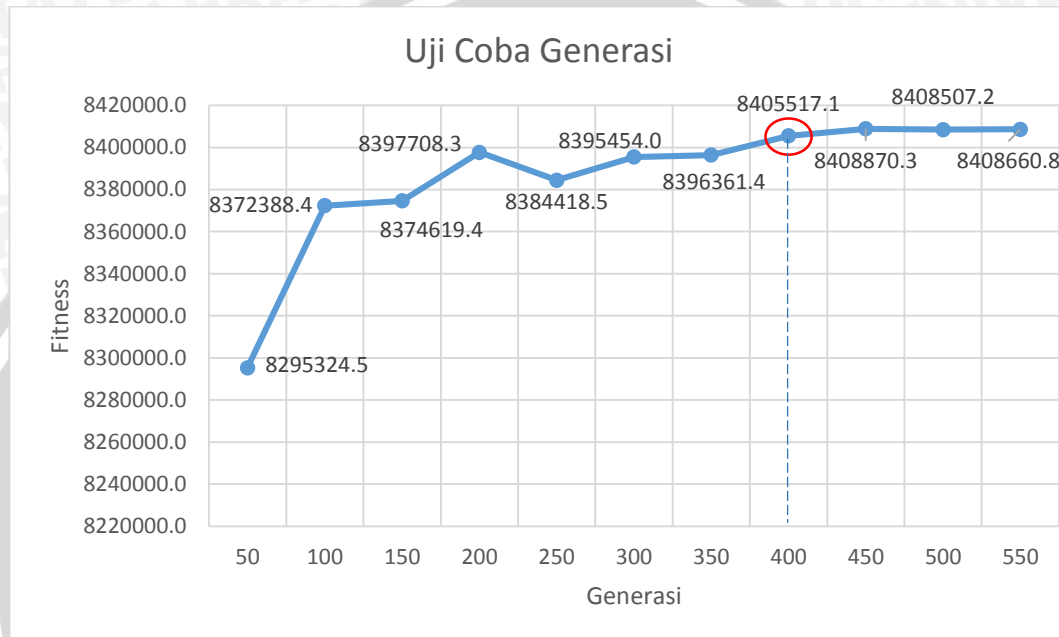
6.2.1 Hasil dan Analisa Uji Coba Banyaknya Generasi

Pengujian ini bertujuan untuk mengetahui jumlah generasi yang optimal untuk menghasilkan rata – rata *fitness* terbaik pada penelitian ini. Data yang digunakan dalam pengujian adalah data dengan luas lahan 14.000 meter persegi, modal Rp 10.000.000 serta jenis tanaman yang dipilih adalah tanaman jagung, kedelai dan kacang hijau. Jumlah populasi yang digunakan sebanyak 50 individu dengan banyak generasi kelipatan 50 mulai dari 50 sampai 550 generasi. Nilai *crossover rate* dan *mutation rate* yang digunakan adalah 0.5 dan 0.5. Setiap generasi dilakukan pengujian sebanyak 10 kali percobaan dan dihitung rata-rata nilai *fitness* nya. Dari uji coba tersebut akan diperoleh berapa banyak generasi yang optimal untuk pemecahan masalah pada penelitian ini. Hasil uji coba generasi diampikan pada Tabel 6.2.

Tabel 6.2 Hasil Percobaan Banyaknya Generasi

Banyak	Nilai fitness percobaan ke-										Rata-Rata	Rata-Rata
Genrasi	1	2	3	4	5	6	7	8	9	10	Fitness	Eksekusi (ms)
50	8419972.4	8250677.5	8375613.1	8368811.8	8066761.5	8038433.3	8313230.4	8423224.6	8389404.0	8307116.4	8295324.5	490
100	8424640.0	8266165.6	8424315.9	8362515.2	8420077.4	8420860.0	8383409.8	8241329.6	8355856.6	8424714.2	8372388.4	690
150	8420333.6	8224084.7	8327391.6	8424022.6	8280723.0	8420078.8	8424383.8	8423842.0	8384402.1	8416931.6	8374619.4	885
200	8347050.4	8424533.6	8421673.4	8424022.6	8424714.2	8422141.0	8327829.3	8338804.8	8422141.0	8424172.4	8397708.3	980
250	8420801.2	8408254.7	8423985.2	8424714.2	8209336.0	8317212.5	8372840.8	8421599.2	8423842.0	8421599.2	8384418.5	1043
300	8422682.8	8424383.8	8396234.1	8345320.2	8422832.6	8334200.6	8422395.8	8339273.3	8422832.6	8424383.8	8395454.0	1076
350	8418919.6	8415517.6	8394882.8	8409797.2	8337890.8	8297019.9	8421779.8	8421599.2	8423374.4	8422832.6	8396361.4	1130
400	8301877.7	8423879.3	8424714.2	8405360.4	8405360.4	8410519.6	8424533.6	8416781.8	8423842.0	8418302.2	8405517.1	1203
450	8423842.0	8422682.8	8422141.0	8406213.2	8409797.2	8422141.0	8422832.6	8318610.5	8420801.2	8419642.0	8408870.3	1209
500	8421523.6	8421673.4	8423406.6	8423013.2	8298902.5	8421599.2	8406213.2	8422141.0	8424458.0	8422141.0	8408507.2	1245
550	8406213.2	8421131.6	8422832.6	8318610.5	8423692.8	8406213.2	8422141.0	8419100.2	8422215.2	8424458.0	8408660.8	1350

Berdasarkan hasil uji coba banyaknya generasi pada Tabel 6.2 dapat diketahui bahwa pada setiap uji coba dengan jumlah generasi yang berbeda, akan menghasilkan rata-rata nilai *fitness* yang berbeda juga. Dari data tersebut dapat dibuat sebuah grafik untuk melihat perbedaan dari hasil pengujian banyaknya generasi terhadap nilai *fitness* yang dapat dilihat pada grafik Gambar 6.1.



Gambar 6.1 Grafik Hasil Uji Coba Banyaknya Generasi

Dari grafik Gambar 6.1 dapat dilihat bahwa jumlah generasi berpengaruh terhadap algoritma genetika yang terlihat pada nilai *fitness* nya. Nilai *fitness* terendah terdapat pada generasi 50 dikarenakan jumlah generasi tersebut masih kurang untuk memproses dengan menggunakan algoritma genetika secara optimal, karena area pencarian algoritma genetika masih sempit (Mahmudy, 2013). Secara umum semakin banyak generasi maka akan didapatkan nilai *fitness* yang lebih baik, tetapi terlalu banyak generasi membuat waktu proses semakin panjang. Dari Gambar 6.1 tersebut, bisa disimpulkan bahwa jumlah generasi yang paling optimal adalah dengan jumlah generasi 400, karena setelah generasi 400 sulit didapatkan nilai *fitness* yang lebih baik. Kondisi ini yang disebut konvergensi, artinya hampir semua kromosom mempunyai nilai yang berdekatan sehingga tidak memungkinkan diproduksi kromosom yang lebih baik (Mahmudy, 2013).

6.2.2 Hasil Dan Uji Coba Ukuran Populasi

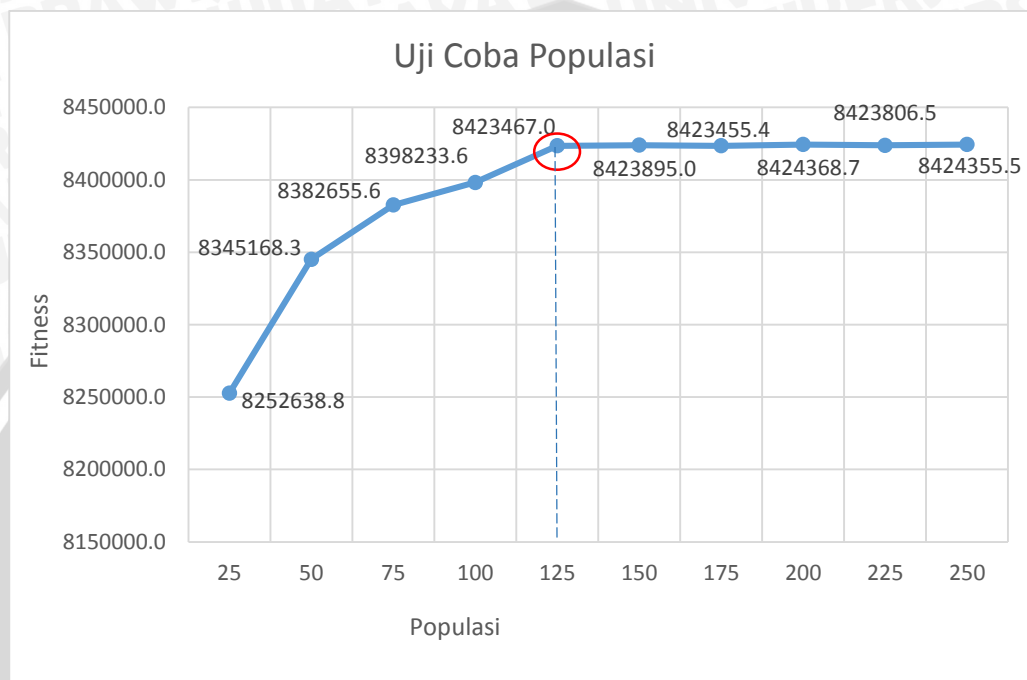
Pengujian ini bertujuan untuk mengetahui ukuran populasi yang optimal untuk menghasilkan rata – rata nilai *fitness* terbaik pada penelitian ini. Data yang digunakan dalam pengujian populasi adalah data dengan luas lahan 14.000 meter persegi, modal Rp 10.000.000 serta jenis tanaman yang dipilih adalah tanaman jagung, kedelai dan kacang hijau. Jumlah generasi yang dipakai adalah 400 dengan banyak populasi kelipatan 25 dari 25 sampai 250 populasi. Nilai *crossover rate* dan *mutation rate* yang digunakan adalah 0,5 dan 0,5. Setiap generasi dilakukan sebanyak 10 kali percobaan dan dihitung rata-rata nilai *fitness* nya. Dari uji coba tersebut akan diperoleh berapa ukuran populasi yang optimal untuk menyelesaikan masalah pada penelitian ini. Hasil dari uji coba populasi bisa dilihat pada Tabel 6.3.



Tabel 6.3 Hasil Uji Coba Populasi

Banyak Populasi	Nilai fitness percobaan ke-										Rata-Rata Fitness	Rata-rata Eksekusi (ms)
	1	2	3	4	5	6	7	8	9	10		
25	8108663.5	8424640.0	7794835.0	8359920.3	8422246.0	8268867.2	8226391.7	8179614.2	8318964.3	8422246.0	8252638.8	1163
50	8420440.0	8240768.4	8423119.6	8424640.0	8422321.6	8146077.7	8320389.6	8423119.6	8311766.4	8319039.5	8345168.3	1204
75	8353580.3	8424315.9	8420440.0	8406575.8	8424129.0	8278689.0	8371134.9	8387086.4	8424315.9	8336289.0	8382655.6	1230
100	8424640.0	8424640.0	8424139.1	8424172.4	8325442.5	8423819.7	8401721.2	8421779.8	8424024.1	8287956.9	8398233.6	1284
125	8416781.8	8424315.9	8423406.6	8424201.2	8424172.4	8424640.0	8424353.0	8424423.8	8424353.0	8424022.6	8423467.0	1305
150	8423555.0	8424640.0	8424353.0	8424533.6	8424422.0	8421312.2	8423555.0	8423661.4	8424383.8	8424533.6	8423895.0	1346
175	8423480.8	8423480.8	8423300.2	8424201.2	8422246.0	8424383.8	8424172.4	8424422.0	8421312.2	8423555.0	8423455.4	1402
200	8424640.0	8424640.0	8422682.8	8424714.2	8424353.0	8424640.0	8424640.0	8424640.0	8424203.2	8424533.6	8424368.7	1428
225	8424640.0	8424640.0	8424315.9	8422246.0	8422246.0	8424640.0	8424353.0	8424315.9	8422246.0	8424422.0	8423806.5	1452
250	8424640.0	8423842.0	8424640.0	8424143.2	8424640.0	8423916.2	8424383.8	8424533.6	8424432.8	8424383.8	8424355.5	2497

Hasil uji coba banyaknya populasi terhadap nilai *fitness* terdapat pada Tabel 6.4. Dari data tersebut dibuat sebuah grafik untuk melihat perbedaan dari hasil pengujian banyaknya populasi terhadap nilai *fitness* yang dapat dilihat pada Gambar 6.2.



Gambar 6.2 Grafik Hasil Uji Coba Banyaknya Populasi

Dari grafik Gambar 6.2 dapat dilihat bahwa ukuran populasi berpengaruh terhadap hasil algoritma genetika yang terlihat pada nilai *fitness* nya. Nilai *fitness* terendah terdapat pada populasi 25 dikarenakan ukuran populasi masih sedikit untuk memproses data dengan menggunakan algoritma genetika, sehingga area eksplorasi algoritma genetika semakin sempit dan solusi yang ditemukan tidak akan terlalu baik. Namun, terlalu besar ukuran populasi juga belum tentu hasil algoritma genetika menjadi lebih optimal. Terlalu besar ukuran populasi yang digunakan membuat waktu proses semakin panjang. Dari grafik Gambar 6.2 tersebut bisa disimpulkan bahwa ukuran populasi yang optimal adalah dengan jumlah populasi 125, karena setelah jumlah populasi 125 sulit didapatkan nilai *fitness* yang lebih baik. Kondisi ini yang disebut konvergensi, artinya hampir semua kromosom mempunyai nilai yang berdekatan sehingga tidak memungkinkan diproduksi kromosom yang lebih baik (Mahmudy, 2013).

6.2.3 Hasil dan Analisa Uji Coba Kombinasi *Crossover Rate* dan *Mutation Rate*

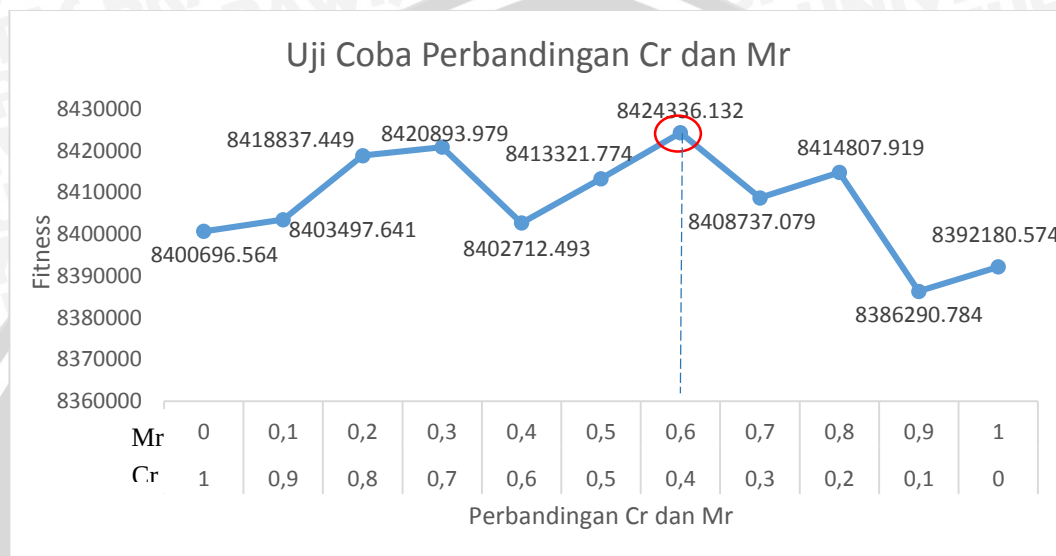
Uji coba kombinasi *crossover rate* (cr) dan *mutation rate* (mr) bertujuan untuk mengetahui kombinasi cr dan mr yang optimal pada penelitian ini. Jumlah populasi yang digunakan adalah jumlah populasi yang menghasilkan nilai *fitness* terbaik pada uji coba ukuran populasi yaitu 125. Banyak generasi yang digunakan adalah generasi terbaik dari proses uji coba generasi yaitu 400. Nilai *crossover rate* dan *mutation rate* yang digunakan dalam pengujian adalah nilai antara 0 sampai dengan 1. Setiap generasi dilakukan percobaan sebanyak 10 kali dan dihitung rata-rata nilai *fitness* nya. Dari uji coba tersebut akan diperoleh berapa besar *crossover rate* dan *mutation rate* yang paling optimal untuk menyelesaikan masalah optimasi penggunaan lahan pertanian. Hasil dari uji coba kombinasi *crossover rate* dan *mutation rate* dapat dilihat pada Tabel 6.4.



Tabel 6.4 Hasil Uji Coba Kombinasi *Crossover rate* dan *Mutation rate*

Kombinasi		Nilai fitness percobaan ke-										Rata-Rata Fitness
Cr	Mr	1	2	3	4	5	6	7	8	9	10	
1	0	8424353.0	8334396.8	8392976.2	8423193.8	8424383.8	8415425.2	8424640.0	8423916.2	8319658.0	8424022.6	8400696.6
0.9	0.1	8424172.4	8423374.4	8424022.6	8279441.9	8424022.6	8423976.4	8424353.0	8416125.2	8424353.0	8371134.9	8403497.6
0.8	0.2	8424203.2	8424385.7	8421673.4	8424315.9	8424533.6	8423300.2	8424353.0	8424533.6	8375115.5	8421960.4	8418837.4
0.7	0.3	8423936.1	8420108.2	8423300.2	8423480.8	8422832.6	8424315.9	8423842.0	8424203.2	8423585.8	8399335.0	8420894.0
0.6	0.4	8417038.0	8424640.0	8423442.2	8422034.6	8424357.6	8423948.4	8225821.4	8424315.9	8423480.8	8418046.0	8402712.5
0.5	0.5	8424640.0	8424640.0	8424315.9	8362515.2	8420077.4	8420860.0	8383409.8	8424203.2	8423842.0	8424714.2	8413321.8
0.4	0.6	8424640.0	8424127.6	8423661.4	8424353.0	8424640.0	8424714.2	8424640.0	8424315.9	8424172.4	8424096.8	8424336.1
0.3	0.7	8424640.0	8424096.8	8424640.0	8424533.6	8293930.8	8423871.4	8399297.2	8424496.4	8424022.6	8423842.0	8408737.1
0.2	0.8	8424353.0	8420801.2	8423871.4	8424640.0	8424640.0	8424383.8	8423842.0	8424458.0	8332773.9	8424315.9	8414807.9
0.1	0.9	8424383.8	8396329.8	8424172.4	8423735.6	8370748.2	8132158.7	8424203.2	8424640.0	8421193.2	8421343.0	8386290.8
0	1	8411954.2	8405114.2	8423508.8	8419924.8	8422485.4	8243901.2	8420284.6	8418739.0	8332773.9	8423119.6	8392180.6

Hasil uji coba kombinasi *crossover rate* dan *mutation rate* terhadap nilai *fitness* terdapat pada Tabel 6.5. Dari data tersebut, dapat dibuat sebuah grafik untuk melihat perbedaan nilai *fitness* dari hasil pengujian kombinasi *crossover rate* dan *mutation rate*. Grafik hasil uji coba kombinasi *crossover rate* dan *mutation rate* dapat dilihat pada Gambar 6.3.



Gambar 6.3 Hasil Uji Coba Kombinasi *Crossover rate* dan *Mutation rate*

Pada Gambar 6.3 dapat dilihat *fitness* rata – rata terbaik adalah 8424336.13 yaitu pada kombinasi *crossover rate* 0,4 dan *mutation rate* 0,6. Rata – rata *fitness* terendah adalah 8386290.8, terdapat pada kombinasi *crossover rate* 0,1 dan *mutation rate* 0,9. Dapat disimpulkan bahwa kombinasi *crossover rate* dan *mutation rate* terbaik 0,4 : 0,6. Nilai *crossover rate* dan *mutation rate* yang terlalu rendah kurang bagus untuk algoritma genetika dalam melakukan proses eksplorasi dan eksploitasi untuk menemukan solusi dari suatu permasalahan.

6.2.4 Hasil dan Analisa Perbandingan Seleksi Metode *Elitism* dan *Binary Tournament*

Pada pengujian ini dilakukan untuk membandingkan metode seleksi mana yang terbaik pada permasalahan optimasi penggunaan lahan pertanian ini. Pengujian setiap metode seleksi dilakukan dengan skenario uji seperti berikut :

1. Menguji program dengan metode seleksi *elitism* dengan jumlah generasi 400, jumlah populasi 125, serta kombinasi nilai Cr dan Mr adalah 0,4 dan 0,6. Pengujian dilakukan sebanyak 10 kali percobaan.

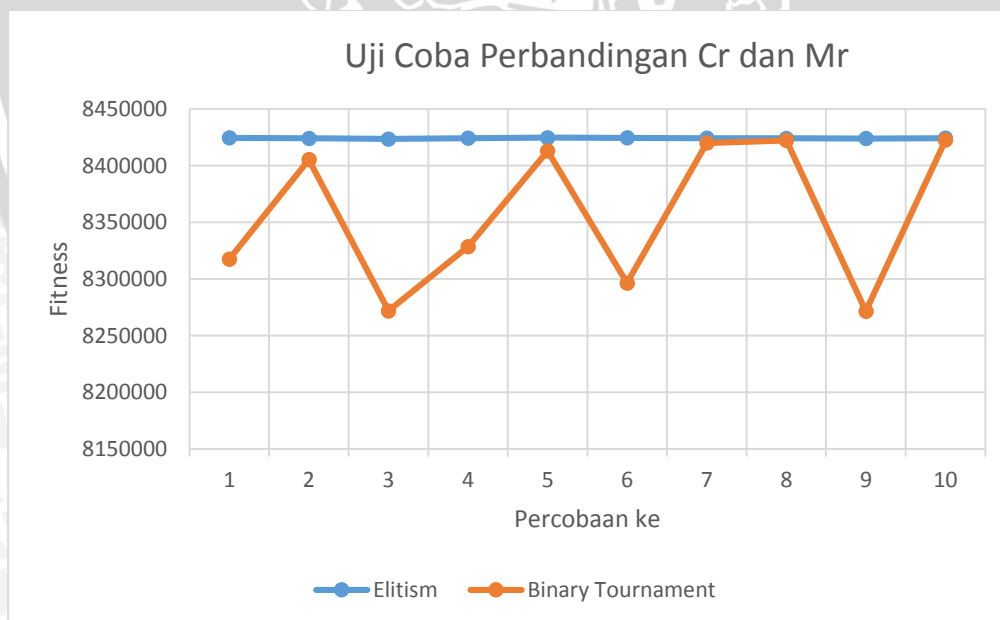
2. Menguji program dengan metode seleksi *Binary Tournament* dengan jumlah generasi 400, jumlah populasi 125, serta kombinasi nilai Cr dan Mr adalah 0,4 dan 0,6. Pengujian dilakukan sebanyak 10 kali percobaan.

Hasil dari setiap metode seleksi akan didapatkan nilai *fitness* dari masing-masing percobaan, untuk mengetahui metode seleksi mana yang lebih baik untuk menghasilkan nilai *fitness* yang optimal. Hasil rata-rata nilai *fitness* pada percobaan metode seleksi *elitism* dan *binary tournament* ditunjukkan pada Tabel 6.5.

Tabel 6.5 Hasil Uji Coba Metode Seleksi

Metode Seleksi	Nilai fitness percobaan ke-										Rata-rata Fitness
	1	2	3	4	5	6	7	8	9	10	
Elitism	8424640	8424128	8423661	8424353	8424714	8424640	8424316	8424172	8424097	8424302	8424302
Binary Tournament	8317674	8405342	8271876	8328564	8413019	8296458	8420079	8422322	8271623	8422683	8356964

Dari Tabel 6.5, terlihat bahwa ada perbedaan nilai *fitness* ketika diuji dengan metode seleksi *elitism* dan *binary tournament*. Dari data pada tabel tersebut dapat dibuat sebuah grafik untuk melihat perbedaan nilai *fitness* dari setiap percobaan yang dapat dilihat pada Gambar 6.4.



Gambar 6.4 Perbedaan Nilai Fitness Uji Coba Seleksi *Elitism* dan *Binary Tournament*

Pada Gambar 6.4 merupakan grafik hasil uji coba metode seleksi *elitism* dan *binary tournament*. Dari grafik tersebut, dapat disimpulkan bahwa nilai *fitness* dari seleksi metode *elitism* selalu lebih baik daripada metode *binary tournament*, dengan rata-rata nilai *fitness* metode *elitism* adalah 8424302. Hasil ini lebih baik daripada rata-rata nilai *fitness* seleksi dengan metode *binary tournament* yaitu 8356964. Hal ini membuktikan bahwa pada penelitian ini, seleksi dengan metode *elitism* hasilnya lebih baik dari pada *binary tournament*. Hasil ini sesuai dengan penelitian yang dilakukan Putri (2014) yang membandingkan metode seleksi *elitism* dan *binary tournament*. Dari setiap percobaan yang dilakukan, seleksi *elitism* selalu menghasilkan nilai *fitness* lebih tinggi, karena pada seleksi *elitism*, semua individu terbaik sebanyak populasi langsung terpilih menjadi generasi selanjutnya. Sedangkan pada *binary tournament*, dilakukan random individu terlebih dahulu untuk dilakukan perbandingan *fitness*, sehingga individu dengan *fitness* rendah masih berpeluang untuk menjadi generasi selanjutnya (Putri, 2014).

6.3 Analisa Hasil Pengujian Parameter Optimal Algoritma Genetika

Berdasarkan hasil pengujian yang dilakukan pada sistem dengan menggunakan parameter algoritma genetika yang optimal yaitu dengan jumlah populasi sebanyak 125, jumlah generasi sebanyak 400, kombinasi *crossover rate* 0,4 dan *mutation rate* 0,6 didapatkan kromosom-kromosom terbaik sebagai solusi untuk menentukan proporsi luas lahan optimal untuk setiap jenis tanaman tertentu yang dipilih. Dari 5 kali percobaan didapatkan hasil rekomendasi proporsi luas optimal dengan nilai *fitness* terbesar ditunjukkan dalam Tabel 6.6. Sementara penentuan proporsi yang direkomendasikan pakar ditampilkan pada Tabel 6.7.

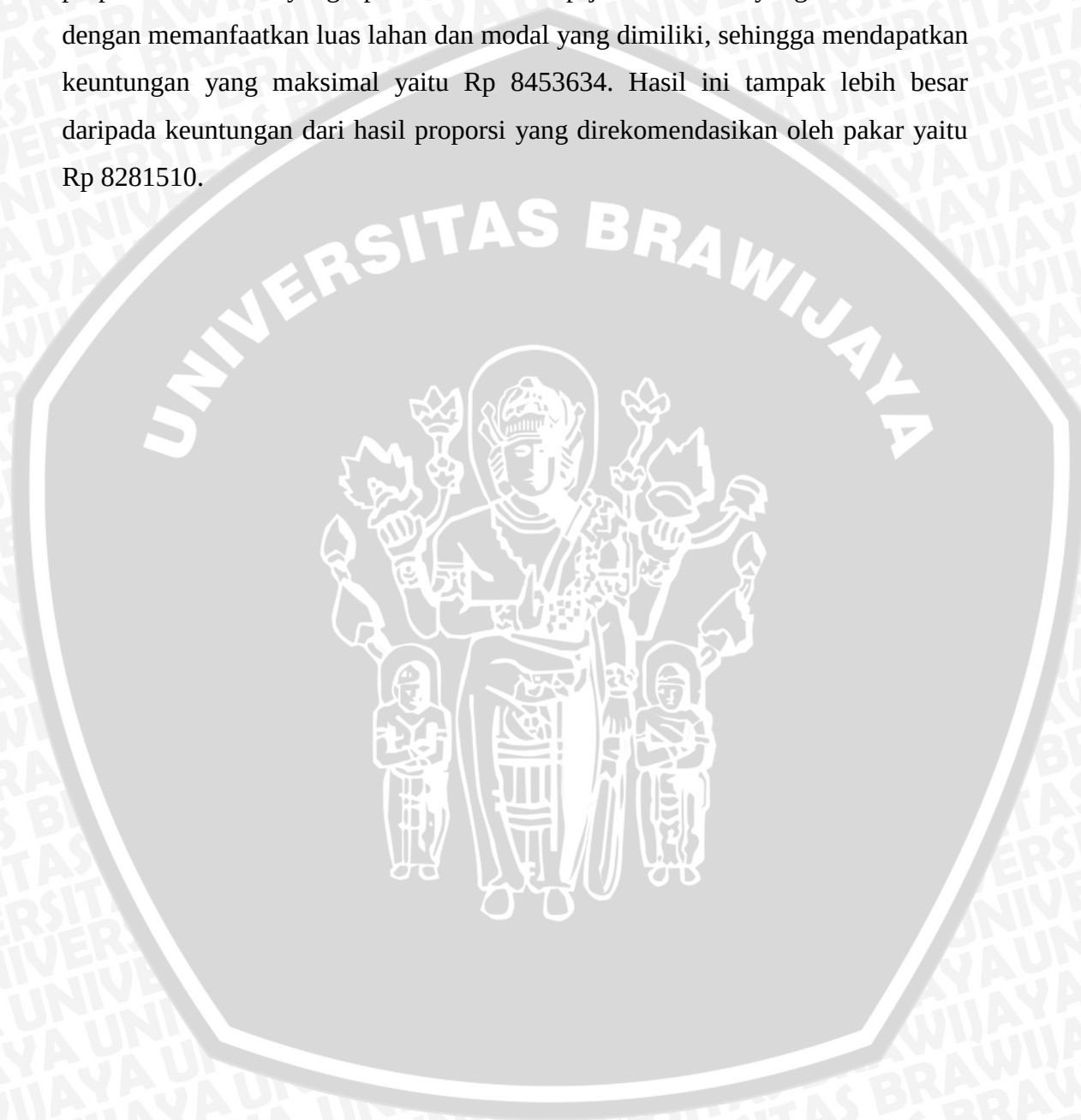
Tabel 6.6 Rekomendasi Proporsi Luas Optimal

Jagung (M ²)	Kedelai (M ²)	Kacang Hijau (M ²)	Biaya Produksi (Rp)	Keuntungan (Rp)
3070.2	1198.4	9730	9999245.2	8453634

Tabel 6.7 Rekomendasi Proporsi Oleh Pakar

Jagung (M ²)	Kedelai (M ²)	Kacang Hijau (M ²)	Biaya Produksi (Rp)	Keuntungan (Rp)
4125	2000	6630	9998163	8281510

Berdasarkan perbandingan Tabel 6.6 dan Tabel 6.7, maka dapat disimpulkan bahwa hasil akhir dari sistem dapat merekomendasikan penentuan proporsi luas lahan yang optimal untuk setiap jenis tanaman yang akan ditanam dengan memanfaatkan luas lahan dan modal yang dimiliki, sehingga mendapatkan keuntungan yang maksimal yaitu Rp 8453634. Hasil ini tampak lebih besar daripada keuntungan dari hasil proporsi yang direkomendasikan oleh pakar yaitu Rp 8281510.



BAB VII

PENUTUP

7.1 Kesimpulan

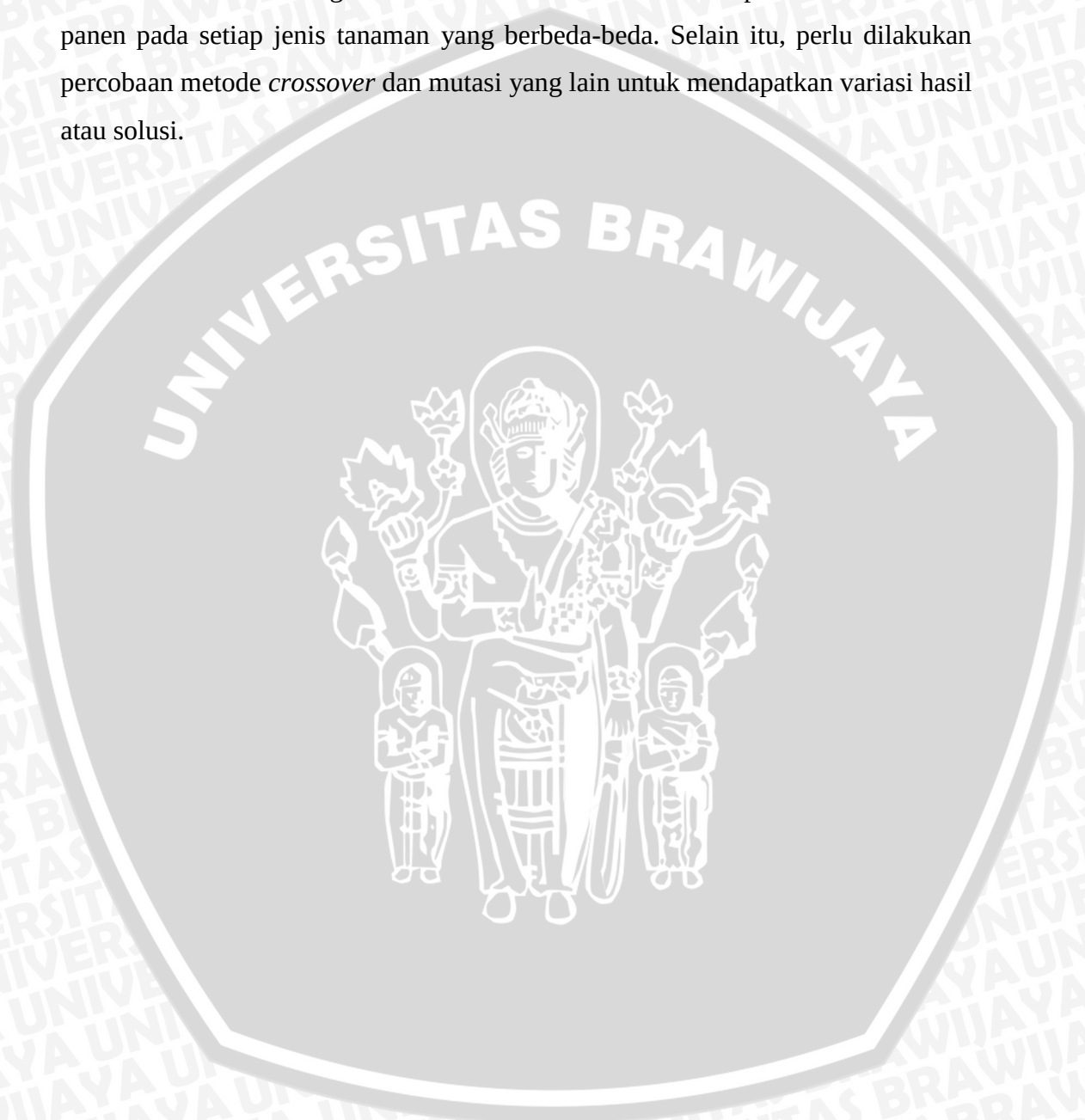
Kesimpulan yang didapat dari hasil uji coba yang telah dilakukan dalam penelitian ini adalah sebagai berikut :

1. Representasi kromosom dengan cara *real coded* dengan proses reproduksi *extended intermediate crossover* dan *random mutation* yang digunakan dalam penelitian ini mampu menyelesaikan permasalahan optimasi dalam penggunaan lahan pertanian. Algoritma genetika mampu menentukan proporsi penggunaan luas lahan yang optimal pada setiap jenis tanaman yang akan ditanam dengan memanfaatkan luas lahan dan modal yang dimiliki untuk menghasilkan keuntungan yang maksimal.
2. Dari hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa nilai parameter algoritma genetika berpengaruh terhadap hasil optimasi yang dihasilkan. Ukuran parameter yang kecil menyebabkan area pencarian algoritma genetika semakin sempit. Tetapi, ukuran parameter yang terlalu besar menyebabkan waktu komputasi semakin lama.
3. Dari hasil uji coba generasi, didapatkan generasi yang paling optimal adalah 400 dengan rata-rata nilai *fitness* 8405517,1. Sementara ukuran populasi yang paling optimal adalah 125 dengan rata-rata nilai *fitness* 8423467,0. Kombinasi *crossover rate* dan *mutation rate* yang terbaik dalam penelitian ini secara berturut-turut adalah 0,4 dan 0,6 dengan rata-rata nilai *fitness* 8424336,1.
4. Dengan menggunakan parameter terbaik pada algoritma genetika dalam penyelesaian permasalahan optimasi penggunaan lahan pertanian ini, didapatkan hasil nilai keuntungan yang optimal dengan mengalokasikan modal dan lahan yang dimiliki petani.

7.2 Saran

Saran yang dapat diberikan setelah menyelesaikan penelitian ini adalah perbaikan solusi yang dihasilkan bisa melakukan penggabungan (*hybrid*) algoritma

genetika dengan algoritma lainya. Teknik penggabungan ini terbukti efektif dalam penelitian sebelumnya (Mahmudy, 2008). Untuk pengembangan selanjutnya, agar dapat melakukan penambahan analisis terhadap faktor kualitatif seperti karakteristik dan kandungan unsur hara dalam tanah serta memperhatikan masa usia panen pada setiap jenis tanaman yang berbeda-beda. Selain itu, perlu dilakukan percobaan metode *crossover* dan mutasi yang lain untuk mendapatkan variasi hasil atau solusi.



DAFTAR PUSTAKA

- Anindita, Yoga Anung. 2013. Makalah Polikultur Sebagai Bagian Dari Pertanian Berkelanjutan. Universitas Sebelas Maret. Surakarta.
- Dinas Pertanian Tanaman Pangan Jawa Barat. 2008. Rencana Strategis Dinas Pertanian Tanaman Pangan Provinsi Jawa Barat 2008-2013. Bandung.
- Fiarni, Cut dan Bastiyan. 2013, *Sistem Rekomendasi Portofolio Investasi Berbasis Algoritma Genetika*, Seminar Nasional Sistem Informasi Indonesia, 2 - 4 Desember 2013.
- Kholifah, U., Trisyani, N., & Yuniar, I. 2008. Pengaruh Padat Tebar yang Berbeda terhadap Kelangsungan Hidup dan Pertumbuhan pada Polikultur Udang Windu (*Penaeus Monodon Fab*) dan Ikan Bandeng (*Chanos Chanos*) pada Hapa di Tambak Brebes. Jawa Tengah
- Mahmudy, Wayan Firdaus. 2013. Algoritma Evolusi. Program Teknologi Informasi dan Ilmu Komputer. Universitas Brawijaya. Malang
- Mahmudy, WF., Marian, RM & Luong, LHS. 2013. *Optimization of part type selection and loading probPrelem with alternative production plans in flexible manufacturing system using hybrid genetic algorithm – Part 1 : Modelling and representation*, 5th International Conference on Knowledge and Smart Technology (KST), Chonburi, Thailand, 31 Jan – 1 Feb, pp. 75-80.
- Mahmudy, Wayan Firdaus. 2008. Optimasi Fungsi Tanpa Kendala menggunakan Algoritma Genetika dengan Kromosom Biner dan Perbaikan Kromosom Hill Climbing, *Kursor*, Vol 4, No.1, pp. 23-29.
- Mahudy, Wayan Firdaus. 2009. Optimasi Fungsi Tak Berkendala Menggunakan Algoritma Genetika Terdistribusi dengan Pengkodean Real. Seminar Nasional Basic Science VI FMIPA. Universitas Brawijaya.
- Megawati, Prasetyo., Soekarto, SD. 2014. Hubungan jumlah baris kacang-kacangan terhadap hama tanaman jagung dan tanaman kacang-kacangan. Universitas Jember

- Putri, FB, Mahmudy, WF & Ratnawati, DE. 2014. Penerapan Algoritma Genetik untuk Vehicle Routing Problem with Time Windows (VRPTW). Pada Kasus Optimasi Distribusi Beras Bersubsidi. DORO : Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, Vol 5, No.1..
- Sabirin, Mazuki, Pasaribu TW & Lestari A. 2010. Ebook Modul Sekolah Lapang Polikultur : BITRA Indonesia. Medan. <http://content.yudu.com/Library/A1oirr/ModulKebunTanamanCam/resources/37.htm>. [Diakses 1 April, 2015]
- Sari, AP., Mahmudy, WF & Dewi, C. 2014. Optimasi Asupan Gizi Pada Ibu Hamil dengan Menggunakan Algoritma Genetika. DORO : Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, Vol 4, No. 5.
- Smith, Alice.E, Coit, David W. 1996, Penalty Functions, Section C 5.2 of Handbook of Evolutionary Computation, Pensylvania : University of Pittsburgh, Pittsburgh, Pensylvania, USA.
- Sunu, Pratignja MP., Wartoyo SP. 2006. Buku Ajar Dasar- Dasar Hortikultura. Universitas Sebelas Maret. Surakarta.
- Sutanta, Edhy. 2005. Aplikasi Sistem Pendukung Pengambilan Keputusan untuk Optimalisasi Penggunaan Lahan Pertanian. Institut Sains & Teknologi AKPRIND. Yogyakarta.
- Tyas, Rizkhy Ayuning, 2013. Implementasi Algoritma Genetika untuk Optimasi 0/1 Multi-Dimensional Knapsack Problem dalam Penentuan Menu akan Sehat. Program Teknologi Informasi dan Ilmu Komputer. Universitas Brawijaya.
- Wardhani, LK., Safrizal, M. & Chair, A. 2011. Optimasi Komposisi Bahan Pakan Ikan Air Tawar Menggunakan Metode Multi-Objective Genetic Algorithm. UIN Sultan Syarif Kasim. Riau. ISSN : 1907-5022, Hal A-113.
- Warsani, Henki. 2013. Kajian Pemanfaatan Lahan Sawah Di Kecamatan Kuantan Tengah Kabupaten Kuantan Singingi. Universitas Pendidikan Indonesia.
- Zulkarnain. 2010. Dasar – Dasar Hortikultura Pertanian Organik. Bumi Aksara. Jakarta.

