

**OPTIMASI KOMPOSISI PUPUK UNTUK SISTEM
PENANAMAN TUMPANGSARI MENGGUNAKAN ALGORITMA
PARTICLE SWARM OPTIMIZATION (PSO)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Pasca Imanuddin Akbar
NIM: 125150200111052



PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

OPTIMASI KOMPOSISI PUPUK UNTUK SISTEM
PENANAMAN TUMPANGSARI MENGGUNAKAN ALGORITMA
PARTICLE SWARM OPTIMIZATION (PSO)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Pasca Imanuddin Akbar
NIM: 125150200111052

Skripsi ini telah diuji dan dinyatakan lulus pada
30 Mei 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Imam Cholissodin, S.Si, M.Kom
NIK: 201201 850719 1 001

Drs. Achmad Ridok, M.Kom
NIP: 19680825 199403 1 002

Mengetahui
Pjs. Ketua Program Studi Informatika / Ilmu Komputer

Issa Arwani, S.Kom, M.Sc
NIP: 19830922 201212 1 003

PERNYATAAN ORISINALITAS

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 disitasi dalam naskah ini dan disebutkan dalam 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, 30 Mei 2015

Pasca Imanuddin Akbar

NIM. 125150200111052



KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah *Subhanahu wa ta'ala* yang telah memberikan rahmat dan karunianya, sehingga penulis dapat membuat dan menyelesaikan skripsi dengan judul “Optimasi Komposisi Pupuk Untuk Sistem Penanaman Tumpangsari Menggunakan Algoritma *Particle Swarm Optimization* (PSO)”.

Skripsi ini dapat hadir seperti sekarang ini tak lepas dari bantuan banyak pihak. Untuk itu sudah sepantasnya kami mengucapkan rasa terima kasih kepada mereka yang telah berjasa membantu penulis selama proses pembuatan makalah ini dari awal hingga akhir. Atas dukungan moral dan materil yang diberikan dalam penyusunan makalah ini, maka penulis mengucapkan banyak terima kasih kepada:

1. Bapak Imam Cholissodin, S. Si., M. Kom. selaku dosen pembimbing 1 yang memberikan arahan serta masukan dalam pengerjaan skripsi ini.
2. Bapak Drs. Achmad Ridok, M. Kom. selaku dosen pembimbing 2 yang memberikan arahan serta masukan dalam pengerjaan skripsi ini.
3. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, PhD selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Issa Arwani, S. Kom., M. Sc selaku Kepala Program Studi Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
5. Seluruh dosen dan karyawan Program Studi Informatika Fakultas Ilmu Komputer Universitas Brawijaya, atas ilmu, bimbingan dan bantuannya hingga penulis selesai menyusun tugas akhir ini.
6. Orang tua yang telah membantu baik moril maupun materi.
7. Rekan-rekan Program Studi Informatika Fakultas Ilmu Komputer Universitas Brawijaya yang juga telah banyak membantu penulis.

Namun tidak lepas dari semua itu, kami menyadari sepenuhnya bahwa ada kekurangan baik dari segi materi maupun penyusun bahasanya. Oleh karena itu, kritik dan saran yang bersifat membangun sangat kami harapkan dalam penyempurnaan makalah ini.

Akhirnya penyusun mengharapkan semoga skripsi ini dapat diambil hikmah dan manfaatnya sehingga dapat memberikan inspirasi terhadap pembaca.

Malang, 30 Mei 2016

Pasca Imanuddin Akbar

gmacpasca.ia@gmail.com

ABSTRAK

Pasca Imanuddin Akbar, 2016. Optimasi Komposisi Pupuk untuk Sistem Penanaman Tumpangsari Menggunakan Algoritma *Particle Swarm Optimization* (PSO). Skripsi Program Studi Teknik Informatika, Fakultas Ilmu Komputer Universitas Brawijaya. Pembimbing: Imam Cholissodin, S.Si.,M,Kom. dan Drs. Achmad Ridok, M.Kom.

Indonesia merupakan negara beriklim tropis dengan jumlah pulau mencapai 17.504 sehingga memiliki wilayah yang cukup luas dan variasi agroklimat yang beragam. Permasalahan dalam produksi pertanian menyebabkan dibutuhkannya peralihan pola tanam. Penerapan sistem tumpangsari dapat meningkatkan hasil produksi pertanian dan meningkatkan efektivitas dalam penggunaan cahaya, air serta unsur hara, disamping dapat menekan pertumbuhan gulma. Untuk meningkatkan keberhasilan penerapan sistem tumpangsari diperlukan penggunaan dosis pupuk yang dapat memenuhi kebutuhan tanaman. Sedangkan penggunaan pupuk yang berlebihan dapat menyebabkan biaya yang dikeluarkan petani terlalu besar. Oleh karena itu dibutuhkan teknologi yang dapat memudahkan petani untuk mendapatkan komposisi pupuk yang dapat memenuhi tanaman tumpangsari dengan biaya seminimal mungkin, salah satunya sistem cerdas untuk optimasi campuran pupuk untuk sistem penanaman tumpangsari dengan menggunakan algoritma *Particle Swarm Optimization* (PSO).

Pada hasil pengujian *Particle Swarm Optimization* pada komposisi pupuk tanaman jagung dan kacang kedelai di Kabupaten Tulungagung Kecamatan Sumbergempol, jumlah iterasi yang digunakan adalah sebanyak 10 iterasi, komponen sosial yaitu $c1f$ sebesar 0,5, $c1i$ sebesar 2,5, $c2f$ sebesar 2,5 dan $c2i$ sebesar 0,5, jumlah partikel sebesar 80, jumlah iterasi 70 serta batas inersia yaitu w_{min} sebesar 0,4 dan w_{max} sebesar 0,6 didapatkan solusi yang dapat menghemat biaya sebesar Rp 79.380,00 atau 4,05% dari biaya pakar per hektar dengan bobot penalti sebesar 162.900. Hasil rekomendasi tersebut bila dibandingkan dengan pemupukan yang digunakan oleh petani dapat menghemat biaya sebesar Rp 982.580,00 atau 34,32% dan perbaikan bobot penalti sebesar 1.837.100.

Kata kunci: Tumpangsari, Optimasi, Biaya, *Particle Swarm Optimization*.

ABSTRACT

Pasca Imanuddin Akbar, 2016. Fertilizer Composition Optimization For Intercropping Planting Pattern Using *Particle Swarm Optimization (PSO) Algorithm*. Skripsi Program Studi Teknik Informatika, Fakultas Ilmu Komputer Universitas Brawijaya. Adviser: Imam Cholissodin, S.Si.,M,Kom. dan Drs. Achmad Ridok, M.Kom.

Indonesia is a tropical country with a number of islands reached 17,504 so it has a large area and diverse agro-climatic variations. Problems in agricultural production led to the need for shift cropping patterns. Application of cropping system can increase agricultural production and improve effectiveness in the use of light, water and nutrients, in addition to suppress the growth of weeds. To enhance the successful implementation of the cropping system required the use of fertilizers to meet crop needs. While the excessive use of fertilizers can cause the cost farmers too big. Therefore the need for technology that can enable farmers to obtain fertilizer composition that can meet intercropping with minimum cost, one intelligent systems to optimize the mix of fertilizer to the system by using algorithms multicropping Particle Swarm Optimization (PSO).

In the test results on the Particle Swarm Optimization fertilizer composition corn and soybeans in Tulungagung subdistrict Sumbergempol, the number of iterations used is 10 iterations, the social component is $c1f$ of 0.5, $c1i$ of 2.5, $c2f$ of 2.5 and $c2i$ by 0.5, the number of particles of 80, and the number of iterations of 70 and inertia that is w_{min} by 0.4 and 0.6 w_{max} obtained solutions can save as much as Rp 79,380.00 or 4.05% of the cost of experts per hectares with a weight penalty of 162 900. The result of these recommendations when compared with fertilizer used by farmers can save the cost of Rp 982,580.00 or 34.32% and improved weight penalty of 1.8371 million.

Keyword: Intercropping, Optimation, Cost, Particle Swarm Optimization.

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xi
DAFTAR PERSAMAAN.....	xiii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan Masalah.....	2
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Kajian Pustaka	4
2.2 Tumpangsari	6
2.2.1 Pengertian Tumpangsari.....	6
2.2.2 Jenis-jenis Tumpangsari	6
2.2.3 Kelebihan Sistem Pertanian Tumpangsari	7
2.2.4 Tumpangsari Jagung dan Kedelai.....	7
2.3 Pupuk	8
2.3.1 Macam-Macam Pupuk	8
2.3.2 Pemupukan Provinsi Jawa Timur	11
2.4 Particle Swarm Optimization	15
2.4.1 Inisialisasi Partikel Serta Membangkitkan Kecepatan Awal.....	16

2.4.2 Evaluasi Partikel / Menghitung Nilai <i>Fitness</i>	16
2.4.3 Mencari Partikel Lokal Terbaik atau <i>pBest</i>	16
2.4.4 Mencari Partikel Global Terbaik atau <i>gBest</i>	16
2.4.5 Memperbarui Kecepatan Partikel	16
2.4.6 Memperbarui Posisi Partikel	17
2.4.7 Bobot Inersia <i>Particle Swarm Optimization</i>	17
2.4.8 Komponen Kognitif dan Sosial <i>Particle Swarm Optimization</i>	18
BAB 3 METODOLOGI	19
3.1 Studi Literatur	19
3.2 Pengumpulan Data	20
3.3 Analisis Kebutuhan	20
3.4 Perancangan Sistem	20
3.5 Implementasi Sistem	20
3.6 Pengujian dan Evaluasi Hasil	21
3.7 Pengambilan Kesimpulan dan Saran	21
BAB 4 PERANCANGAN	22
4.1 Deskripsi Umum Sistem	22
4.2 Perancangan Implementasi <i>Particle Swarm Optimization</i>	22
4.2.1 Inisialisasi Partikel	24
4.2.2 Proses Evaluasi / Perhitungan <i>Fitness</i>	26
4.2.3 Proses Pencarian <i>pBest</i>	26
4.2.4 Proses Pencarian <i>gBest</i>	27
4.2.5 Proses Memperbarui Kecepatan	28
4.2.6 Proses Memperbarui Posisi	30
4.3 Contoh Perhitungan Manual <i>Particle Swarm Optimization</i>	31
4.3.1 Inisialisasi Partikel	32
4.3.2 Inisialisasi Kecepatan	33
4.3.3 Proses Evaluasi / Perhitungan <i>Fitness</i>	33
4.3.4 Proses Pencarian <i>pBest</i>	34
4.3.5 Proses Pencarian <i>gBest</i>	35
4.3.6 Menghitung Bobot Inersia	35
4.3.7 Menghitung Komponen Kognitif dan Sosial	35

4.3.8 Proses Memperbarui Kecepatan.....	36
4.3.9 Proses Memperbarui Posisi.....	36
4.4 Perancangan Antarmuka	37
4.4.1 Halaman Beranda.....	37
4.4.2 Halaman Optimasi Pupuk.....	38
4.4.3 Halaman Data.....	40
4.5 Skenario Pengujian	41
4.5.1 Pengujian Jumlah Partikel	42
4.5.2 Pengujian Jumlah Iterasi	42
4.5.3 Pengujian Batas Bobot Inersia	43
4.5.4 Pengujian Konvergensi.....	43
BAB 5 IMPLEMENTASI	44
5.1 Lingkungan Implementasi.....	44
5.1.1 Lingkungan Perangkat Keras	44
5.1.2 Lingkungan Perangkat Lunak	44
5.2 Batasan Implementasi	44
5.3 Implementasi Program	45
5.3.1 Implementasi Proses Inisialisasi.....	45
5.3.2 Implementasi Proses Evaluasi / Menghitung <i>Fitness</i>	46
5.3.3 Implementasi Proses Pencarian <i>pBest</i>	46
5.3.4 Implementasi Proses Pencarian <i>gBest</i>	47
5.3.5 Implementasi Proses Memperbarui Kecepatan	47
5.3.6 Implementasi Proses Memperbarui Posisi	48
5.4 Implementasi Antarmuka	49
5.4.1 Implementasi Antarmuka Halaman Beranda.....	49
5.4.2 Implementasi Antarmuka Halaman Optimasi Pupuk.....	49
5.4.3 Implementasi Antarmuka Halaman Data.....	50
BAB 6 PENGUJIAN DAN ANALISIS.....	52
6.1 Pengujian	52
6.2 Hasil Pengujian.....	52
6.2.1 Hasil Pengujian Terhadap Jumlah Partikel Terbaik.....	52
6.2.2 Hasil Pengujian Terhadap Jumlah Iterasi Terbaik	53

6.2.3 Hasil Pengujian Batas Inersia Terbaik	53
6.2.4 Hasil Pengujian Konvergensi	54
6.3 Analisis Hasil Pengujian.....	55
6.3.1 Analisis Hasil Pengujian Jumlah Partikel Terbaik	55
6.3.2 Analisis Hasil Pengujian Jumlah Iterasi Terbaik	56
6.3.3 Analisis Hasil Pengujian Jumlah Bobot Inersia	56
6.3.4 Analisis Hasil Pengujian Konvergensi	56
6.3.5 Analisis Keseluruhan Hasil Pengujian.....	57
BAB 7 PENUTUP	59
7.1 Kesimpulan.....	59
7.2 Saran	59
DAFTAR PUSTAKA.....	61



DAFTAR TABEL

Tabel 2.1 Tabel Perbandingan Objek dan Metode	5
Tabel 2.2 Tabel Rekomendasi Pemupukan Kabupaten Bojonegoro.....	11
Tabel 2.3 Tabel Rekomendasi Pemupukan Kabupaten Lamongan.....	12
Tabel 2.4 Tabel Rekomendasi Pemupukan Kabupaten Gresik	13
Tabel 2.5 Tabel Rekomendasi Pemupukan Kabupaten Madiun	13
Tabel 2.6 Tabel Rekomendasi Pemupukan Kabupaten Ngawi	14
Tabel 2.7 Tabel Rekomendasi Pemupukan Kabupaten Ngawi	14
Tabel 4.1 Tabel Interval Pupuk Tanaman Tumpangsari.....	24
Tabel 4.2 Tabel Harga Pupuk	26
Tabel 4.3 Representasi Partikel Ke-1 Pada Populasi Awal	32
Tabel 4.4 Representasi Seluruh Partikel Populasi Awal	32
Tabel 4.5 Rekomendasi Pemupukan Kecamatan Babat	33
Tabel 4.6 Hasil Evaluasi Partikel Pada Populasi Awal	34
Tabel 4.7 Partikel Ke-1 Pada Populasi Awal Dengan <i>Fitness</i> -nya	34
Tabel 4.8 Partikel <i>pBest</i>	35
Tabel 4.9 Partikel <i>gBest</i>	35
Tabel 4.10 Kecepatan Partikel.....	36
Tabel 4.11 Posisi Terbaru Partikel ke-1.....	37
Tabel 4.12 Representasi Seluruh Partikel Populasi Awal	37
Tabel 4.13 Rancangan Pengujian Jumlah Partikel.....	42
Tabel 4.14 Rancangan Pengujian Jumlah Iterasi.....	42
Tabel 4.15 Rancangan Pengujian Batas Bobot Inersia	43
Tabel 4.16 Rancangan Pengujian Konvergensi	43
Tabel 6.1 Hasil Pengujian Jumlah Partikel Terbaik.....	52
Tabel 6.2 Hasil Pengujian Jumlah Iterasi Terbaik.....	53
Tabel 6.3 Hasil Pengujian Batas Inersia Terbaik.....	54
Tabel 6.4 Data Daerah yang Diuji.....	54
Tabel 6.5 Hasil Pengujian Konvergensi	54
Tabel 6.6 Perbandingan Komposisi Pupuk oleh Pakar dan Sistem	57

Tabel 6.7 Perbandingan Komposisi Pupuk oleh Petani dan Sistem 58



DAFTAR PERSAMAAN

Persamaan (2.1) Persamaan Evaluasi Nilai <i>Fitness</i>	16
Persamaan (2.2) Fungsi Memperbarui Kecepatan	17
Persamaan (2.3) Fungsi Memperbarui Posisi	17
Persamaan (2.4) Fungsi <i>Time Varying Inertia Weight (TVIW)</i>	17
Persamaan (2.5) Fungsi <i>Time Varying Acceleration Coefficient (TVAC)</i>	18
Persamaan (4.1) Modifikasi Persamaan Evaluasi Nilai <i>Fitness</i>	26

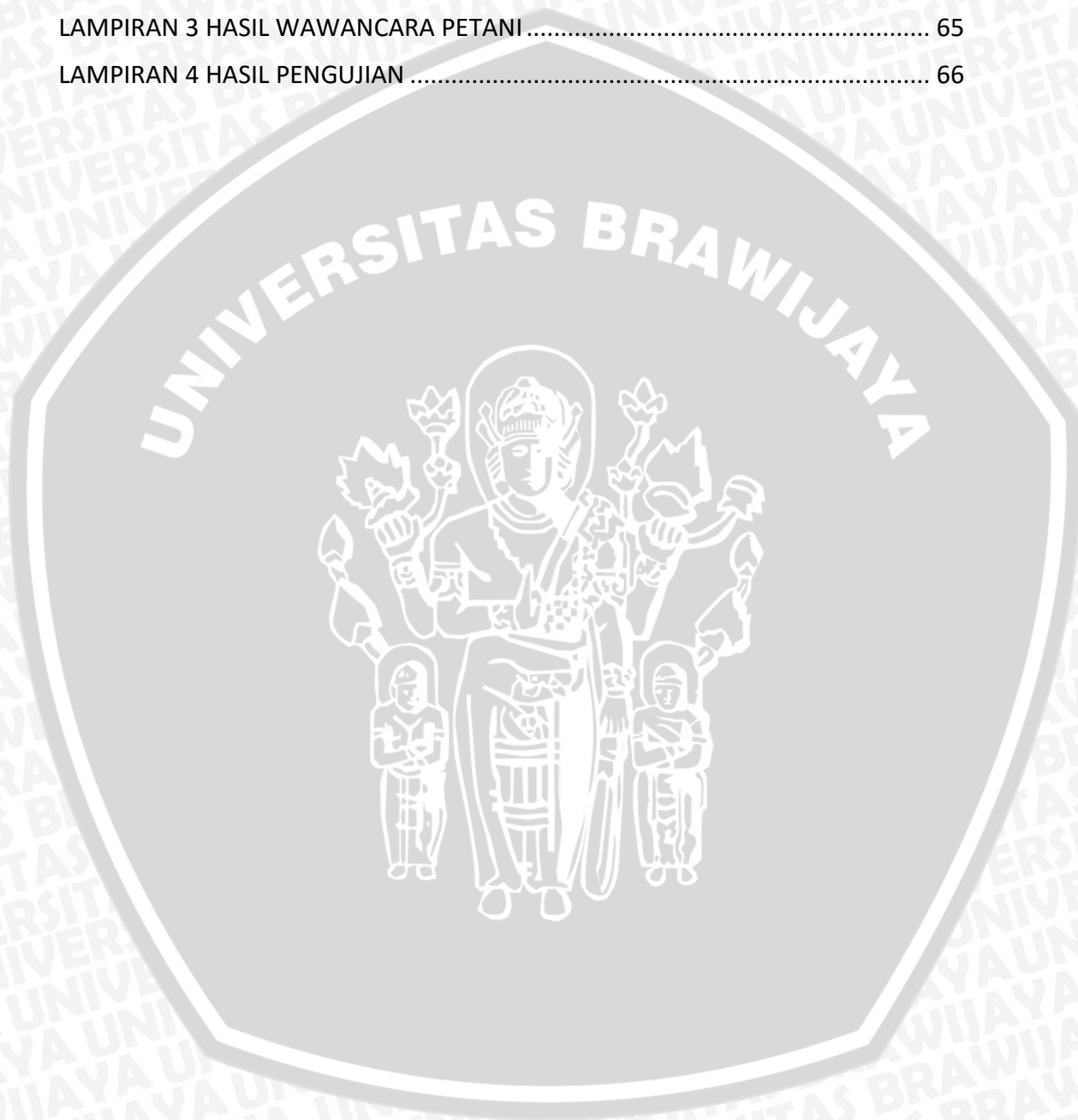


DAFTAR GAMBAR

Gambar 2.1 Pola Tanam Tumpangsari Pada Tanaman Ubikayu	6
Gambar 2.2 Tumpangsari Jagung dan Kedelai	8
Gambar 2.3 Pupuk Buatan	9
Gambar 2.4 Pupuk Cair	9
Gambar 2.5 Pupuk Daun	10
Gambar 2.6 Pupuk Dengan Reaksi Fisiologis Masam	10
Gambar 2.7 Pupuk NPK	11
Gambar 3.1 Tahapan Penelitian	19
Gambar 4.1 Alur Kerja Sistem	22
Gambar 4.2 Proses Perhitungan Optimasi Komposisi Pupuk Menggunakan <i>Particle Swarm Optimization</i>	23
Gambar 4.3 Kombinasi Pembentukan Partikel	24
Gambar 4.4 Inisialisasi Partikel	25
Gambar 4.5 Proses Pencarian <i>pBest</i>	27
Gambar 4.6 Proses Pencarian <i>gBest</i>	28
Gambar 4.7 Proses Memperbarui Kecepatan	29
Gambar 4.8 Proses Memperbarui Posisi	30
Gambar 4.9 Rancangan Antarmuka Halaman Beranda	38
Gambar 4.10 Rancangan Antarmuka Halaman Optimasi Pupuk	39
Gambar 4.11 Rancangan Antarmuka Halaman Beranda	41
Gambar 5.1 Implementasi Antarmuka Halaman Beranda	49
Gambar 5.2 Implementasi Antarmuka Halaman Optimasi Pupuk	50
Gambar 5.3 Implementasi Antarmuka Halaman Data	51
Gambar 6.1 Analisis Pengujian Jumlah Partikel	55
Gambar 6.2 Analisis Pengujian Jumlah Iterasi Terbaik	56
Gambar 6.3 Analisis Pengujian Konvergensi	57

DAFTAR LAMPIRAN

LAMPIRAN 1 SURAT IZIN PENELITIAN	63
LAMPIRAN 2 HASIL WAWANCARA PAKAR	64
LAMPIRAN 3 HASIL WAWANCARA PETANI	65
LAMPIRAN 4 HASIL PENGUJIAN	66



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan negara beriklim tropis dengan jumlah pulau mencapai 17.504 sehingga memiliki wilayah yang cukup luas dan variasi agroklimat yang beragam. Menurut Badan Pusat Statistik, pada tahun 2013 Indonesia memiliki luas lahan sawah lebih dari 8,1 juta Hektar. Selama ini masih banyak permasalahan dalam usaha peningkatan produksi pertanian (Indriati, 2009), di antaranya adalah penggunaan pupuk anorganik dan pestisida secara tidak terkontrol dapat menyebabkan terjadinya pencemaran lingkungan. Di samping dibutuhkan biaya usaha tani yang tinggi, konversi lahan pertanian menjadi kompleks perumahan, kawasan industri, kawasan perdagangan dan berbagai sarana publik lain berakibat terhadap semakin sempitnya lahan pertanian produktif. Sistem pertanian secara monokultur atau terpaku pada satu jenis tanaman yang digunakan pada usaha tani dapat menyebabkan berkurangnya unsur-unsur hara tertentu yang dapat menyebabkan defisiensi atau kerusakan pada lahan tanah pertanian.

Salah satu solusi untuk mengatasi permasalahan di atas adalah penerapan sistem tumpangsari. Pola tanam ganda atau tumpangsari merupakan sistem pengelolaan lahan pertanian dengan mengkombinasikan penanaman beberapa jenis tanaman dalam satu masa tanam. Dengan penanaman lebih dari satu tanaman dapat meningkatkan hasil produksi pertanian dengan luas lahan yang lebih sempit. Selain itu sistem penanaman tumpangsari dapat meningkatkan efektivitas dalam penggunaan cahaya, air serta unsur hara, di samping dapat menekan pertumbuhan gulma. Sistem tumpangsari lebih menguntungkan baik secara agronomis maupun ekonomis (Indriati, 2009). Untuk mengevaluasi manfaat yang ditimbulkan dari pola tanam tumpangsari dapat dilihat dari hasil perhitungan Nilai Kesetaraan Lahan (NKL) yang bernilai lebih dari 1 (Prasetyo, 2009).

Dalam pola tanam tumpangsari terdapat banyak faktor yang harus dipertimbangkan di antaranya kesesuaian masa tanam, jarak antar tanaman, pola, serta jumlah populasi tanaman perlu diperhatikan agar tidak terjadi persaingan antar tanaman. Selain itu, untuk meningkatkan keberhasilan penerapan sistem tumpangsari diperlukan penggunaan dosis pupuk yang dapat memenuhi kebutuhan tanaman. Hal ini diperlukan agar tidak terdapat persaingan antar tanaman yang dapat mempengaruhi hasil panen. Sedangkan komposisi pupuk yang berlebihan dapat menyebabkan masalah biaya yang dikeluarkan petani menjadi terlalu besar. Oleh karena itu dibutuhkan teknologi yang dapat memudahkan petani untuk mendapatkan komposisi pupuk yang dapat memenuhi tanaman tumpangsari dengan biaya seminimal mungkin.

Berdasarkan permasalahan di atas pada penelitian ini, didesain sebuah sistem untuk optimasi campuran pupuk untuk sistem penanaman tumpangsari dengan menggunakan algoritma *Particle Swarm Optimization* (PSO). Algoritma *Particle Swarm Optimization* (PSO) merupakan sebuah teknik optimasi stokastik

berdasarkan populasi yang diinspirasi oleh perilaku sosial kawanan burung atau sekelompok ikan dalam mencari makanan. Algoritma *Particle Swarm Optimization* (PSO) pertama kali dikembangkan pada tahun 1995 oleh Dr. Eberhart dan Dr. Kennedy (Krusienski dan Jenkins, 2006). Dibandingkan dengan algoritma evolusioner lainnya seperti Algoritma Genetika, *Particle Swarm Optimization* memiliki beberapa kelebihan, antara lain mudah diimplementasikan dan membutuhkan parameter yang lebih sedikit (Wati, 2013), serta lebih efisien dimana untuk menyelesaikan suatu permasalahan PSO membutuhkan lebih sedikit komputasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas didapat rumusan masalah sebagai berikut:

1. Bagaimana mengimplementasikan *Particle Swarm Optimization* untuk optimasi komposisi pupuk yang digunakan pada tanaman tumpangsari.
2. Bagaimana parameter yang digunakan untuk optimasi komposisi pupuk untuk tanaman tumpangsari menggunakan algoritma *Particle Swarm Optimization* sehingga dapat menghasilkan hasil rekomendasi yang optimal.
3. Bagaimana kualitas dari rekomendasi hasil optimasi komposisi pupuk untuk tanaman tumpangsari menggunakan algoritma *Particle Swarm Optimization*.

1.3 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

1. Mengimplementasikan algoritma *Particle Swarm Optimization* untuk membuat sistem cerdas untuk rekomendasi komposisi pupuk tanaman tumpangsari.
2. Mendapatkan parameter optimal untuk optimasi komposisi pupuk untuk tanaman tumpangsari menggunakan algoritma *Particle Swarm Optimization*.
3. Mendapatkan nilai evaluasi dari hasil rekomendasi komposisi pupuk tanaman tumpangsari menggunakan algoritma *Particle Swarm Optimization*.

1.4 Manfaat

Manfaat yang diperoleh dari penelitian ini adalah sebagai berikut:

1. Membantu petani untuk menentukan komposisi pupuk yang tepat sehingga mengoptimalkan pertumbuhan tanaman tumpangsari.
2. Memaksimalkan keuntungan petani dengan cara mengurangi biaya pemupukan.

1.5 Batasan Masalah

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

1. Jenis tanaman yang digunakan sebagai objek penelitian merupakan tanaman pangan yaitu jagung dan kacang kedelai.
2. Pupuk yang digunakan pada penelitian ini adalah pupuk Urea, pupuk SP-36 dan pupuk KCl.

3. Data rekomendasi pupuk yang digunakan pada penelitian ini adalah data rekomendasi pupuk tanaman jagung dan kedelai pada enam kabupaten/kota di Propinsi Jawa Timur yaitu Bojonegoro, Lamongan, Gresik, Madiun, Ngawi dan Tulungagung.

1.6 Sistematika Pembahasan

Penelitian ini disusun berdasarkan sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini membahas tentang alasan yang menjadi latar belakang penelitian, rumusan masalah yang dihadapi, batasan permasalahan yang diselesaikan, tujuan dan manfaat penelitian, serta sistematika pembahasan.

BAB II LANDASAN KEPUSTAKAAN

Pada bab ini membahas tentang beberapa tinjauan pustaka dan dasar teori pendukung yang berkaitan dengan *Particle Swarm Optimization*, metode tanam tumpangsari, nutrisi pupuk tanaman serta teori untuk menyelesaikan permasalahan ini.

BAB III METODOLOGI

Pada bab ini membahas tentang metode atau tahapan penelitian yang meliputi studi literatur dan penyusunan dasar teori, analisis serta perancangan perangkat lunak, implementasi, pengujian dan analisis sistem dan pengambilan kesimpulan dan saran.

BAB IV PERANCANGAN

Pada bab ini membahas tentang perancangan sistem optimasi komposisi pupuk untuk sistem penanaman tumpangsari menggunakan algoritma *Particle Swarm Optimization* secara terperinci.

BAB IV IMPLEMENTASI

Pada bab ini membahas tentang implementasi dari sistem optimasi komposisi pupuk untuk sistem penanaman tumpangsari menggunakan algoritma *Particle Swarm Optimization* sesuai dengan perancangan sistem pada bab sebelumnya.

BAB VI PENGUJIAN DAN ANALISIS

Pada bab ini membahas tentang tahapan pengujian serta analisis hasil pengujian terhadap implementasi sistem optimasi komposisi pupuk untuk sistem penanaman tumpangsari menggunakan algoritma *Particle Swarm Optimization* yang telah direalisasikan.

BAB VII PENUTUP

Pada bab ini memaparkan kesimpulan yang diperoleh dari hasil implementasi serta pengujian perangkat lunak yang dikembangkan dan saran yang diperoleh dari penelitian untuk pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini berisi pemaparan landasan kepustakaan yang meliputi kajian pustaka dan dasar teori yang diperlukan untuk penelitian. Kajian pustaka adalah membahas penelitian yang telah ada dan yang diuraikan sebelumnya. Dasar teori membahas teori yang diperlukan untuk menyusun penelitian yang diusulkan.

2.1 Kajian Pustaka

Algoritma Genetika telah banyak diterapkan dalam berbagai penelitian sebagai solusi untuk membantu menyelesaikan permasalahan optimasi. Salah satunya adalah penelitian yang berjudul "*Genetic Algorithm for Bus Frequency Optimization*". Pada penelitian ini masalah frekuensi bus dipecah menjadi dua bagian yaitu waktu tunggu dan lama penumpang naik bus, yang kemudian keduanya diintegrasikan sehingga menjadi satu program. Untuk membuat sistem lebih sederhana diasumsikan bahwa banyaknya penumpang yang menunggu pada tujuan pemberhentian pertama adalah 0. Seleksi *Roulette Wheel* untuk menentukan kromosom, sedangkan pada tahap seleksi digunakan metode *Elitism* untuk menentukan kromosom yang akan digunakan generasi selanjutnya. Iterasi dilakukan hingga total waktu pergi tidak atau hampir tidak berubah dimana telah didapat solusi yang optimal.

Pada penelitian dengan judul "Model Penjadwalan Matakuliah Secara Otomatis Berbasis Algoritma *Particle Swarm Optimization* (PSO)". Penelitian tersebut bertujuan untuk membuat suatu sistem cerdas berbasis komputer menggunakan metode *Particle Swarm Optimization* (PSO) untuk penentuan jadwal matakuliah dengan cara mengoptimalkan berbagai sumber daya pembelajaran yaitu pengajar/dosen, ruang kelas, mahasiswa dan memenuhi peraturan yang berlaku pada sebuah kurikulum (Wati dan Rochman, 2013). Proses PSO pada penelitian ini diawali dengan inisialisasi populasi partikel. Kemudian, dilakukan perhitungan untuk mencari nilai *cost* semua partikel dan mencari *pBest* serta *gBest*. Jika kondisi berhenti tercapai maka sistem menampilkan hasil optimal dari perhitungan, jika belum tercapai sistem melakukan perhitungan kecepatan dan posisi yang baru untuk masing-masing partikel dan mengulang proses mulai dari perhitungan untuk mencari nilai *cost* semua partikel. Sedangkan untuk permasalahan optimasi pemupukan sebelumnya telah dilakukan penelitian dengan menggabungkan AHP dan Algoritma Genetika pada "Implementasi Analytical Hierarchy Process (AHP) dan Algoritma Genetika untuk Rekomendasi dan Optimasi Pemupukan Berimbang Tanaman Hortikultura".

Berdasarkan tinjauan pustaka di atas, peneliti akan melakukan implementasi metode *Particle Swarm Optimization* (PSO) untuk melakukan optimasi komposisi pupuk untuk mencapai tingkat efisiensi biaya yang diinginkan. Perbandingan objek dan metode penelitian yang akan dilakukan dengan penelitian yang telah dilakukan ditunjukkan oleh Tabel 2.1 berikut ini.

Tabel 2.1 Tabel Perbandingan Objek dan Metode

No	Judul	Objek	Metode	Hasil
1	<i>Tasks Scheduling in Computational Grid using a Hybrid Discrete Particle Swarm Optimization</i>	Penjadwalan matakuliah	<i>Particle Swarm Optimization (PSO)</i>	Fitness = (λ makespan + (1- λ) mean_flowtime + Penalty)-1 Dimana $\lambda = 0.7$
2	Implementasi Analytical Hierarchy Process (AHP) dan Alogaritma Genetika untuk Rekomendasi dan Optimasi Pemupukan Berimbang Tanaman Hortikultura	Jenis pupuk dan dosis pupuk	AHP dan Algoritma Genetika	Ditemukan <i>fitness</i> terbesar ada pada populasi sebesar 80, ukuran generasi sebesar 800, nilai cr 0,2 dan mr 0,8, dengan nilai α antara -1 sampai 1 dan nilai r pada rentang nilai -1 dan 1.
3	Optimasi Penjadwalan Mata Kuliah Di Jurusan Teknik Informatika PENS dengan Menggunakan Alogartitma <i>Particle Swarm Optimization (PSO)</i>	Dosen dan jadwal mata kuliah	<i>Particle Swarm Optimization (PSO)</i>	1. Jumlah partikel yang dibangkitkan mempengaruhi optimasi. 2. Inertia factor (w)=0.5 dapat menghasilkan rata-rata jadwal yang lebih optimal.
4	Usulan Penulis: Optimasi Komposisi Pupuk Untuk Sistem Penanaman Tumpangsari Menggunakan Algoritma <i>Particle Swarm Optimization (PSO)</i>	Jenis pupuk dan dosis pupuk	<i>Particle Swarm Optimization (PSO)</i>	

2.2 Tumpangsari

Pada sub-bab ini membahas gambaran umum pola tanam tumpangsari dan keuntungan penggunaan pola tanam ini.

2.2.1 Pengertian Tumpangsari

Tumpangsari merupakan sistem pertanian dengan mengkombinasikan dua atau lebih jenis tanaman pada sebidang lahan pertanian dalam satu jangka waktu tertentu. Sistem ini merupakan sistem intensifikasi pertanian atau pengoptimalan hasil pertanian dengan menggunakan satu lahan dalam dimensi waktu dan ruang tertentu. Contoh sistem penanaman tumpangsari dapat dilihat pada Gambar 2.1 dibawah ini.



Gambar 2.1 Pola Tanam Tumpangsari Pada Tanaman Ubikayu

(Sumber: Badan Penyuluhan dan Pengembangan
Sumber Daya manusia Pertanian)

2.2.2 Jenis-jenis Tumpangsari

Terdapat beberapa macam metode penanaman tumpangsari, di antaranya adalah sebagai berikut (Sutedjo, 1989):

1. Tumpangsari campuran (*mixed intercropping*), yaitu dua atau lebih jenis tanaman secara serentak tanpa menggunakan pola atau pengaturan barisan yang jelas.
2. Tumpangsari barisan (*row intercropping*), yaitu penanaman dua atau lebih jenis tanaman secara serentak dalam barisan.
3. Tumpangsari lajur (*strip intercropping*), yaitu penanaman dua atau lebih jenis tanaman secara serentak dalam lajur yang berbeda yang cukup lebar sebagai penanaman bebas tetapi cukup sempit untuk tanaman dapat berinteraksi secara argonomi.
4. Tumpangsari sisipan (*relay intercropping*), yaitu penanaman dua atau lebih jenis tanaman secara serentak selama sebagian kehidupan tanaman. Tanaman kedua ditanam setelah tanaman pertama sudah mencapai masa reproduksi tetapi belum siap panen.

2.2.3 Kelebihan Sistem Pertanian Tumpangsari

Berikut merupakan kelebihan yang dimiliki sistem penanaman tumpangsari.

1. Menekan serangan hama dan penyakit
Penanaman tumpangsari merupakan penanaman dengan menggunakan lebih dari dua jenis tanaman pada satu lahan yang sama. Dengan menanamkan tanaman yang dapat mengusir hama atau penyakit tertentu pada sistem tanam tumpangsari akan menyebabkan berkurangnya serangan hama dan penyakit.
2. Efisien penggunaan ruang dan waktu
Pada pola tanam monokultur, terdapat sisa lahan kosong yang tidak dapat digunakan. Dengan penanaman tumpangsari, lahan kosong ini dapat dimanfaatkan untuk tanaman lainnya sehingga petani tidak perlu menunggu tanaman utama selesai menggunakan lahan.
3. Mencegah dan mengurangi pengangguran musim
Dengan menggunakan beberapa jenis tanaman yang memiliki jangka waktu atau musim tertentu untuk dapat tumbuh dengan baik. Hal ini menyebabkan petani manganggur untuk menunggu masa tanam berikutnya. Pada sistem penanaman tumpangsari, tanaman yang digunakan memiliki musim dan siklus tanam yang lebih beragam. Dengan perbedaan musim tanam antara tanaman satu dengan tanaman lainnya, petani dapat melakukan rotasi sehingga tidak perlu menganggur.
4. Memperbaiki sifat fisik tanah
Setiap jenis tanaman memiliki kebutuhan nutrisi unsur hara yang berbeda. Dengan adanya rotasi penanaman jenis tanaman, memungkinkan tanah untuk mengembalikan unsur-unsur hara yang telah diserap oleh jenis tanaman sebelumnya.
5. Meningkatkan nilai tambah produktivitas hasil
Dengan penanaman tumpangsari, lahan kosong yang dimanfaatkan untuk tanaman lainnya dapat menjadi nilai tambah yang dapat meningkatkan keuntungan petani.

2.2.4 Tumpangsari Jagung dan Kedelai

Tumpangsari kedelai dan jagung dapat dilakukan dengan menggunakan sistem penanaman tumpangsari lajur (*strip intercropping*) dimana tanaman kedelai ditanam pada sela-sela lajur tanaman jagung. Dalam tumpangsari kedelai dan jagung, harus memperhatikan waktu tanam antara kedelai dan jagung serta pengaturan jarak antara tanaman jagung dan kedelai. Jarak tanam terlalu sempit akan menyebabkan terjadinya kompetisi unsur hara, cahaya matahari dan air, yang dapat mempengaruhi pertumbuhan serta hasil panen kedua tanaman.

Fisik dari tanaman kedelai yang lebih pendek dari tanaman jagung menyebabkan tanaman kedelai memerlukan ruang tumbuh yang lebih sedikit. Menurut Balai Penelitian Tanaman Aneka Kacang dan Umbi, jarak ideal yang dibutuhkan oleh tanaman kedelai adalah 40 cm x 15 cm sedangkan jarak ideal yang dibutuhkan oleh tanaman jagung adalah 75 cm x 40 cm. Dengan ketentuan

tersebut, bila jarak tanam jagung tumpangsari dibuat 150 cm x 40 cm, maka setidaknya akan terdapat 6 sampai 8 tanaman kedelai pada satu jarak tanam jagung seperti pada Gambar 2.2. Penanaman kedelai 6 baris dalam satu jarak tanam jagung dapat memberikan hasil lebih banyak dibanding jumlah baris kedelai kurang dari 6 dengan nilai kesetaraan lahan (NKL) > 1 (Putri, 2011; Aminah dkk, 2014).

0	0	0	0	0	0
X	X	X	X	X	X
X	X	X	X	X	X
X	X	X	X	X	X
X	X	X	X	X	X
X	X	X	X	X	X
X	X	X	X	X	X
X	X	X	X	X	X
X	X	X	X	X	X
0	0	0	0	0	0
0 : Jagung			X : Kedelai		

Gambar 2.2 Tumpangsari Jagung dan Kedelai

2.3 Pupuk

Pupuk merupakan materi yang ditambahkan pada lahan atau media tanam untuk memenuhi kebutuhan nutrisi tanaman sehingga tanaman tersebut mampu tumbuh dan berproduksi secara optimal. Material pupuk dapat berupa bahan organik ataupun non-organik. Unsur hara yang terkandung dalam pupuk di antaranya adalah nitrogen (N), fosfat (P), kalsium (Ca), kalium (K), seng (Zn), tembaga (Cu), belerang (S) dan besi (Fe) (Soemarno, 2013).

Komposisi pupuk yang digunakan tergantung dari jenis tumbuhan dan jenis tanah yang digunakan. Hal ini disebabkan oleh perbedaan karakteristik dan susunan kimia tanah. Pemupukan yang benar harus mempertimbangkan kemampuan tanah dalam menyediakan unsur hara dan kebutuhan hara tanaman itu sendiri sehingga hasil produksi meningkat tanpa merusak lahan atau media tanam akibat pemupukan yang berlebihan (Soemarno, 2013).

2.3.1 Macam-Macam Pupuk

Pupuk dapat dibedakan berdasarkan bahan asal, senyawa, fasa, cara penggunaan, reaksi fisiologi, jumlah dan macam hara yang dikandungnya (Sumarno, 2013). Berdasarkan asalnya pupuk dibedakan menjadi:

1. Pupuk buatan, yaitu pupuk yang dibuat menggunakan senyawa kimia oleh pabrik. Pupuk buatan digolongkan berdasarkan unsur hara yang dikandungnya, sifat kelarutannya dan sifat keasaman yang terjadi dalam penggunaannya. Pupuk buatan yang sering dipakai adalah TSP, Urea, Rustika dan Nitrophonska. Pupuk ini dibuat oleh pabrik dengan mengubah sumber daya alam melalui proses fisika dan/atau kimia.

2. Pupuk alami, yaitu pupuk yang dibuat dengan bahan alam tanpa proses yang berarti. Misalnya pupuk kompos, pupuk kandang guano, pupuk hijau dan pupuk batuan fosfat.



Gambar 2.3 Pupuk Buatan

Sumber: PT. Peteokimia Gresik

Berdasarkan senyawanya, pupuk dibedakan menjadi:

1. Pupuk organik, yaitu pupuk yang berupa senyawa organik. Kebanyakan pupuk alam tergolong pupuk organik adalah pupuk kandang, kompos, guano.
2. Pupuk anorganik, yaitu pupuk dalam bentuk senyawa kimia yang mudah terlarut. Pupuk anorganik memberikan nutrisi yang langsung terlarut ke tanah dan siap diserap tumbuhan tanpa memerlukan proses pelapukan.

Berdasarkan bentuknya, pupuk dibedakan menjadi:

1. Pupuk padat, pupuk ini umumnya mempunyai kelarutan yang beragam mulai dari yang mudah larut air sampai yang sukar larut dalam air.
2. Pupuk cair, pupuk ini berupa cairan dan cara penggunaannya dilarutkan dulu dengan air.



Gambar 2.4 Pupuk Cair

Sumber: PT. Mitra Alam Selaras

Berdasarkan cara penggunaannya, pupuk dibedakan menjadi:

1. Pupuk daun, yaitu pupuk yang cara pemupukannya dengan dilarutkan dalam air kemudian disemprotkan ke permukaan daun.
2. Pupuk akar, merupakan pupuk yang diberikan ke dalam tanah di sekitar akar agar diserap oleh akar tanaman.



Gambar 2.5 Pupuk Daun

Sumber: PT. Gresik Cipta Sejahtera

Berdasarkan reaksi fisiologisnya, pupuk dibedakan menjadi:

1. Pupuk yang mempunyai reaksi fisiologis masam, artinya bila pupuk tersebut diberikan ke dalam tanah ada kecenderungan tanah menjadi lebih asam. Misalnya adalah pupuk KCL dan Urea.
2. Pupuk yang memiliki reaksi fisiologis basa adalah pupuk yang jika diberikan ke dalam tanah cenderung menyebabkan pH tanah meningkat. Contohnya adalah pupuk *chili saltpeter*, *calnitro* dan kalsium sianida.



Gambar 2.6 Pupuk Dengan Reaksi Fisiologis Masam

Sumber: PT. Gresik Cipta Sejahtera

Berdasarkan jumlah hara yang dikandungnya, pupuk dibedakan menjadi:

1. Pupuk tunggal, merupakan pupuk yang hanya mengandung satu hara tanaman saja.
2. Pupuk majemuk, merupakan pupuk yang mengandung dua atau lebih unsur yang diperlukan.



Gambar 2.7 Pupuk NPK

Sumber: PT. Petrokimia Gresik

2.3.2 Pemupukan Provinsi Jawa Timur

Berikut merupakan rekomendasi pemupukan tanaman jagung dan kacang kedelai oleh pakar pada 6 kabupaten di Jawa Timur yaitu kabupaten Bojonegoro, Lamongan, Gresik, Madiun, Ngawi dan Tulungagung.

Tabel 2.2 Tabel Rekomendasi Pemupukan Kabupaten Bojonegoro

Kecamatan	Pupuk Tunggal (kg/ha)					
	Jagung			Kedelai		
	Urea	SP-36	KCl	Urea	SP-36	KCl
Balen	350	125	75	50	75	75
Baureno	350	100	75	50	50	75
Bojonegoro	350	100	75	50	50	75
Bubulan	350	100	100	50	50	100
Dander	350	100	75	50	50	75
Gayam	350	125	75	50	75	75
Gondang	350	125	75	50	75	75
Kalitidu	350	100	100	50	50	100
Kanor	350	100	75	50	50	75
Kapas	350	100	75	50	50	75
Kasiman	350	125	75	50	75	75
Kedewan	350	125	75	50	75	75
Kedungadem	350	100	50	50	50	50

Kepoh Baru	350	125	75	50	75	75
Malo	350	125	75	50	75	75
Margomulyo	350	125	75	50	75	75
Ngambon	350	150	75	50	100	75
Ngasem	350	125	50	50	75	50
Ngraho	350	125	50	50	75	50
Padangan	350	125	50	50	75	50
Purwosari	350	125	50	50	75	50
Sekar	350	100	50	50	50	50
Sugihwaras	350	100	50	50	50	50
Sukosewu	350	100	75	50	50	75
Sumberejo	350	100	50	50	50	50
Tambakrejo	350	125	50	50	75	50
Temayang	350	100	50	50	50	50
Trucuk	350	125	50	50	75	50

Sumber: Badan Pengkajian Teknologi Pertanian, 2014.

Tabel 2.3 Tabel Rekomendasi Pemupukan Kabupaten Lamongan

Kecamatan	Pupuk Tunggal (kg/ha)					
	Jagung			Kedelai		
	Urea	SP-36	KCl	Urea	SP-36	KCl
Babat	350	125	75	50	75	75
Bluluk	350	125	75	50	75	75
Brondong	350	150	75	50	100	75
Deket	350	100	50	50	50	50
Gragah	350	100	50	50	50	50
Kalitengah	350	150	50	50	100	50
Karang Geneng	350	150	75	50	100	75
Karangbinangun	350	100	50	50	50	50
Kedungpring	350	100	75	50	50	75
Kembangbahu	350	125	75	50	75	75
Lamongan	350	125	75	50	75	75
Laren	350	125	75	50	75	75
Maduran	350	125	75	50	75	75
Mantup	350	125	50	50	75	50
Modo	350	100	75	50	50	75
Ngimbang	350	125	75	50	75	75
Paciran	350	150	75	50	75	75
Pucuk	350	125	75	50	75	75
Sambeng	350	125	75	50	75	50
Sarirejo	350	100	50	50	50	75
Sekaran	350	125	75	50	75	75
Solokuro	350	150	75	50	100	75

Sugio	350	100	75	50	50	75
Sukodadi	350	100	75	50	50	75
Sukorame	350	125	75	50	75	75
Tikung	350	125	75	50	75	75
Turi	350	150	75	50	100	75

Sumber: Badan Pengkajian Teknologi Pertanian, 2014.

Tabel 2.4 Tabel Rekomendasi Pemupukan Kabupaten Gresik

Kecamatan	Pupuk Tunggal (kg/ha)					
	Jagung			Kedelai		
	Urea	SP-36	KCl	Urea	SP-36	KCl
Balongpanggang	350	125	50	50	75	50
Benjeng	350	125	50	50	75	50
Bunga	350	100	75	50	50	75
Cerme	350	125	75	50	75	75
Driyorejo	350	125	75	50	75	75
Duduksampeyan	350	125	50	50	75	50
Dukun	350	100	50	50	50	50
Kebomas	350	125	50	50	75	50
Kedamean	350	125	75	50	75	75
Manyar	350	125	50	50	75	50
Menganti	350	125	50	50	75	50
Panceng	350	150	50	50	100	50
Sangkapura	350	125	75	50	75	75
Sidayu	350	100	75	50	50	75
Tambak	350	125	75	50	75	75
Ujungpangkah	350	100	75	50	50	75
Wringinanom	350	125	75	50	75	75

Sumber: Badan Pengkajian Teknologi Pertanian, 2014.

Tabel 2.5 Tabel Rekomendasi Pemupukan Kabupaten Madiun

Kecamatan	Pupuk Tunggal (kg/ha)					
	Jagung			Kedelai		
	Urea	SP-36	KCl	Urea	SP-36	KCl
Balerejo	350	125	75	50	75	75
Dagangan	350	100	100	50	50	100
Dolopo	350	125	100	50	75	100
Geger	350	100	100	50	50	100
Gemarang	350	125	75	50	75	75
Jiwan	350	125	75	50	75	75
Kare	350	125	100	50	75	100
Kebonsari	350	125	100	50	75	100
Madiun	350	150	50	50	100	50

Mejayan	350	125	50	50	75	50
Pilangkenceng	350	125	50	50	75	50
Saradan	350	125	50	50	75	50
Sawahan	350	125	75	50	75	75
Wonosari	350	125	100	50	75	100
Wungu	350	100	100	50	50	100

Sumber: Badan Pengkajian Teknologi Pertanian, 2014.

Tabel 2.6 Tabel Rekomendasi Pemupukan Kabupaten Ngawi

Kecamatan	Pupuk Tunggal (kg/ha)					
	Jagung			Kedelai		
	Urea	SP-36	KCl	Urea	SP-36	KCl
Bringin	350	125	75	50	75	75
Geneng	350	100	75	50	50	75
Gerih	350	100	75	50	50	75
Jogorogo	350	100	100	50	50	100
Karanganyar	350	100	75	50	50	75
Karangjati	350	125	75	50	75	75
Kasreman	350	125	75	50	75	75
Kedungalar	350	100	100	50	50	100
Kendal	350	100	75	50	50	75
Kwadungan	350	100	75	50	50	75
Mantingan	350	125	75	50	75	75
Ngawi	350	125	75	50	75	75
Ngrambe	350	100	50	50	50	50
Padas	350	125	75	50	75	75
Pangkur	350	125	75	50	75	75
Paron	350	125	75	50	75	75
Pitu	350	100	50	50	50	50
Sine	350	125	75	50	75	75
Widodaren	350	125	75	50	75	75

Sumber: Badan Pengkajian Teknologi Pertanian, 2014.

Tabel 2.7 Tabel Rekomendasi Pemupukan Kabupaten Ngawi

Kecamatan	Pupuk Tunggal (kg/ha)					
	Jagung			Kedelai		
	Urea	SP-36	KCl	Urea	SP-36	KCl
Bandung	350	125	75	50	75	75
Besuki	350	125	75	50	75	75
Boyolangu	350	100	100	50	50	100
Campur Darat	350	100	100	50	50	75
Gondang	350	125	75	50	75	100
Kalidawir	350	125	100	50	75	100

Karangrejo	350	125	75	50	75	75
Kauman	350	125	75	50	75	75
Kedungwaru	350	125	75	50	75	75
Ngantru	350	100	75	50	50	75
Ngunut	350	125	75	50	75	75
Pager Wojo	350	100	75	50	50	75
Pakel	350	100	100	50	50	100
Pucang Laban	350	125	100	50	75	100
Rejotangan	350	125	75	50	75	75
Sendang	350	100	75	50	50	75
Sumbergempol	350	125	100	50	75	100
Tulungagung	350	125	75	50	75	75

Sumber: Badan Pengkajian Teknologi Pertanian, 2014.

2.4 Particle Swarm Optimization

Algoritma *Particle Swarm Optimization* (PSO) merupakan sebuah teknik optimasi berdasarkan populasi yang diinspirasi oleh perilaku sosial kawanan burung atau sekelompok ikan dalam mencari makanan. Algoritma *Particle Swarm Optimization* (PSO) pertama kali dikembangkan pada tahun 1995 oleh Dr. Eberhart dan Dr. Kennedy (Krusienski dan Jenkins, 2006). Dibandingkan dengan Algoritma Genetika, *Particle Swarm Optimization* memiliki beberapa kelebihan, antara lain mudah diimplementasikan dan parameter yang harus ditentukan lebih sedikit dan lebih efisien (Wati dkk, 2013). *Particle Swarm Optimization* dapat digunakan dalam menyelesaikan permasalahan non linier, pencarian dengan ruang yang besar dan dapat memberikan hasil yang lebih baik dengan lebih efisien.

Particle Swarm Optimization memiliki beberapa kesamaan dengan teknik komputasi evolusioner seperti Algoritma Genetika, dimana terdapat individu yang merupakan representasi dari sebuah solusi yang dibangkitkan secara acak dari kemungkinan solusi yang ada. Tidak seperti Algoritma Genetika, pada *Particle Swarm Optimization* setiap partikel tidak digantikan oleh partikel baru. Setiap partikel menyesuaikan pergerakannya sesuai dengan pengalamannya sendiri serta pengalaman tetangganya.

Langkah-langkah Algoritma *Particle Swarm Optimization* (PSO) dalam menyelesaikan suatu permasalahan adalah sebagai berikut:

1. Inisialisasi partikel serta membangkitkan kecepatan awal.
2. Evaluasi partikel/menghitung nilai *fitness*.
3. Mencari partikel lokal terbaik atau *pBest*.
4. Mencari partikel global terbaik atau *gBest*.
5. Memperbarui kecepatan partikel.
6. Memperbarui posisi partikel.
7. Bila kondisi berhenti belum terpenuhi mengulang dari langkah ke-2.

2.4.1 Inisialisasi Partikel Serta Membangkitkan Kecepatan Awal

Inisialisasi populasi partikel dilakukan dengan membangkitkan populasi partikel secara acak sebanyak jumlah partikel atau p , dimana setiap partikel berisi variable atau dimensi sebanyak d . Jadi populasi awal berupa matriks dengan ukuran $p \times d$. Selanjutnya dilakukan pembangkitan nilai kecepatan partikel dengan ukuran yang sama dengan ukuran partikel yaitu $p \times d$ dengan nilai 0.

2.4.2 Evaluasi Partikel / Menghitung Nilai *Fitness*

Perhitungan nilai *fitness* semua partikel diawali dengan melakukan proses dekoding partikel ke dalam domain variabel $x_1, \dots, x_d$. Setelah itu nilai variabel setiap partikel dievaluasi oleh fungsi yang telah ditentukan $f(x_1, \dots, x_d)$. Persamaan yang digunakan untuk menilai kualitas atau menghitung nilai *fitness* didapat dari penelitian yang sudah dilakukan sebelumnya dengan objek yang sama yaitu optimasi komposisi pupuk yaitu pada penelitian yang berjudul "Implementasi *Analytical Hierarchy Process* (AHP) dan Alogaritma Genetika untuk Rekomendasi dan Optimasi Pemupukan Berimbang Tanaman Hortikultura". Di mana pada penelitian tersebut untuk melakukan evaluasi dari partikel hasil representasi komposisi pupuk digunakan persamaan sebagai berikut.

$$fitness = \frac{1}{(total\ harga + penalti)} \quad (2.1)$$

2.4.3 Mencari Partikel Lokal Terbaik atau *pBest*

Untuk menentukan *pBest* (*fitness* lokal terbaik) maka hasil evaluasi fungsi $f(x_1, \dots, x_d)$ diperoleh nilai *fitness* sebanyak jumlah partikel atau p yaitu $f_1, \dots, f_p$. Selanjutnya *pBest* ditentukan dengan cara membandingkan nilai elvaluasi dari setiap partikel dengan nilai evaluasi partikel tersebut pada iterasi sebelumnya. Jika nilai *fitness* dari sebuah partikel lebih baik dari nilai *fitness* sebelumnya maka *pBest* akan diperbarui.

2.4.4 Mencari Partikel Global Terbaik atau *gBest*

Untuk menentukan *gbest* (*fitness* global terbaik) maka hasil evaluasi fungsi $f(x_1, \dots, x_d)$ diperoleh nilai *fitness* sebanyak jumlah partikel atau p yaitu $f_1, \dots, f_p$. Selanjutnya *gBest* ditentukan dengan cara membandingkan nilai elvaluasi dari setiap partikel atau $f_1, \dots, f_p$. Nilai *fitness* yang terbaik akan menjadi *gBest* yang baru.

2.4.5 Memperbarui Kecepatan Partikel

Pada algoritma *Particle Swarm Optimization*, solusi diperbaiki dengan menggunakan kecepatan atau v_t yang merupakan hasil dari penjumlahan nilai momentum dan pengalaman yang didapat dari partikel terbaik yaitu *pBest* dan *gBest*. Momentum didapat degan mengkalikan bobot inersia dengan kecepatan pada iterasi sebelumnya atau $t-1$, di mana bobot inersia menentukan seberapa besar dampak dari kecepatan iterasi sebelumnya pada kecepatan iterasi sekarang. Komponen kognitif atau c_1 menentukan seberapa besar atau seberapa cepat

partikel tersebut atau p_t mendekati partikel lokal terbaik atau $pBest$. Sedangkan komponen sosial atau c_2 menentukan seberapa besar atau seberapa cepat partikel tersebut atau p_t mendekati partikel global terbaik atau $gBest$. Untuk nilai r_1 dan r_2 merupakan nilai acak dengan interval $[0, 1]$ yang berarti r_1 dan r_2 dapat bernilai 0 dan 1 itu sendiri. Berikut merupakan rumus yang digunakan dalam perhitungan kecepatan untuk masing-masing dimensi pada partikel.

$$v_t = wv_{t-1} + c_1r_1(pBest_t - p_t) + c_2r_2(gBest_t - p_t) \quad (2.2)$$

2.4.6 Memperbarui Posisi Partikel

Pada algoritma *Particle Swarm Optimization*, solusi yang merupakan posisi partikel atau p_t diperbaiki dengan berpindah mendekati partikel yang lebih optimal sebesar kecepatan atau v_t sehingga posisi menghasilkan partikel pada iterasi berikutnya atau p_{t+1} . Berikut merupakan rumus yang digunakan untuk menghitung posisi yang baru untuk masing-masing partikel.

$$p_{t+1} = p_t + v_t \quad (2.3)$$

2.4.7 Bobot Inersia *Particle Swarm Optimization*

Bobot inersia merupakan parameter penting dalam algoritma *Particle Swarm Optimization*. Bobot inersia mempengaruhi eksplorasi atau ruang pencarian di dalam proses optimasi *Particle Swarm Optimization* (Bansal, 2011). Pada awalnya, *Particle Swarm Optimization* yang diperkenalkan oleh Dr. Eberhart dan Dr. Kennedy pada tahun 1995 tidak menggunakan bobot inersia. Bobot inersia pertama kalinya diperkenalkan pada tahun 1998 oleh Shi dan Eberhart dengan menggunakan konsep *Constant Inertia Weight*. Shi dan Eberhart mengutarakan bahwa bobot inersia bernilai besar mempermudah pencarian global sedangkan bobot inersia bernilai kecil mempermudah pencarian lokal. Selanjutnya, *Dynamic Inertia Weight* ditemukan yang kemudian meningkatkan kemampuan dari algoritma *Particle Swarm Optimization*. Eberhart dan Shi membuktikan bahwa strategi *Random Inertia Weight* meningkatkan tingkat konvergensi algoritma *Particle Swarm Optimization* pada iterasi awal.

Pada penelitian ini untuk menentukan bobot inersia digunakan strategi *time varying inertia weight* (TVIW) dimana bobot inersia dihitung menggunakan persamaan (Su dkk, 2011):

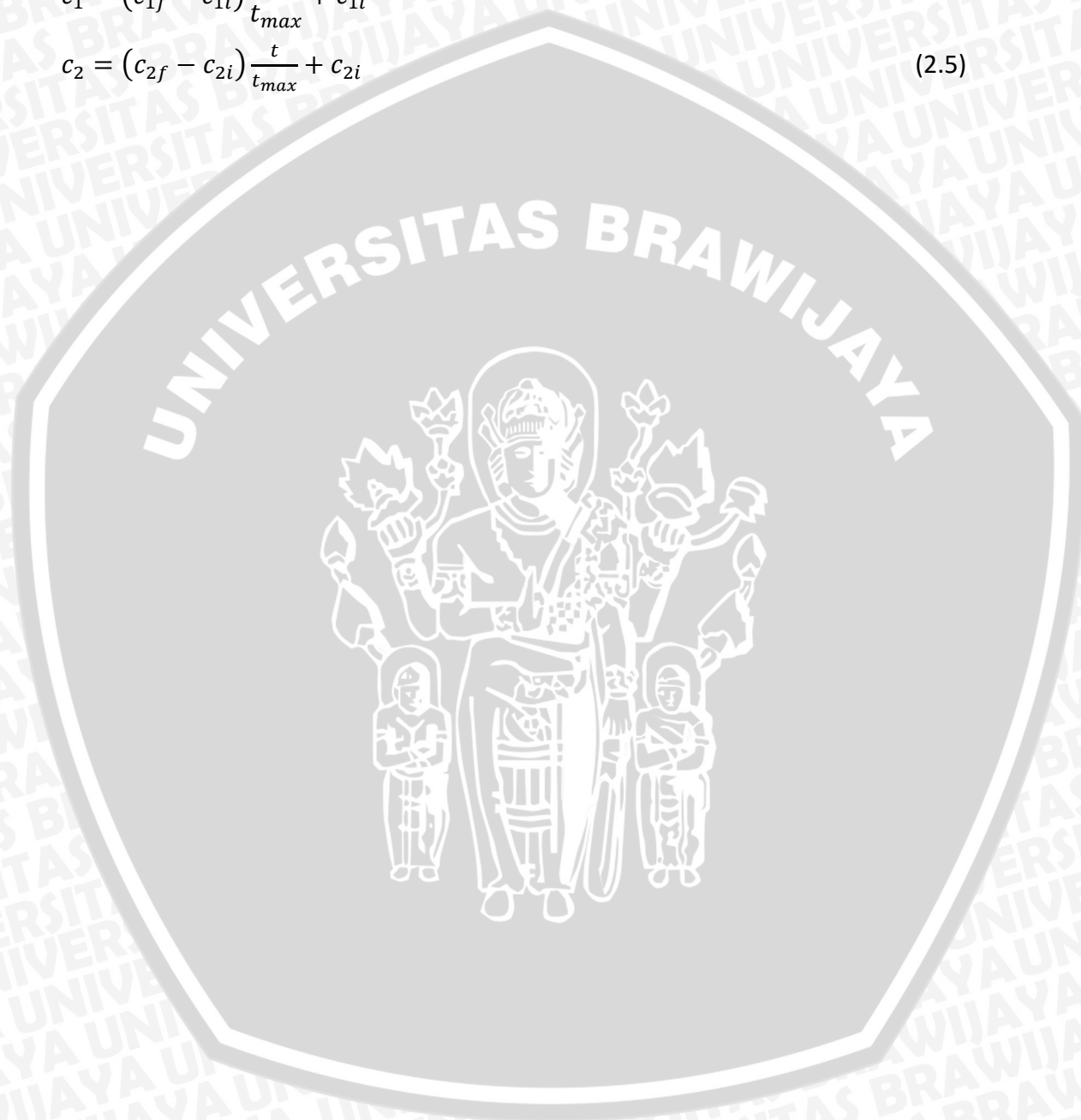
$$w_t = w_{min} + (w_{max} - w_{min}) \frac{(t_{max} - t)}{t_{max}} \quad (2.4)$$

Dengan menggunakan persamaan (2.5), nilai bobot inersia berubah pada setiap iterasi. Pada iterasi awal bobot inersia akan bernilai besar. Hal ini mempermudah eksplorasi pada iterasi di awal proses *Particle Swarm Optimization* sehingga sistem dapat menemukan daerah yang berpotensi untuk menghasilkan nilai partikel terbaik. Sedangkan semakin nilai iterasi mendekati batas maksimum iterasi, maka semakin kecil nilai bobot inersia. Hal ini mempermudah eksploitasi pada iterasi di akhir proses *Particle Swarm Optimization* sehingga sistem dapat menemukan partikel terbaik yang ada pada daerah yang didapat pada tahap eksplorasi.

2.4.8 Komponen Kognitif dan Sosial *Particle Swarm Optimization*

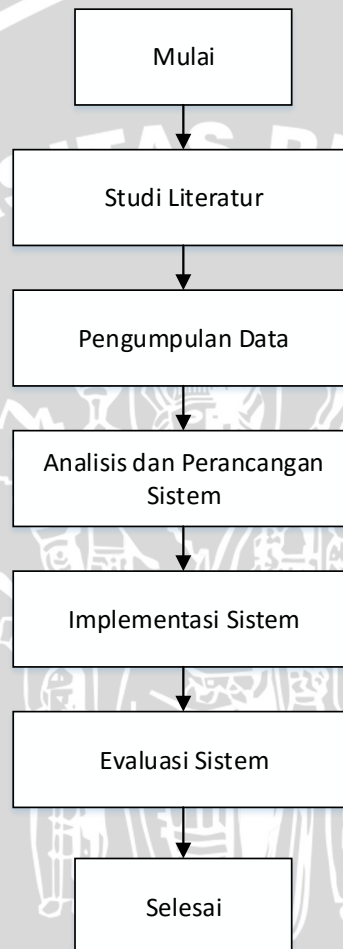
Pada penelitian ini, untuk menentukan komponen kognitif dan sosial digunakan strategi *time varying acceleration coefficient* (TVAC) dimana komponen sosial dihitung menggunakan persamaan:

$$\begin{aligned} c_1 &= (c_{1f} - c_{1i}) \frac{t}{t_{max}} + c_{1i} \\ c_2 &= (c_{2f} - c_{2i}) \frac{t}{t_{max}} + c_{2i} \end{aligned} \quad (2.5)$$



BAB 3 METODOLOGI

Pada bab ini akan dijelaskan metode yang digunakan dalam pengembangan sistem optimasi komposisi pupuk untuk sistem penanaman tumpangsari menggunakan algoritma *Particle Swarm Optimization*. Tahapan penelitian yang dilakukan meliputi studi literatur, pengumpulan data, analisis dan perancangan sistem, implementasi sistem, pengujian dan evaluasi hasil seperti pada Gambar 3.1 berikut.



Gambar 3.1 Tahapan Penelitian

3.1 Studi Literatur

Studi literatur dilakukan dengan mengumpulkan dan mempelajari literatur yang berkaitan dengan sistem optimasi, sistem penanaman tumpangsari, metode *Particle Swarm Optimization* (PSO), pemrograman dengan menggunakan bahasa pemrograman Java, kebutuhan nutrisi tanaman dan pengujian sistem. Sumber literatur berupa buku teks, *paper* dan jurnal.

3.2 Pengumpulan Data

Pengumpulan data dilakukan untuk menunjang keberhasilan dari penelitian. Pengumpulan data dari dataset yang dikumpulkan dalam melakukan pola tanam Tumpangsari yang didapat dari Balai Pengkajian Teknologi Pertanian untuk dataset kebutuhan komposisi pupuk jagung dan kacang. Penulis juga melakukan wawancara kepada pakar untuk mendapatkan informasi seperti penggunaan pupuk pertanian tumpangsari jagung dan kedelai. Data yang diperoleh berupa kebutuhan sistem yang akan dikembangkan seperti data interval pupuk tiap kabupaten dan rekomendasi pupuk kecamatan. Selain itu penulis juga melakukan wawancara kepada petani di daerah Tulungagung untuk mendapatkan informasi penggunaan pupuk pertanian yang digunakan oleh petani untuk penanaman tumpangsari jagung dan kedelai.

3.3 Analisis Kebutuhan

Analisa kebutuhan dilakukan dengan menentukan kebutuhan apa saja yang dibutuhkan untuk membangun sistem optimasi nutrisi pupuk tanaman tumpangsari menggunakan *Particle Swarm Optimization*. Berikut ini kebutuhan yang dibutuhkan dalam pengembangan sistem ini.

1. Kebutuhan *Hardware*, meliputi:
 - Laptop
2. Kebutuhan *Software*, meliputi:
 - Netbeans IDE 8.1
 - Ms. Word 2016
 - Ms. Visio 2016
3. Data yang dibutuhkan, meliputi:
 - Data kebutuhan pupuk jagung dan kacang

3.4 Perancangan Sistem

Perancangan sistem yang dibuat ini disusun berdasarkan atas langkah kerja dari sistem secara umum dengan model maupun desain arsitektur sistem yang mudah untuk diimplementasikan serta pengujiannya. Pertama pengguna akan memilih daerah tumpangsari, harga pupuk dan beberapa parameter algoritma *Particle Swarm Optimization*. Dari data tersebut maka akan dibangkitkan secara acak angka yang menyatakan besar dosis pupuk pada masing-masing jenis pupuk sesuai dengan aturan yang telah diterapkan oleh pakar sebelumnya. Kemudian dilakukan perhitungan optimasi nutrisi pupuk yang diinginkan beserta dengan harga yang optimal untuk pupuk tersebut. Hasil dari sistem ini merupakan rekomendasi komposisi nutrisi pupuk yang optimal.

3.5 Implementasi Sistem

Untuk mencapai hasil yang diharapkan, implementasi sistem dilakukan berdasarkan pada hasil analisis yang didapat pada perancangan sistem. Implementasi sistem dilakukan dengan menggunakan bahasa pemrograman Java dan *tools* pendukung lainnya. Perancangan yang dilakukan meliputi perancangan alur kerja sistem dari tahap penerimaan masukan dari user, proses optimasi pupuk

Particle Swarm Optimization serta perancangan antarmuka.

3.6 Pengujian dan Evaluasi Hasil

Pengujian dilakukan untuk mengetahui apakah sistem optimasi yang diimplementasikan sudah dapat berjalan sesuai spesifikasi kebutuhan yang telah ditentukan dan memenuhi tujuan yang diharapkan. Pengujian yang dilakukan mencakup pengujian parameter yang digunakan dalam *Particle Swarm Optimization* serta nilai hasil optimasi pupuk. Hasil pengujian kemudian dievaluasi untuk mempertimbangkan tingkat keberhasilan program.

3.7 Pengambilan Kesimpulan dan Saran

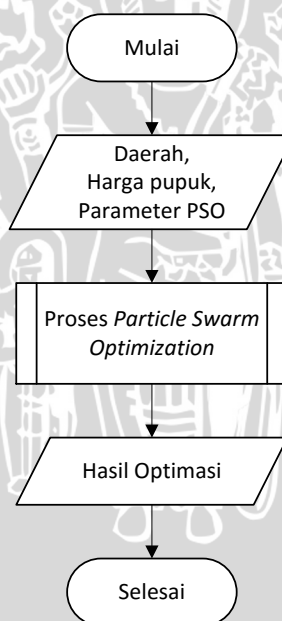
Tahap akhir meliputi penarikan kesimpulan terhadap hasil yang diperoleh dari penelitian yang dilakukan sehingga menjawab rumusan masalah yang telah ditentukan. Kemudian hasil pengujian digunakan sebagai bahan pertimbangan untuk dijadikan saran. Pemberian saran bertujuan sebagai acuan perbaikan sehingga penelitian ini dapat dilanjutkan untuk menjadi sistem yang lebih baik atau digunakan untuk mengembangkan aplikasi dengan objek yang sama dengan menggunakan metode optimasi lainnya.



BAB 4 PERANCANGAN

4.1 Deskripsi Umum Sistem

Secara umum sistem yang dikembangkan merupakan sebuah aplikasi perangkat lunak berbasis desktop yang berfungsi untuk melakukan perhitungan komposisi pupuk tanaman tumpangsari menggunakan algoritma *Particle Swarm Optimization*. Pertama pengguna memilih daerah dimana tanaman tumpangsari ditanam yang terdiri dari Kabupaten dan Kecamatan, kemudian pengguna menentukan harga masing-masing pupuk yaitu Urea, SP-36 dan KCL yang digunakan dan memasukkan beberapa parameter *Particle Swarm Optimization* yaitu jumlah partikel, banyak iterasi, inersia maksimum, inersia minimum, $c1i$, $c1f$, $c2i$, $c2f$. Setelah parameter dimasukan, program akan melakukan proses optimasi menggunakan algoritma *Particle Swarm Optimization* sampai kondisi akhir terpenuhi. Dari proses optimasi *Particle Swarm Optimization* ini menghasilkan hasil optimasi berupa komposisi pupuk jagung dan kacang kedelai untuk satu lahan dengan luas . Gambar 4.1 berikut merupakan diagram alir dari alur kerja sistem secara umum.

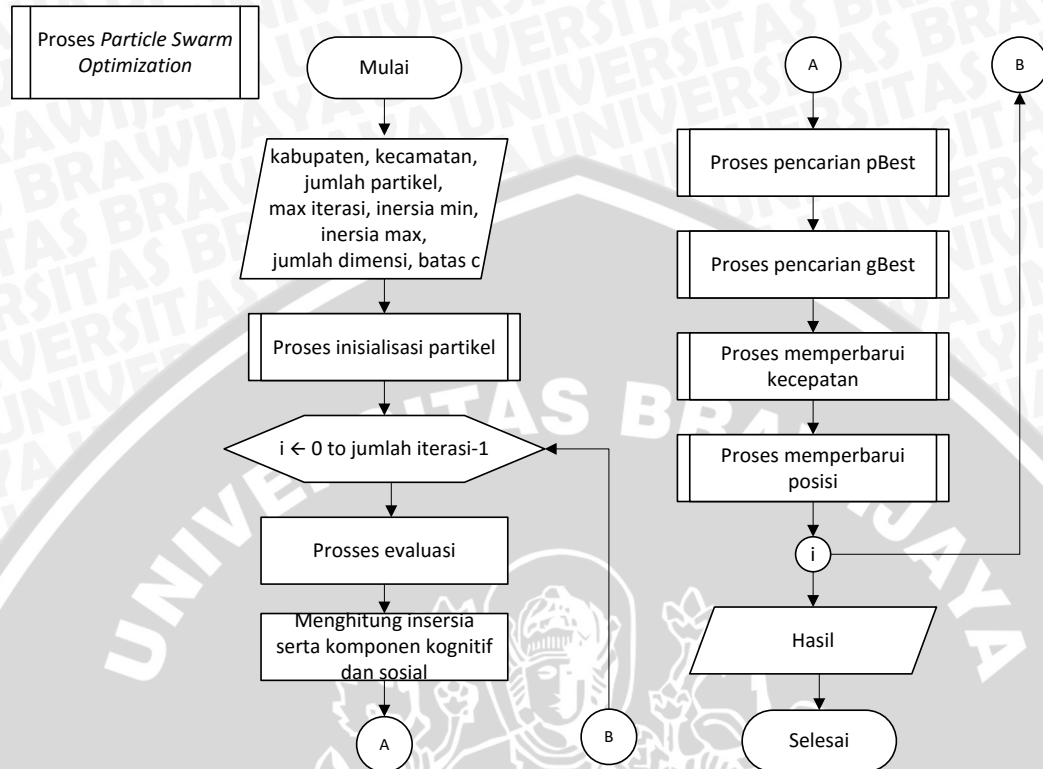


Gambar 4.1 Alur Kerja Sistem

4.2 Perancangan Implementasi *Particle Swarm Optimization*

Dalam proses optimasi *Particle Swarm Optimization* pada sistem dibagi kedalam beberapa subproses yang mempresentasikan langkah-langkah dalam algoritma *Particle Swarm Optimization*. Berikut merupakan diagram alir proses

perhitungan *Particle Swarm Optimization* yang digunakan untuk implementasi sistem optimasi pupuk.



Gambar 4.2 Proses Perhitungan Optimasi Komposisi Pupuk Menggunakan *Particle Swarm Optimization*

Pada Gambar 4.2 di atas merupakan diagram alir dari proses perhitungan optimasi komposisi pupuk menggunakan algoritma *Particle Swarm Optimization* dengan proses sebagai berikut:

1. Inisialisasi kabupaten, kecamatan, jumlah partikel, jumlah iterasi, inersia min, inersia max, jumlah dimensi dan batas c yaitu $c1f$, $c1i$, $c2f$ serta $c2i$.
2. Menginisialisasi partikel sebanyak ukuran jumlah dimensi yang ditentukan.
3. Menghitung nilai fitness setiap partikel dengan menggunakan rumus yang telah ditentukan.
4. Menghitung nilai inersia yang akan digunakan pada satu iterasi berdasarkan $wmin$ dan $wmax$ yang telah ditentukan serta menghitung komponen kognitif dan sosial berdasarkan batas c yaitu nilai $c1f$, $c1i$, $c2f$ serta $c2i$.
5. Mencari partikel $pBest$ dengan cara membandingkan nilai evaluasi setiap partikel dengan nilai evaluasi partikel tersebut pada iterasi sebelumnya.
6. Mencari partikel $gBest$ dengan cara membandingkan nilai evaluasi setiap partikel dalam satu iterasi.
7. Menghitung kecepatan setiap dimensi pada partikel.

8. Memperbarui posisi partikel sesuai dengan kecepatan yang diperoleh.
9. Megulangi tahap 3 sampai tahap 8 sampai jumlah iterasi terpenuhi.
10. Setelah jumlah iterasi tercapai, dikeluarkan hasil akhir yang berupa komposisi pupuk.

4.2.1 Inisialisasi Partikel

Inisialisasi partikel dilakukan dengan cara membangkitkan nilai secara acak komposisi pupuk yang akan digunakan pada tanaman tumpangsari berdasarkan interval daerah tumpangsari berdasarkan Tabel 4.1 berikut.

Tabel 4.1 Tabel Interval Pupuk Tanaman Tumpangsari

Kabupaten	Rekomendasi Pupuk (Kg/Ha)					
	Jagung			Kedelai		
	Urea	SP-36	KCL	Urea	SP-36	KCL
Bojonegoro	350	100-150	50-100	50	50-100	50-100
Gresik	350	100-125	50-75	50	50-100	50-75
Lamongan	350	100-150	50-75	50	50-100	50-75
Madiun	350	100-150	50-100	50	50-100	50-100
Ngawi	350	100-125	50-100	50	50-75	50-100
Tulungagung	350	100-125	75-100	50	50-75	75-100

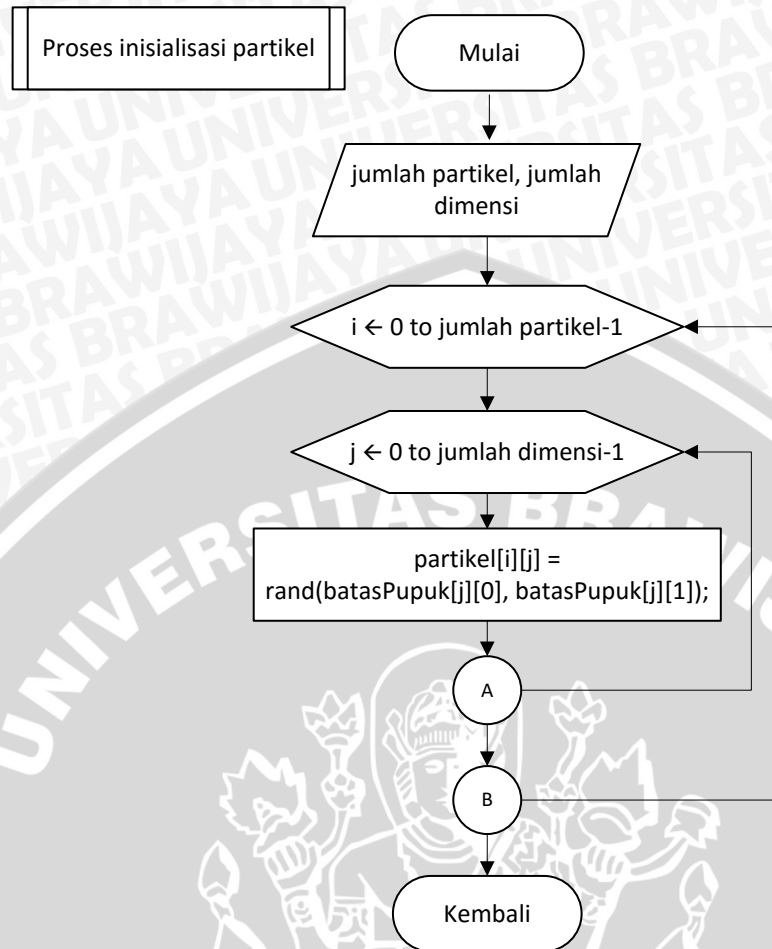
Sumber: Badan Pengkajian Teknologi Pertanian, 2014.

Nilai tersebut lalu dimasukan kedalam suatu format susunan kombinasi pupuk. Panjang dari kromosom adalah sebanyak jumlah gen yang ada. Masing-masing gen mewakili masing-masing kebutuhan hara pada tanaman yang dibangkitkan secara acak sesuai dengan interval data yang ada pada sistem.

Urea Tanaman 1	SP-36 Tanaman 1	KCL Tanaman 1	Urea Tanaman 2	SP-36 Tanaman 2	KCL Tanaman 2
-------------------	--------------------	------------------	-------------------	--------------------	------------------

Gambar 4.3 Kombinasi Pembentukan Partikel

Terdapat enam gen yang menjadi komposisi pada tiap individu dimana dibagi kedalam dua bagian karena individu dibentuk berdasarkan nutrisi pupuk tiap tanaman. Gen pertama hingga gen ketiga merupakan rekomendasi pupuk yang akan diberikan pada jenis tanaman pertama, sedangkan gen keempat hingga gen keenam merupakan kelompok rekomendasi pupuk yang menjadi calon solusi nutris pupuk optimal pada jenis tanaman kedua.



Gambar 4.4 Inisialisasi Partikel

Pada Gambar 4.4 di atas merupakan diagram alir dari proses inisialisasi partikel pada algoritma *Particle Swarm Optimization* dengan proses sebagai berikut:

1. Mengambil variabel jumlah partikel dan jumlah dimensi sebagai parameter proses inisialisasi partikel.
2. Melakukan perulangan sebanyak jumlah partikel.
3. Di dalam perulangan jumlah partikel, melakukan perulangan sebanyak jumlah dimensi.
4. Menggacak nilai dimensi suatu partikel berdasarkan batasan rekomendasi pupuk yang telah ditentukan.
5. Setelah jumlah partikel terpenuhi kembali ke proses *Partikel Swarm Optimization* untuk melanjutkan ke proses evaluasi partikel.

4.2.2 Proses Evaluasi / Perhitungan *Fitness*

Proses evaluasi dilakukan untuk menilai partikel mana yang paling optimal. Proses evaluasi dilakukan dengan cara mencari nilai *fitness* dari setiap partikel yang ada menggunakan modifikasi persamaan (2.1). Dengan menggunakan persamaan (2.1) didapat hasil dari evaluasi atau nilai *fitness* dengan nilai maksimal 7 angka dibelakan koma. Oleh karena hal tersebut untuk mempermudah perbandingan nilai *fitness*, pada penelitian ini digunakan nilai variabel pembilang dengan nilai 10^7 .

$$fitness = \frac{10^7}{(total\ harga + penalti)} \quad (4.1)$$

Dalam persamaan diatas terdapat total harga dan nilai penalti. Total harga merupakan hasil dari total jumlah perkalian komposisi pupuk dengan harga masing-masing pupuk, sedangkan penalti merupakan selisih atau hasil pengurangan komposisi pupuk rekomendasi pakar pada daerah yang dipilih dengan komposisi pupuk yang dihasilkan oleh sistem. Sehingga semakin kecil total biaya yang digunakan maka semakin besar nilai *fitness* yang menandakan semakin optimal partikel tersebut. Hal tersebut juga berlaku pada nilai penalti, di mana semakin kecil penalti yang digunakan maka semakin besar nilai *fitness* yang menandakan semakin optimal partikel tersebut. Pada Tabel 4.2 berikut merupakan ketentuan harga pupuk.

Tabel 4.2 Tabel Harga Pupuk

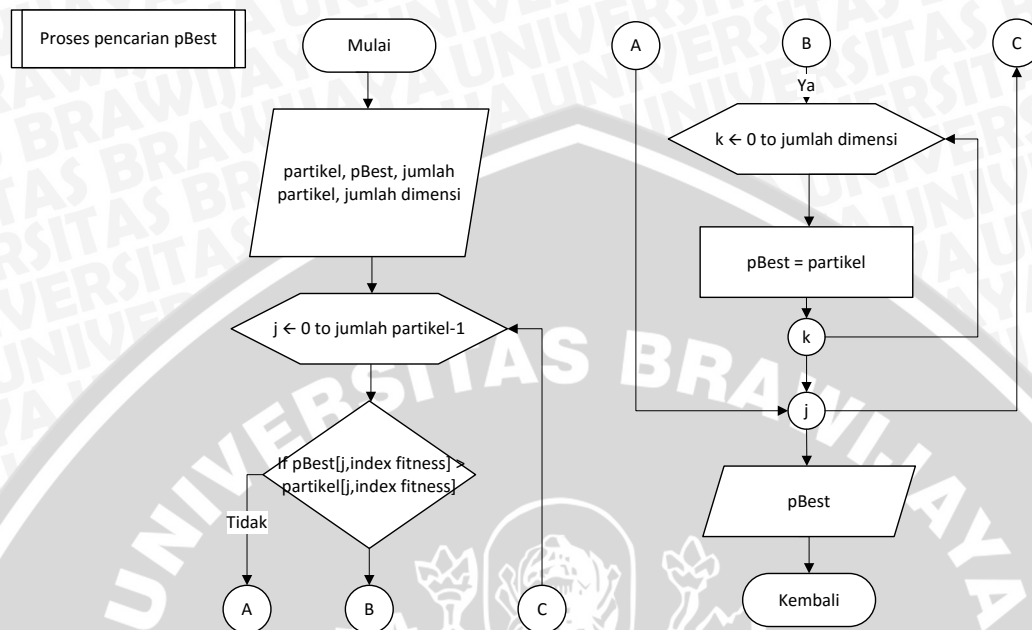
Nama Pupuk	Harga (Rp/Kg)
Urea	1.800
NPK	2.300
ZA	1.400
SP-36	2.000
KCL	5.600
Phonska	2.300
Dolomit	130
Kandang	500
Kompos	500
Hijau	500

Sumber: Badan Pengkajian Teknologi Pertanian, 2014.

4.2.3 Proses Pencarian *pBest*

Proses pencarian *pBest* dimulai dengan melakukan perulangan sebanyak jumlah partikel. Perulangan ini dilakukan untuk mencari *fitness* yang optimal dari setiap partikel. *Fitness* partikel pada iterasi tersebut dibandingkan dengan *fitness* partikel tersebut pada iterasi sebelumnya. Partikel dengan nilai *fitness* paling

optimal akan disimpan sebagai $pBest$ yang baru. Alur kerja pencarian $pBest$ dapat dilihat pada Gambar 4.5 di bawah ini.



Gambar 4.5 Proses Pencarian $pBest$

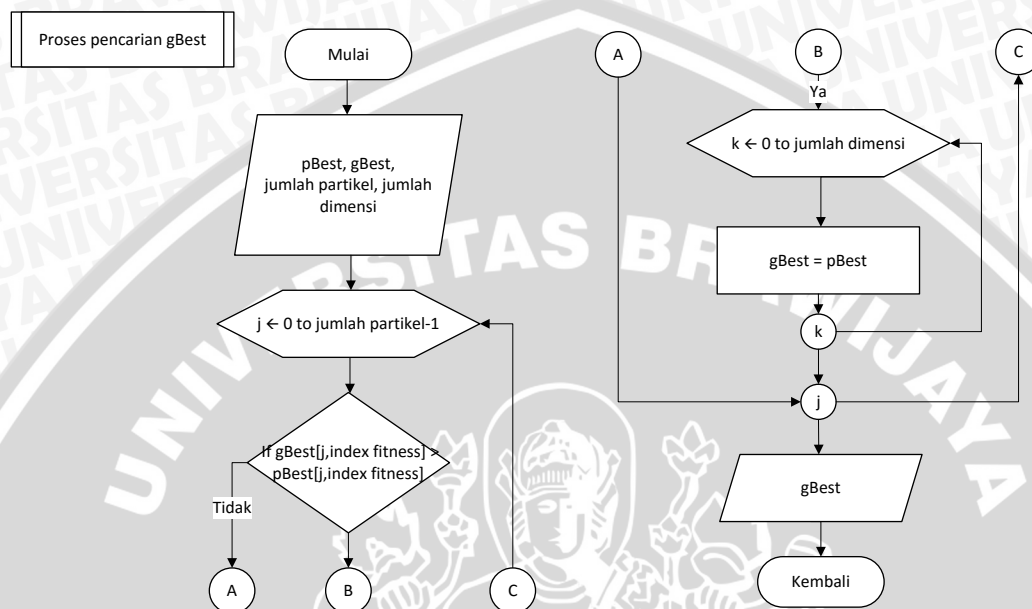
Pada Gambar 4.5 di atas merupakan diagram alir dari proses pencarian $pBest$ pada algoritma *Particle Swarm Optimization* dengan proses sebagai berikut:

1. Mengambil array partikel dan $pBest$ serta variabel jumlah partikel dan jumlah dimensi sebagai parameter proses pencarian $pBest$.
2. Melakukan perulangan sebanyak jumlah partikel.
3. Membandingkan nilai *fitness* partikel dengan nilai *fitness* partikel yang telah tersimpan pada $pBest$.
4. Bila nilai *fitness* partikel lebih besar dari nilai *fitness* $pBest$, maka partikel tersebut merupakan partikel lokal terbaik yang baru.
5. Partikel lokal terbaik yang baru dimasukkan ke *array* $pBest$ dengan melakukan perulangan sebanyak dimensi kemudian mengganti nilai tiap dimensi $pBest$ yang lama dengan yang baru.
6. Setelah didapat $pBest$ pada tiap partikel, sistem kembali ke proses *Partikel Swarm Optimization* untuk melanjutkan ke proses pencarian $gBest$.

4.2.4 Proses Pencarian $gBest$

Proses pencarian $gBest$ dimulai dengan melakukan perulangan sebanyak jumlah partikel. Perulangan ini dilakukan untuk mencari partikel global terbaik.

Fitness partikel pada iterasi tersebut dibandingkan dengan *fitness* pada *gBest* sebelumnya. Hal ini memungkinkan sistem mendapatkan partikel dengan *fitness* paling optimal dari seluruh partikel dari seluruh iterasi yang akan disimpan sebagai *gBest* yang baru. Alur kerja pencarian *gBest* dapat dilihat pada Gambar 4.6 di bawah ini.

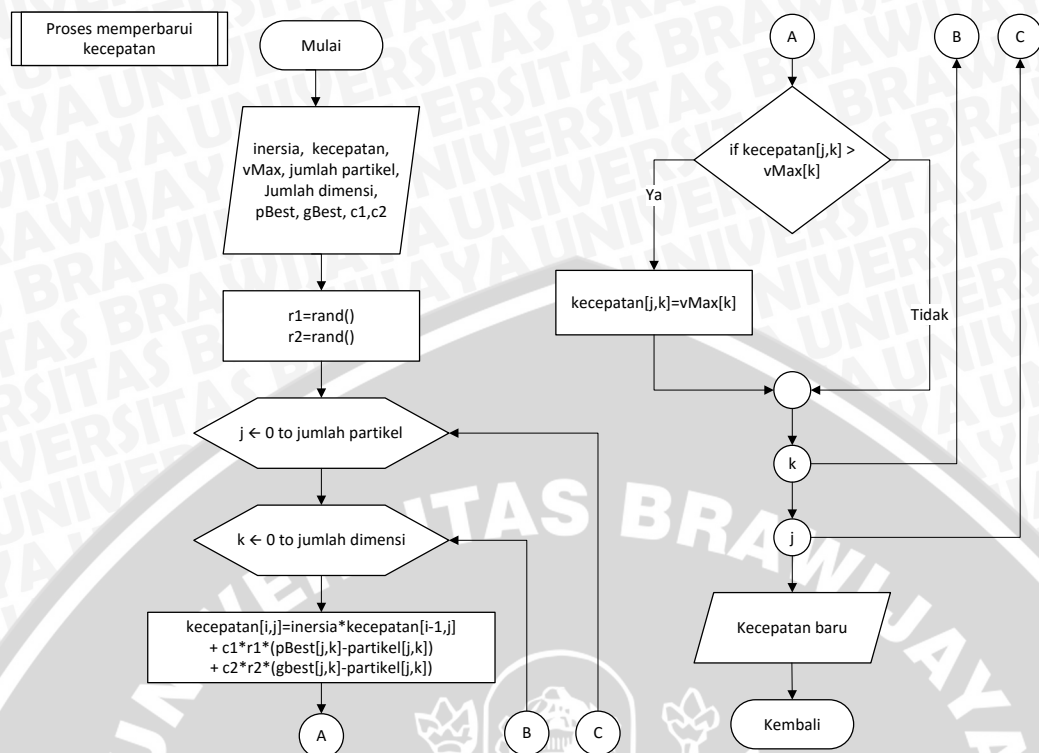


Gambar 4.6 Proses Pencarian *gBest*

Pada Gambar 4.6 di atas merupakan diagram alir dari proses pencarian *gBest* pada algoritma *Particle Swarm Optimization* dengan proses sebagai berikut:

1. Mengambil array *pBest* dan *gBest* serta variabel jumlah partikel dan jumlah dimensi sebagai parameter proses pencarian *gBest*.
2. Melakukan perulangan sebanyak jumlah partikel.
3. Membandingkan nilai *fitness* partikel pada array *pBest* dengan nilai *fitness* partikel pada array *gBest* sehingga didapat partikel global terbaik yang baru.
4. Partikel global terbaik yang baru dimasukkan ke variabel *pBest* dengan melakukan perulangan sebanyak dimensi kemudian mengganti nilai tiap dimensi *gBest* yang lama dengan yang baru.
5. Setelah didapat *gBest* iterasi tersebut, sistem kembali ke proses *Partikel Swarm Optimization* untuk melanjutkan ke proses memperbaiki kecepatan.

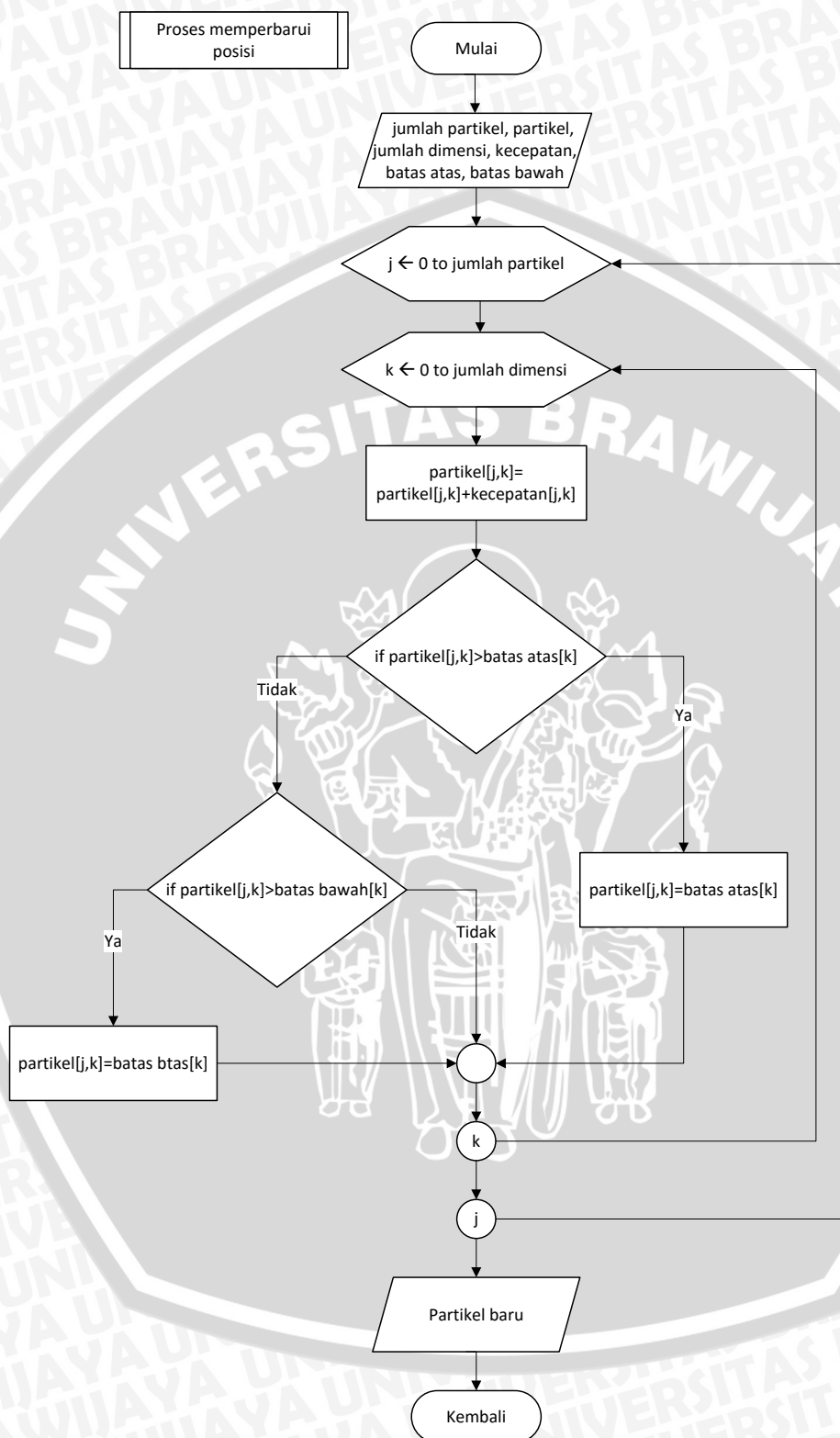
4.2.5 Proses Memperbarui Kecepatan



Gambar 4.7 Proses Memperbarui Kecepatan

Seperti pada Gambar 4.7 proses memperbarui kecepatan dimulai dengan menentukan secara acak nilai r_1 dan r_2 . Setelah itu dilakukan perulangan sebanyak jumlah partikel dan jumlah dimensi untuk menghitung kecepatan setiap dimensi pada setiap partikel. Jika kecepatan yang didapat melebihi batas kecepatan maksimal, maka besar kecepatan tersebut dianggap sama dengan batas kecepatan.

4.2.6 Proses Memperbarui Posisi



Gambar 4.8 Proses Memperbarui Posisi

Pada Gambar 4.8 di atas merupakan diagram alir dari proses memperbaiki posisi pada algoritma *Particle Swarm Optimization* dengan proses sebagai berikut:

1. Mengambil array partikel dan kecepatan serta variabel jumlah partikel, jumlah dimensi, batas atas dan batas bawah sebagai parameter proses memperbaiki posisi.
2. Melakukan perulangan sebanyak jumlah partikel.
3. Di dalam perulangan sebanyak jumlah partikel, melakukan perulangan sebanyak jumlah dimensi.
4. Menghitung posisi baru dari setiap dimensi partikel.
5. Bila nilai dari dimensi partikel kurang dari batas bawah maka nilai partikel dianggap bernilai sama dengan batas bawah.
6. Bila nilai dari dimensi partikel lebih dari batas atas maka nilai partikel dianggap bernilai sama dengan batas atas.
7. Setelah didapat posisi partikel terbaru, sistem kembali ke proses *Partikel Swarm Optimization* untuk melanjutkan iterasi selanjutnya.

4.3 Contoh Perhitungan Manual *Particle Swarm Optimization*

Dalam proses optimasi komposisi pupuk pada pola tanam tumpangsari menggunakan *Particle Swarm Optimization* berdasarkan gambar 4.1 terdiri dari dua tahapan utama. Tahap pertama adalah proses memasukan parameter daerah, harga dan parameter *Particle Swarm Optimization* oleh pengguna. Kemudian tahap kedua adalah proses pencarian rekomendasi komposisi pupuk dengan algoritma *Particle Swarm Optimization* sesuai dengan masukan yang ditentukan.

Pada contoh perhitungan manual *Particle Swarm Optimization* ini menggunakan parameter sebagai berikut:

- | | |
|-----------------------|----------------|
| a. Kabupaten Lamongan | : Lamongan |
| b. Kecamatan Babat | : Babat |
| c. Harga Urea | : Rp. 1.800,00 |
| d. Harga SP-36 | : Rp. 2.000,00 |
| e. Harga KCL | : Rp. 5.600,00 |
| f. Jumlah Iterasi | : 2 |
| g. Jumlah Partikel | : 5 |
| h. w_{min} | : 0,4 |
| i. w_{max} | : 0,9 |
| j. $c1i$ | : 2,5 |
| k. $c1f$ | : 0,5 |
| l. $c2i$ | : 0,5 |
| m. $c2f$ | : 2,5 |

Penjelasan mengenai detail tahapan perhitungan rekomendasi komposisi pupuk dengan algoritma *Particle Swarm Optimization* akan dibahas pada subbab 4.3.1 sampai 4.3.

4.3.1 Inisialisasi Partikel

Populasi awal dibangkitkan sebanyak jumlah partikel yang telah ditentukan sebelumnya. Nilai pada setiap dimensi partikel didapatkan dari mengacak nilai sesuai dengan interval pada data interval rekomendasi pupuk oleh pakar pada Tabel 4.1 seperti berikut.

- Urea jagung dengan interval 350 kg/ha, sehingga didapat nilai acak sebesar 350 kg/ha.
- SP-36 jagung dengan interval 100 sampai dengan 150 kg/ha, sehingga didapat nilai acak sebesar 149 kg/ha.
- KCL jagung dengan interval 50 sampai dengan 75 kg/ha, sehingga didapat nilai acak sebesar 65 kg/ha.
- Urea kedelai dengan interval 50 kg/ha, sehingga didapat nilai acak sebesar 50 kg/ha.
- SP-36 kedelai dengan interval 50 sampai dengan 100 kg/ha, sehingga didapat nilai acak sebesar 64 kg/ha.
- KCL kedelai dengan interval 50 sampai dengan 75 kg/ha, sehingga didapat nilai acak sebesar 72 kg/ha.

Dari nilai yang telah didapat di atas dapat dibentuk menjadi satu susunan partikel dengan susunan seperti pada Tabel 4.3 berikut.

Tabel 4.3 Representasi Partikel Ke-1 Pada Populasi Awal

Partikel ke-	Jagung			Kedelai		
	Urea	SP-36	KCL	Urea	SP-36	KCL
1	350	149	65	50	64	72

Kemudian tahapan di atas akan diulangi sebanyak jumlah partikel yang telah ditentukan. Proses tersebut akan menghasilkan populasi partikel awal seperti pada Tabel 4.4 berikut.

Tabel 4.4 Representasi Seluruh Partikel Populasi Awal

Partikel ke-	Jagung			Kedelai		
	Urea	SP-36	KCL	Urea	SP-36	KCL
1	350	149	65	50	64	72
2	350	141	54	50	68	74
3	350	124	69	50	66	73
4	350	106	56	50	56	66
5	350	139	68	50	55	61

4.3.2 Inisialisasi Kecepatan

Setelah populasi awal dibangkitkan, tahap selanjutnya adalah proses inisialisasi kecepatan. Pada algoritma *Particle Swarm Optimization* kecepatan mempengaruhi posisi suatu partikel ke arah tertentu, sehingga besar kecepatan yang mempengaruhi setiap dimensi pada suatu partikel berbeda. Oleh karena itu kecepatan dapat direpresentasikan ke dalam tabel dengan ukuran jumlah partikel dikali ukuran dimensi partikel tersebut. Untuk inisialisasi dianggap suatu partikel masih dalam keadaan diam, sehingga kecepatan awal dianggap bernilai 0 untuk setiap dimensi pada setiap partikel populasi awal.

4.3.3 Proses Evaluasi / Perhitungan *Fitness*

Tahap awal untuk melakukan perhitungan nilai *fitness* pada proses optimasi ini adalah dengan menghitung nilai penalti untuk masing-masing dimensi pada partikel. Proses perhitungan nilai penalti pada masing-masing dimensi dilakukan dengan cara menghitung selisih antara komposisi pupuk tiap dimensi yang telah dibangkitkan dengan komposisi pupuk kecamatan yang direkomendasikan pakar pada Tabel 2.2 sampai Tabel 2.7. Pada Tabel 2.3 didapat rekomendasi pemupukan pada kecamatan Babat seperti berikut.

Tabel 4.5 Rekomendasi Pemupukan Kecamatan Babat

Kecamatan	Jagung			Kedelai		
	Urea	SP-36	KCL	Urea	SP-36	KCL
Babat	350	125	75	50	75	75

Berikut merupakan perhitunagn nilai penalti pada parikel ke-1 pada populasi awal yang telah diinisialisasi pada Tabel 4.4.

- Selisih rekomendasi Urea Jagung = $|350 - 350| = 0$
- Selisih rekomendasi SP-36 Jagung = $|149 - 125| = 24$
- Selisih rekomendasi KCL Jagung = $|65 - 75| = 10$
- Selisih rekomendasi Urea Kedelai = $|50 - 50| = 0$
- Selisih rekomendasi SP-36 Kedelai = $|64 - 75| = 11$
- Selisih rekomendasi KCL Kedelai = $|72 - 75| = 3$
- Penalti partikel ke-1 = $(0 + 24 + 10 + 0 + 11 + 3) \times 10000 = 480000$

Setelah didapat nilai penalti, dilakukan perhitungan untuk mencari total harga yang dikeluarkan. Total harga merupakan komposisi dikali harga untuk setiap jenis pupuknya. Berikut merupakan perhitunagn total harga pada parikel ke-1 pada populasi awal yang telah diinisialisasi pada Tabel 4.4 dengan menggunakan harga yang telah ditentukan pada Tabel 4.2.

- Harga Urea Jagung = $350 \times 1800 = 630000$
- Harga SP-36 Jagung = $149 \times 2000 = 298000$
- Harga KCL Jagung = $65 \times 5600 = 364000$

- Harga Urea Kedelai = $50 \times 1800 = 90000$
- Harga SP-36 Kedelai = $64 \times 2000 = 128000$
- Harga KCL Kedelai = $72 \times 5600 = 403200$
- Total harga = Harga Urea Jagung + Harga SP – 36 Jagung +
Harga KCL Jagung + Harga Urea Kedelai + Harga SP – 36 Kedelai +
Harga KCL Kedelai = $630000 + 298000 + 364000 + 90000 + 128000 +$
 $403200 = 1913200$

Setelah didapat nilai *penalty* dan total harga, dilakukan perhitungan untuk mencari nilai *fitness* dengan menggunakan rumus yang telah ditentukan.

$$\text{Fitness partikel ke-1} = \frac{10^7}{(\text{total harga} + \text{penalti})} = \frac{10^7}{(1913200 + 480000)} = 4,179$$

Perhitungan di atas akan diulang sampai didapat nilai *fitness* untuk semua partikel pada populasi seperti pada Tabel 4.6 berikut.

Tabel 4.6 Hasil Evaluasi Partikel Pada Populasi Awal

Partikel ke-	Jagung			Kedelai			Fitness
	Urea	SP-36	KCL	Urea	SP-36	KCL	
1	350	149	65	50	64	72	4,179
2	350	141	54	50	68	74	4,339
3	350	124	69	50	66	73	4,819
4	350	106	56	50	56	66	4,189
5	350	139	68	50	55	61	4,201

4.3.4 Proses Pencarian *pBest*

Setelah didapat nilai *fitness* seluruh partikel dilakukan pencarian *pBest*. Untuk mencari partikel *pBest*, partikel pada iterasi tersebut dibandingkan nilai *fitness*-nya dengan nilai *fitness pBest*-nya. Pada contoh ini digunakan partikel ke-1 pada Tabel 4.6.

Tabel 4.7 Partikel Ke-1 Pada Populasi Awal Dengan *Fitness*-nya

Partikel ke-	Jagung			Kedelai			Fitness
	Urea	SP-36	Fitness	Urea	SP-36	KCL	
1	350	149	65	50	64	72	4,179

Karena pada contoh yang digunakan masih pada tahap awal dan nilai *pBest* belum ada, partikel langsung dianggap menjadi partikel *pBest*. Sehingga populasi partikel *pBest* menjadi seperti pada Tabel 4.8 berikut.

Tabel 4.8 Partikel $pBest$

Partikel ke-	Jagung			Kedelai			<i>Fitness</i>
	Urea	SP-36	KCL	Urea	SP-36	KCL	
1	350	149	65	50	64	72	4,179
2	350	141	54	50	68	74	4,339
3	350	124	69	50	66	73	4,819
4	350	106	56	50	56	66	4,189
5	350	139	68	50	55	61	4,201

Misalkan pada iterasi selanjutnya pada partikel ke-1 didapat nilai *fitness* yang lebih baik daripada nilai *fitness* pada partikel $pBest$ maka partikel tersebut akan menggantikan partikel ke-1 pada populasi $pBest$.

4.3.5 Proses Pencarian $gBest$

Setelah didapat populasi $pBest$, dilakukan pencarian $pBest$. Pada contoh ini digunakan populasi $pBest$ yang didapat pada Tabel 4.8. Untuk mencari partikel $gBest$, antar partikel $pBest$ pada iterasi tersebut dibandingkan sehingga didapat partikel dengan nilai *fitness* terbaik. Pada Tabel 4.8 nilai *fitness* terbaik ada pada partikel ke-3 dengan *fitness* 4,819.

Tabel 4.9 Partikel $gBest$

Partikel ke-	Jagung			Kedelai			<i>Fitness</i>
	Urea	SP-36	<i>Fitness</i>	Urea	SP-36	KCL	
3	350	124	69	50	66	73	4,819

4.3.6 Menghitung Bobot Inersia

Sebelum melakukan proses memperbarui kecepatan partikel, dilakukan perhitungan untuk menentukan nilai bobot inersia yang digunakan dengan menggunakan persamaan (2.5). Melanjutkan contoh sebelumnya, karena masih pada tahap inisialisasi didapat nilai $t = 0$ sehingga bobot inersia dihitung seperti berikut.

$$w_t = w_{min} + (w_{max} - w_{min}) \frac{(t_{max} - t)}{t_{max}} = 0,4 + (0,9 - 0,4) \frac{(2 - 0)}{2} = 0,9$$

4.3.7 Menghitung Komponen Kognitif dan Sosial

Sebelum melakukan proses memperbarui kecepatan partikel, dilakukan perhitungan untuk menentukan nilai komponen kognitif dan sosial yang digunakan dengan menggunakan persamaan (2.6). Melanjutkan contoh sebelumnya, karena masih pada tahap inisialisasi didapat nilai $t = 0$ sehingga komponen kognitif dan sosial dihitung seperti berikut.

$$c_1 = (c_{1f} - c_{1i}) \frac{t}{t_{max}} + c_{1i} = (0,5 - 2,5) \frac{0}{2} + 2,5 = 2,5$$

$$c_2 = (c_{2f} - c_{2i}) \frac{t}{t_{max}} + c_{2i} = (0,5 - 2,5) \frac{0}{2} + 0,5 = 0,5$$

4.3.8 Proses Memperbarui Kecepatan

Pada proses memperbarui kecepatan partikel ini digunakan dimensi ke-2 pada partikel ke-1 populasi awal di Tabel 4.4 dengan menggunakan nilai r_1 sebesar 0,5 dan r_2 sebesar 0,6.

$$\begin{aligned} v_t &= wv_{t-1} + c_1r_1(pBest_t - p_t) + c_2r_2(gBest_t - p_t) \\ &= 0,9 \times 0 + 2,5 \times 0,5 \times (149 - 149) + 0,5 \times 0,6 \times (124 - 149) = -7,5 \end{aligned}$$

Perhitungan di atas akan diulang sampai didapat kecepatan untuk semua dimensi pada setiap partikel pada populasi seperti pada Tabel 4.10 berikut.

Tabel 4.10 Kecepatan Partikel

Partikel ke-	Jagung			Kedelai		
	Urea	SP-36	KCL	Urea	SP-36	KCL
1	0,00	-7,50	1,20	0,00	0,60	0,30
2	0,00	-5,10	4,50	0,00	-0,60	-0,30
3	0,00	0,00	0,00	0,00	0,00	0,00
4	0,00	5,40	3,90	0,00	3,00	2,10
5	0,00	-4,50	0,30	0,00	3,30	3,60

4.3.9 Proses Memperbarui Posisi

Setelah didapat kecepatan dilakukan proses memperbarui posisi partikel. Proses memperbarui posisi partikel dilakukan dengan menjumlahkan kecepatan yang didapat dengan nilai setiap dimensi pada posisi partikel. Berikut merupakan contoh proses memperbarui partikel pada partikel ke-1 dengan kecepatan partikel pada Tabel 4.10.

- Urea Jagung = $350 + 0,00 = 350,00$
- SP-36 Jagung = $149 + (-7,50) = 141,50$
- KCL Jagung = $65 + 1,20 = 66,20$
- Urea Kedelai = $50 + 0,00 = 50,00$
- SP-36 Kedelai = $64 + 0,60 = 64,60$
- KCL Kedelai = $72 + 0,30 = 72,30$

Berdasarkan perhitungan diatas didapat posisi terbaru partikel ke-1 yang akan digunakan pada iterasi selanjutnya seperti pada Tabel 4.11 berikut.

Tabel 4.11 Posisi Terbaru Partikel ke-1

Partikel ke-	Jagung			Kedelai		
	Urea	SP-36	KCL	Urea	SP-36	KCL
1	350	149	65	50	64	72

Proses di atas akan diulang sampai seluruh posisi partikel pada populasi awal telah diperbarui sehingga menghasilkan populasi seperti pada Tabel 4.12 berikut.

Tabel 4.12 Representasi Seluruh Partikel Populasi Awal

Partikel ke-	Jagung			Kedelai		
	Urea	SP-36	KCL	Urea	SP-36	KCL
1	350,00	141,50	66,20	50,00	64,60	72,30
2	350,00	135,90	58,50	50,00	67,40	73,70
3	350,00	124,00	69,00	50,00	66,00	73,00
4	350,00	111,40	59,90	50,00	59,00	68,10
5	350,00	134,50	68,30	50,00	58,30	64,60

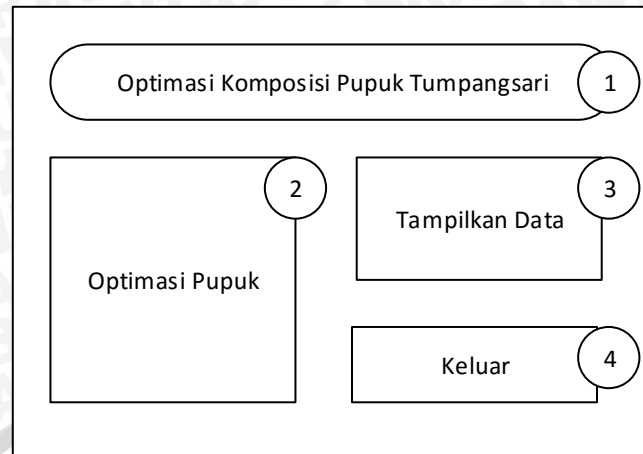
Proses ini akan menjadi akhir dari iterasi dan proses Particle Swarm Optimization akan dilanjutkan pada iterasi selanjutnya.

4.4 Perancangan Antarmuka

Aplikasi Optimasi Komposisi Pupuk Sistem Penanaman Tumpangsari Menggunakan Algoritma *Particle Swarm Optimization* ini terbagi menjadi tiga halaman. Halaman pertama merupakan halaman Beranda sebagai tampilan utama yang akan ditampilkan pertama kali ketika perangkat lunak dijalankan. Pada halaman ini terdapat beberapa menu utama untuk mengakses halaman lainnya. Halaman kedua merupakan halaman Optimasi Pupuk dimana pada halaman ini fungsi utama dari aplikasi yaitu proses optimasi komposisi pupuk dijalankan. Pada halaman Optimasi Pupuk terdapat semua kolom masukan yang akan digunakan sebagai parameter optimasi serta tampilan perhitungan dan hasil optimasi komposisi pupuk. Halaman terakhir adalah halaman data. Pada halaman Data terdapat beberapa tabel untuk menampilkan seluruh data daerah dan rekomendasi pupuk. Rancangan antarmuka akan dijelaskan pada sub-bab 4.3.1 sampai sub-bab 4.3.3.

4.4.1 Halaman Beranda

Halaman Beranda merupakan tampilan utama sistem yang akan ditampilkan pertama kali ketika perangkat lunak Optimasi Komposisi Pupuk Tanaman Tumpangsari Menggunakan Particle Swarm Optimization dijalankan. Pada halaman ini terdapat tiga menu yaitu menu Optimasi Pupuk, Tampilkan Data dan Keluar. Rancangan halaman hasil dapat dilihat pada Gambar 4.9.



Gambar 4.9 Rancangan Antarmuka Halaman Beranda

Keterangan Gambar 4.9:

1. Judul aplikasi.
2. *Button* untuk menuju ke halaman optimasi pupuk.
3. *Button* untuk menuju ke halaman data.
4. *Button* untuk menutup aplikasi.

4.4.2 Halaman Optimasi Pupuk

Halaman Optimasi Pupuk merupakan halaman di mana fungsi utama dari aplikasi yaitu proses optimasi komposisi pupuk dijalankan. Pada halaman ini terdapat kolom masukan yang dibedakan menjadi tiga kategori yaitu Kondisi Tumpangsari, Kondisi Pupuk dan Parameter *Particle Swarm Optimization*. Selain itu terdapat tombol untuk mengendalikan aplikasi yaitu tombol Submit dan Reset serta empat *tab* untuk menampilkan tabel optimasi pupuk. Rancangan halaman hasil dapat dilihat pada Gambar 4.10.

Optimasi Komposisi Pupuk Tumpangsari

Kondisi Tumpangsari

Kabupaten - Pilih Kabupaten -

Kecamatan - Pilih Kecamatan -

Kondisi Pupuk

Harga Urea

Harga SP-36

Harga KCL

Parameter Particle Swarm Optimization

Jumlah Iterasi Wmin C1i C2i

Jumlah Partikel Wmax C1f C2f

Submit Reset

Particle	pBest	gBest	Kecepatan
Generasi ke- - Pilih Generasi -			

Gambar 4.10 Rancangan Antarmuka Halaman Optimasi Pupuk

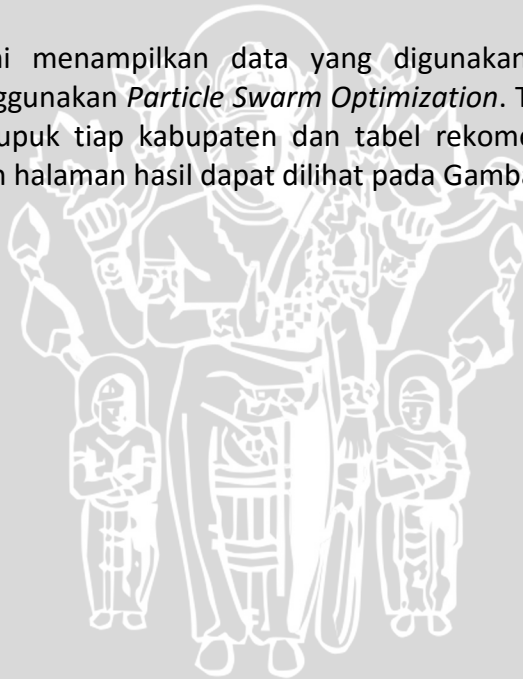
Keterangan Gambar 4.10:

1. Judul halaman.
2. Panel untuk masukan kategori Kondisi Tumpangsari.
3. *Combo box* untuk memilih kabupaten.
4. *Combo box* untuk memilih kecamatan.
5. Panel untuk masukan kategori Kondisi Pupuk.
6. *Text Area* untuk harga pupuk Urea.
7. *Text Area* untuk harga pupuk SP-36.
8. *Text Area* untuk harga pupuk KCL.
9. Panel untuk masukan kategori Parameter Particle Swarm Optimization.
10. *Text Area* untuk jumlah iterasi proses Particle Swarm Optimization.
11. *Text Area* untuk jumlah partikel Particle Swarm Optimization.

12. *Text Area* untuk nilai inersia minimum.
13. *Text Area* untuk nilai inersia maksimum.
14. *Text Area* untuk nilai $c1i$.
15. *Text Area* untuk nilai $c1f$.
16. *Text Area* untuk nilai $c2i$.
17. *Text Area* untuk nilai $c2f$.
18. *Button* untuk memulai proses perhitungan optimasi pupuk.
19. *Button* untuk mereset seluruh masukan ke nilai awal.
20. *Tab* untuk memilih tabel yang ditampilkan.
21. *Combo box* untuk memilih generasi yang ditampilkan ke dalam tabel.
22. Tabel dimana hasil perhitungan ditampilkan.

4.4.3 Halaman Data

Halaman data ini menampilkan data yang digunakan dalam optimasi komposisi pupuk menggunakan *Particle Swarm Optimization*. Terdapat dua tabel yaitu tabel interval pupuk tiap kabupaten dan tabel rekomendasi pupuk tiap kecamatan. Rancangan halaman hasil dapat dilihat pada Gambar 4.11.



The wireframe shows a web page layout for fertilizer data. It includes a header area (1), a table for fertilizer ranges (2) with a large empty space below it (3), a table for fertilizer recommendations (4), two dropdown menus for selecting regency (5) and district (6), and a large empty table area for recommendations (7).

Gambar 4.11 Rancangan Antarmuka Halaman Beranda

Keterangan Gambar 4.11:

1. Judul halaman.
2. Nama tabel untuk Tabel Range Pupuk.
3. Tabel interval pupuk.
4. Nama tabel untuk Tabel Rekomendasi Pupuk.
5. *Combo box* untuk memilih kabupaten yang ditampilkan ke tabel Tabel Rekomendasi Pupuk.
6. *Combo box* untuk memilih kecamatan yang ditampilkan ke tabel Tabel Rekomendasi Pupuk.
7. Tabel rekomendasi pupuk.

4.5 Skenario Pengujian

Setelah dilakukan perancangan sistem, maka langkah selanjutnya adalah membuat perancangan uji coba dan evaluasi dari sistem. Pengujian sistem berfungsi untuk mengetahui tingkat akurasi terhadap penerapan metode *Particle Swarm Optimization* dalam sistem. Terdapat dua skenario pengujian yang dilakukan, yaitu pengujian pengaruh ukuran populasi dan pengujian pengaruh jumlah iterasi. Skenario pengujian yang akan dilakukan dijelaskan pada sub-bab 4.3.1 hingga 4.3.4.

4.5.1 Pengujian Jumlah Partikel

Pengujian jumlah populasi dilakukan untuk mengetahui jumlah populasi yang tepat sehingga dapat menghasilkan komposisi nutrisi pupuk yang optimal. Jumlah populasi yang digunakan untuk pengujian adalah kelipatan 10 yaitu 10, 20, 30, 40, 50, 60, 70, 80, 90 dan 100 dimana pada setiap kelipatan dilakukan pengujian sebanyak 10 kali. Rancangan pengujian populasi dapat dilihat pada Tabel 4.13.

Tabel 4.13 Rancangan Pengujian Jumlah Partikel

Ukuran Populasi	Nilai <i>Fitness</i>										Rata- rata <i>Fitness</i>
	Percobaan ke- <i>i</i>										
	1	2	3	4	5	6	7	8	9	10	
10											
20											
30											
40											
50											
60											
70											
80											
90											
100											

4.5.2 Pengujian Jumlah Iterasi

Pengujian jumlah iterasi merupakan pengujian yang dilakukan untuk mengetahui jumlah iterasi yang tepat untuk menghasilkan komposisi nutrisi pupuk yang optimal. Jumlah iterasi yang digunakan untuk pengujian adalah kelipatan 10 yaitu 10, 20, 30, 40, 50, 60, 70, 80, 90 dan 100 dimana pada setiap kelipatan dilakukan pengujian sebanyak 10 kali. Rancangan pengujian jumlah iterasi dapat dilihat pada Tabel 4.14.

Tabel 4.14 Rancangan Pengujian Jumlah Iterasi

Banyak Iterasi	Nilai <i>Fitness</i>										Rata-rata <i>Fitness</i>
	Percobaan ke- <i>i</i>										
	1	2	3	4	5	6	7	8	9	10	
10											
20											
30											
40											
50											
60											
70											
80											
90											
100											

4.5.3 Pengujian Batas Bobot Inersia

Pengujian batas bobot inersia merupakan pengujian yang dilakukan untuk mengetahui batas bobot inersia yang tepat untuk menghasilkan komposisi nutrisi pupuk yang optimal. Batas bobot inersia yang diuji yaitu w_{min} dengan nilai 0,1 sampai 0,8 dan w_{max} dengan nilai 0,2, sampai 0,9. Rancangan pengujian batas bobot inersia dapat dilihat pada Tabel 4.15.

Tabel 4.15 Rancangan Pengujian Batas Bobot Inersia

		Inertia Min							
		0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
Inertia Max	0,2								
	0,3								
	0,4								
	0,5								
	0,6								
	0,7								
	0,8								
	0,9								

4.5.4 Pengujian Konvergensi

Pengujian jumlah iterasi merupakan pengujian yang dilakukan untuk mengetahui pada jumlah iterasi seberapa konvergensi terjadi. Data yang digunakan untuk pengujian adalah data dari lima kabupaten Rancangan pengujian konvergensi dapat dilihat pada Tabel 4.16.

Tabel 4.16 Rancangan Pengujian Konvergensi

Jumlah Iterasi	Nilai Fitness				
	Daerah 1	Daerah 2	Daerah 3	Daerah 4	Daerah 5
10					
20					
30					
40					
50					
60					
70					
80					
90					
100					

BAB 5 IMPLEMENTASI

5.1 Lingkungan Implementasi

Untuk melakukan proses implementasi perangkat lunak, perlu disiapkan lingkungan implementasi yang mendukung perancangan perangkat lunak tersebut. Lingkungan implementasi dijelaskan pada sub-bab lingkungan perangkat keras dan lingkungan perangkat lunak.

5.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini memiliki spesifikasi sebagai berikut:

1. *Processor Intel(R) Core(TM) i3-2330M CPU @ 2,2 GHz.*
2. *Memory RAM 4 GB.*
3. *Hardisk dengan kapasitas 750 GB.*
4. *Monitor 14 inch.*
5. *Keyboard.*

5.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan perangkat lunak dalam penelitian ini adalah sebagai berikut:

1. *Sistem Operasi Windows 10 Pro 64-bit.*
2. *NetBeans IDE 8.1 digunakan untuk membangun aplikasi sistem rekomendasi dan optimasi komposisi pupuk tanaman tumpangsari.*
3. *Microsoft Office Excel 2016 digunakan untuk melakukan proses manualisasi perhitungan *Particle Swarm Optimization* (PSO).*
4. *Microsoft Office Word 2016 digunakan untuk menyusun laporan penelitian.*
5. *Microsoft Visio 2016 digunakan untuk membuat diagram alir proses kera sistem.*

5.2 Batasan Implementasi

Pada implementasi metode *Particle Swarm Optimization* untuk optimasi komposisi pupuk pada pola tanam tumpangsari ini terdapat beberapa batasan yaitu sebagai berikut:

1. Terdapat dua jenis tanaman tumpangsari yaitu jagung dan kedelai.
2. Unsur hara yang dipertimbangkan untuk rekomendasi komposisi pupuk adalah unsur N, P_2O_5 dan K_2O .
3. Data pupuk rekomendasi yang digunakan berdasarkan kondisi tanah enam kabupaten yang telah dilakukan penelitian sebelumnya yang terdiri dari:
 - a. Bojonegoro
 - b. Gresik
 - c. Lamongan
 - d. Madiun
 - e. Ngawi
 - f. Tulungagung

4. Jenis pupuk yang digunakan adalah pupuk Urea, SP-36 dan KCL.

5.3 Implementasi Program

Berdasarkan perancangan sistem yang telah ditentukan pada bab sebelumnya pada sub-bab ini dijelaskan bagaimana aplikasi Optimasi Komposisi Pupuk Untuk Sistem Penanaman Tumpangsari Menggunakan Algoritma *Particle Swarm Optimization* diterapkan. Pada kelas *Particle Swarm Optimization* terdapat lima fungsi utama yang terdapat dalam implementasi sistem ini yaitu inisialisasi, evaluasi, pencarian *gBest*, pencarian *pBest*, memperbarui kecepatan dan memperbarui posisi. Berdasarkan diagram alir pada sub-bab perancangan perangkat lunak berikut ini adalah penjelasan mengenai fungsi yang diimplementasikan pada sistem ini.

5.3.1 Implementasi Proses Inisialisasi

Berikut merupakan hasil implementasi dari proses inisialisasi *Particle Swarm Optimization*.

```

1 public double[][] inisialisasi(int jumlahPartikel, int[][]
  batasPartikel) {
2     double[][] temp = new
  double[jumlahPartikel][batasPartikel.length + 3];
3     for (int i = 0; i < jumlahPartikel; i++) {
4         for (int j = 0; j < batasPartikel.length; j++)
5         {
6             temp[i][j] = batasPartikel[j][0] +
  Math.random() * (batasPartikel[j][1] -
  batasPartikel[j][0]);
7         }
8         temp[i][batasPartikel.length] = 0;
9         temp[i][batasPartikel.length + 1] = 0;
10        temp[i][batasPartikel.length + 2] = 0;
11    }
12    return temp;
  }
```

Kode Program 5.1 Inisialisasi Partikel

Penjelasan kode program 5.1 adalah sebagai berikut:

1. Baris 2 berfungsi untuk membuat *array* yang digunakan untuk menampung sementara partikel yang dibangkitkan.
2. Baris 3 berfungsi untuk melakukan perulangan untuk pembangkitan partikel sebanyak jumlah partikel.
3. Baris 4 berfungsi untuk melakukan perulangan untuk pembangkitan nilai sebanyak dimensi partikel.
4. Baris 5 berfungsi untuk mengacak nilai di antara batas pupuk yang sesuai.
5. Baris 7 - 9 berfungsi untuk menyiapkan indeks tempat menyimpan hasil evaluasi.
6. Baris 11 untuk memberikan hasil inisialisasi.

5.3.2 Implementasi Proses Evaluasi / Menghitung *Fitness*

Berikut merupakan hasil implementasi dari proses evaluasi / menghitung *fitness Paricle Swarm Optimization*.

```

1  for (int j = 0; j < jumlahPartikel; j++) {
2      partikel[j][partikel[j].length - 3] = 0;
3      partikel[j][partikel[j].length - 2] = 0;
4      for (int k = 0; k <
        rekomendasiKecamatan[0][0].length; k++) {
5          partikel[j][partikel[j].length - 3] +=
            Math.abs(rekomendasiKecamatan[kabupaten][kecamatan][k]
              - partikel[j][k]) * 10000;
6          partikel[j][partikel[j].length - 2] +=
            (partikel[j][k] * hargaPupuk[pupuk[k%3]]);
7      }
8      partikel[j][partikel[j].length - 1] = 10000000 /
            (partikel[j][partikel[j].length - 3] +
              partikel[j][partikel[j].length - 2]);
9  }

```

Kode Program 5.2 Proses Evaluasi

Penjelasan kode program 5.2 adalah sebagai berikut:

1. Baris 1 berfungsi untuk melakukan perulangan sebanyak jumlah partikel.
2. Baris 2-3 berfungsi untuk me-reset nilai pada indeks tempat penyimpanan nilai evaluasi menjadi 0.
3. Baris 4 berfungsi untuk melakukan perulangan sebanyak jumlah dimensi.
4. Baris 5 berfungsi untuk menghitung selisih nilai yang telah dibangkitkan dengan rekomendasi pakar.
5. Baris 6 berfungsi untuk menghitung total biaya yang dikeluarkan.
6. Baris 8 berfungsi untuk menghitung nilai *fitness*.

5.3.3 Implementasi Proses Pencarian *pBest*

Berikut merupakan hasil implementasi dari proses pencarian *pBest*.

```

1  public double[][] cariPBest(double[][] partikel,
2      double[][] pBest) {
3      double[][] temp = new
4      double[pBest.length][pBest[0].length];
5      for (int i = 0; i < partikel.length; i++) {
6          if(partikel[i][partikel[i].length - 1] >
7          pBest[i][pBest[i].length - 1]) {
8              temp[i] = partikel[i];
9          } else {
10             temp[i] = pBest[i];
11         }
12     }
13     return temp;
14 }

```

Kode Program 5.3 Proses Pencarian *pBest*

Penjelasan kode program 5.3 adalah sebagai berikut:

1. Baris 2 berfungsi untuk membuat *array* yang digunakan untuk menampung sementara partikel *pBest*.
2. Baris 3 berfungsi untuk melakukan perulangan sebanyak jumlah partikel.
3. Baris 4 berfungsi untuk mencari *pBest* dengan membandingkan nilai *fitness*.
4. Baris 5-7 berfungsi untuk memasukan partikel dengan *fitness* terbaik ke array sementara *pBest*.
5. Baris 10 berfungsi untuk memberikan hasil array *pbest*.

5.3.4 Implementasi Proses Pencarian *gBest*

Berikut merupakan hasil implementasi dari proses pencarian *gBest Paricle Swarm Optimization*.

1	public int cariGBest(double[][] partikel) {
	int index = 0;
2	for (int i = 1; i < partikel.length; i++) {
3	if(partikel[i][partikel[i].length - 1] >
4	partikel[index][partikel[i].length - 1]) {
	index = i;
5	}
6	}
7	return index;
8	}
9	

Kode Program 5.4 Inisialisasi Proses Pencarian *gBest*

Penjelasan kode program 5.4 adalah sebagai berikut:

1. Baris 2 berfungsi untuk membuat variabel untuk menampung nilai yang merupakan indeks dari *gBest*.
2. Baris 3 berfungsi untuk melakukan perulangan sebanyak jumlah partikel.
3. Baris 4 berfungsi untuk membandingkan nilai *fitness*.
4. Baris 5 berfungsi untuk memberikan hasil index *gbest*.

5.3.5 Implementasi Proses Memperbarui Kecepatan

Berikut merupakan hasil implementasi dari proses memperbarui kecepatan *Paricle Swarm Optimization*.

1	public double[][] updateKecepatan(double[][] partikel,
	double[][] batasKecepatan, double[][] pBest, double[]
	gBest, double[][] kecepatan, double inersia, double[]
	c, double[] r) {
2	double[][] temp = new
	double[kecepatan.length][kecepatan[0].length];
3	for (int i = 0; i < partikel.length; i++) {
4	for (int j = 0; j < kecepatan[i].length; j++)
	{
5	

```

        temp[i][j] = inersia * kecepatan[i][j] +
c[0] * r[0] * (pBest[i][j] - partikel[i][j]) + c[1] *
6      r[1] * (gBest[j] - partikel[i][j]);
7      if(temp[i][j] < batasKecepatan[j][0]) {
8          temp[i][j] = batasKecepatan[j][0];
9      } else if(temp[i][j] >
10     batasKecepatan[j][1]) {
11         temp[i][j] = batasKecepatan[j][1];
12     }
13 }
14 return temp;
}

```

Kode Program 5.5 Memperbarui Kecepatan

Penjelasan kode program 5.5 adalah sebagai berikut:

1. Baris 2 berfungsi untuk membuat *array* yang digunakan untuk menampung sementara kecepatan partikel.
2. Baris 3 berfungsi untuk melakukan perulangan sebanyak jumlah partikel.
3. Baris 4 berfungsi untuk melakukan perulangan sebanyak jumlah dimensi.
4. Baris 5 berfungsi untuk menghitung kecepatan partikel.
5. Baris 6-10 berfungsi untuk membatasi kecepatan partikel.
6. Baris 13 berfungsi untuk memberikan *array* kecepatan yang didapatkan.

5.3.6 Implementasi Proses Memperbarui Posisi

Berikut merupakan hasil implementasi dari proses memperbarui posisi *Paricle Swarm Optimization*.

```

1  public double[][] updatePosisi(double[][] partikel,
2     double[][] kecepatan, int[][] batasPartikel) {
3     double[][] temp = partikel;
4     for (int i = 0; i < partikel.length; i++) {
5         for (int j = 0; j < kecepatan[i].length; j++)
6         {
7             temp[i][j] = partikel[i][j] +
8             kecepatan[i][j];
9             if(temp[i][j] < batasPartikel[j][0]) {
10                 temp[i][j] = batasPartikel[j][0];
11             } else if(temp[i][j] >
12             batasPartikel[j][1]) {
13                 temp[i][j] = batasPartikel[j][1];
14             }
15         }
16     }
17     return temp;
18 }

```

Kode Program 5.6 Proses Memperbarui Posisi

Penjelasan kode program 5.6 adalah sebagai berikut:

1. Baris 2 berfungsi untuk membuat *array* yang digunakan untuk menampung sementara partikel yang telah diperbarui.

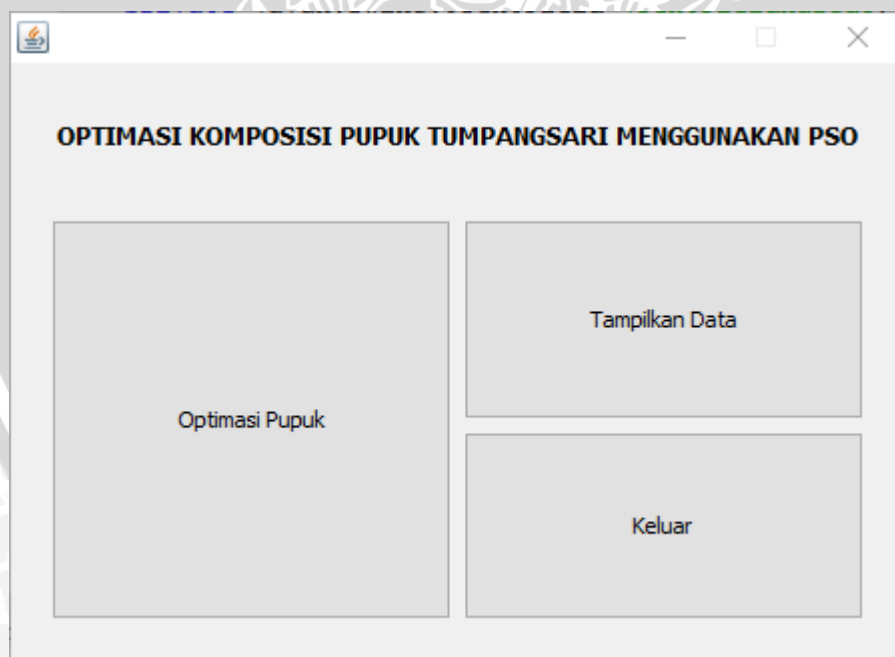
2. Baris 3 berfungsi untuk melakukan perulangan sebanyak jumlah partikel.
3. Baris 4 berfungsi untuk melakukan perulangan sebanyak jumlah dimensi.
4. Baris 5 berfungsi untuk memperbarui nilai posisi partikel.
5. Baris 6-10 berfungsi untuk membatasi nilai posisi partikel.
6. Baris 13 berfungsi untuk memberikan *array* partikel yang telah diperbarui.

5.4 Implementasi Antarmuka

Berdasarkan perancangan antarmuka yang telah ditentukan pada bab sebelumnya aplikasi Optimasi Komposisi Pupuk Untuk Sistem Penanaman Tumpangsari Menggunakan Algoritma *Particle Swarm Optimization* memiliki tiga halaman yaitu halaman Beranda, halaman Optimasi Pupuk dan halaman Data. Penjelasan mengenai implementasi antarmuka sistem akan dijelaskan pada sub bab 5.4.1 sampai 5.4.3.

5.4.1 Implementasi Antarmuka Halaman Beranda

Halaman Beranda berfungsi sebagai tampilan utama yang akan ditampilkan pertama kali ketika perangkat lunak dijalankan. Pada halaman ini terdapat judul aplikasi yang terdapat pada bagian atas halaman kemudian diikuti oleh tiga menu *button* di bawahnya. *Button* “Optimasi Pupuk” dan “Tampilkan Data” berfungsi untuk membuka halaman Otimasi Pupuk dan halaman Data sedangkan *button* “Keluar” berfungsi ntuk mengakhiri aplikasi.



Gambar 5.1 Implementasi Antarmuka Halaman Beranda

5.4.2 Implementasi Antarmuka Halaman Optimasi Pupuk

Pada halaman Optimasi Pupuk fungsi utama dari aplikasi yaitu proses optimasi komposisi pupuk dijalankan. Pada halaman ini terdapat semua kolom masukan yang akan digunakan sebagai parameter optimasi serta tampilan

perhitungan dan hasil optimasi komposisi pupuk. Kolom masukan dibedakan menjadi tiga kategori yaitu Kondisi Tumpangsari, Kondisi Pupuk dan Parameter *Particle Swarm Optimization*. Selain itu terdapat tombol untuk mengendalikan aplikasi yaitu tombol Submit dan Reset serta empat *tab* untuk menampilkan tabel optimasi pupuk.

Gambar 5.2 Implementasi Antarmuka Halaman Optimasi Pupuk

5.4.3 Implementasi Antarmuka Halaman Data

Pada halaman Data terdapat beberapa tabel untuk menampilkan data yang digunakan dalam optimasi komposisi pupuk menggunakan *Particle Swarm Optimization*. Terdapat judul halaman perangkat lunak yang terletak pada bagian atas halaman. Terdapat dua tabel yaitu tabel interval pupuk tiap kabupaten dan tabel rekomendasi pupuk tiap kecamatan. Selain itu terdapat *combo box* untuk

memilih kabupaten yang akan menentukan data rekomendasi pupuk tiap kecamatan yang ditampilkan pada tabel rekomendasi pupuk.

Data Pupuk

Tabel Range Pupuk

No	Kot...	Ure...	SP...	KCL...	Ure...	SP...	KCL...
1	Gresik	350-...	100-...	50-75	50-50	50-100	50-75
2	Madiun	350-...	100-...	50-100	50-50	50-100	50-100
3	Bojon...	350-...	100-...	50-100	50-50	50-100	50-100
4	Lamo...	350-...	100-...	50-75	50-50	50-100	50-75
5	Ngawi	350-	100-	50-100	50-50	50-75	50-100

Tabel Rekomendasi Pupuk

Kabupaten

No	Kec...	Ure...	SP...	KCL...	Ure...	SP...	KCL...
1	Balon...	350	125	50	50	75	50
2	Benjeng	350	125	50	50	75	50
3	Bunga	350	100	75	50	50	75
4	Cerme	350	125	75	50	75	75
5	Driyo...	350	125	75	50	75	75
6	Dudu...	350	125	50	50	75	50
7	Dukun	350	100	50	50	50	50
8	Kebo...	350	125	50	50	75	50

Gambar 5.3 Implementasi Antarmuka Halaman Data

BAB 6 PENGUJIAN DAN ANALISIS

Pengujian dan analisis membahas mengenai pengujian dan analisis yang dilakukan terhadap aplikasi yang telah dibuat. Pengujian yang dilakukan berdasarkan atas skenario pengujian yang telah dirancang pada Bab 4.

6.1 Pengujian

Pada sub-bab ini menjelaskan mengenai pengujian aplikasi yang telah diimplementasikan. Pengujian yang dilakukan berdasarkan atas perancangan skenario pengujian sebagai berikut:

- Pengujian jumlah partikel terbaik.
- Pengujian jumlah iterasi terbaik.
- Pengujian batas inersia terbaik.
- Pengujian konvergensi.

6.2 Hasil Pengujian

Berdasarkan rancangan pengujian yang telah dibuat pada Bab 4, maka hasil pengujian dari aplikasi dijelaskan pada sub-bab 6.2.1 hingga sub-bab 6.2.4.

6.2.1 Hasil Pengujian Terhadap Jumlah Partikel Terbaik

Pada sub-bab ini dilakukan pengujian terhadap jumlah populasi untuk mendapatkan jumlah populasi yang terbaik berdasarkan perancangan skenario pengujian 4.4.1. Data yang digunakan dalam pengujian ini adalah tumpangsari tanaman jagung dan kacang kedelai pada Kabupaten Bojonegoro Kecamatan Balen, jumlah iterasi yang digunakan adalah sebanyak 10 iterasi, batas inersia yaitu w_{min} sebesar 0,4 dan w_{max} sebesar 0,9 dan komponen sosial yaitu $c1f$ sebesar 0,5, $c1i$ sebesar 2,5, $c2f$ sebesar 2,5 dan $c2i$ sebesar 0,5. Jumlah partikel yang diuji adalah kelipatan 10, dimulai dari ukuran populasi 10 hingga 100. Pengujian ini dilakukan sebanyak 10 kali pada setiap generasi. Untuk detail hasil pengujian jumlah partikel terbaik dapat dilihat pada lampiran 4.

Tabel 6.1 Hasil Pengujian Jumlah Partikel Terbaik

Ukuran Populasi	Nilai <i>Fitness</i>							Rata-rata <i>Fitness</i>
	Percobaan ke- <i>i</i>							
	1	2	3		8	9	10	
10	4,8891205	4,7778706	5,0046370		4,9054511	4,8626360	4,8424822	4,884249
20	4,7787748	4,9729592	4,8803624		5,0616170	5,0178667	5,0069274	4,941107
30	5,0477749	4,9731687	4,9380996		4,9302675	5,0313202	5,0233450	4,980439
40	4,9037123	4,9718075	4,9751606		4,8848010	5,0045028	4,9672824	4,978390
50	4,9928285	5,0430267	4,8124958		5,0019870	4,9611931	5,0396336	5,006354
60	4,9495664	5,0715169	4,9670526		5,0295276	5,0302861	5,0024315	5,012858
70	5,0911072	4,9167149	4,9546722		4,9923132	5,0537612	5,0629481	5,019746
80	5,0036750	5,0631055	4,9968085		5,0713228	5,0781540	5,0683818	5,040307

90	5,0386842	5,0781757	5,0416350		5,0095429	5,0662036	5,0206718	5,037828
100	5,0471878	5,0545701	5,0490090		5,0125496	5,0447538	5,0158663	5,038935

6.2.2 Hasil Pengujian Terhadap Jumlah Iterasi Terbaik

Pada sub-bab ini dilakukan pengujian terhadap ukuran iterasi untuk mendapatkan ukuran iterasi yang terbaik berdasarkan perancangan skenario pengujian 4.4.2. Data yang digunakan dalam pengujian ini adalah tumpangsari tanaman jagung dan kacang kedelai pada Kabupaten Bojonegoro Kecamatan Balen, dengan batas inersia yaitu w_{min} sebesar 0,4 dan w_{max} sebesar 0,9 dan komponen sosial yaitu $c1f$ sebesar 0,5, $c1i$ sebesar 2,5, $c2f$ sebesar 2,5 dan $c2i$ sebesar 0,5. Sedangkan jumlah partikel terbaik yang telah dilakukan pengujian sebelumnya, yaitu sebesar 80. Jumlah iterasi yang diuji adalah kelipatan 10, dimulai dari ukuran populasi 10 hingga 100. Pengujian ini dilakukan sebanyak 10 kali pada setiap iterasi. Untuk detail hasil pengujian jumlah iterasi terbaik dapat dilihat pada lampiran 4.

Tabel 6.2 Hasil Pengujian Jumlah Iterasi Terbaik

Ukuran Populasi	Nilai <i>Fitness</i>							Rata-rata <i>Fitness</i>
	Percobaan ke- <i>i</i>							
	1	2	3		8	9	10	
10	5,0036750	5,0631055	4,9968085		5,0713228	5,0781540	5,0683818	5,040307
20	5,0979713	5,0909443	5,0891716		5,0933610	5,0946594	5,0777039	5,085648
30	5,0995523	5,0991298	5,0958892		5,1000237	5,0935641	5,0783689	5,096314
40	5,0996282	5,0980768	5,1012609		5,1013531	5,0896234	5,1019356	5,098784
50	5,1011297	5,1013916	5,1018066		5,1009841	5,1016635	5,1018541	5,100301
60	5,1012255	5,1020140	5,1007630		5,1014771	5,1009069	5,1017364	5,101217
70	5,1019272	5,1020385	5,0953627		5,1018879	5,1020275	5,1017568	5,101224
80	5,1019068	5,1019440	5,0765977		5,1029034	5,1019164	5,1019692	5,099511
90	5,1020380	5,1000378	5,0849009		5,1019997	5,1019997	5,1020356	5,099851
100	5,1020380	5,0914444	5,1020396		5,1020407	5,1019997	5,1020356	5,100917

6.2.3 Hasil Pengujian Batas Inersia Terbaik

Pada sub-bab ini dilakukan pengujian terhadap jumlah populasi untuk mendapatkan jumlah populasi yang terbaik berdasarkan perancangan skenario pengujian 4.4.3. Data yang digunakan dalam pengujian ini adalah tumpangsari tanaman jagung dan kacang kedelai pada Kabupaten Bojonegoro Kecamatan Balen, dengan komponen sosial yaitu $c1f$ sebesar 0,5, $c1i$ sebesar 2,5, $c2f$ sebesar 2,5 dan $c2i$ sebesar 0,5. Sedangkan jumlah partikel dan jumlah iterasi yang digunakan merupakan jumlah partikel dan jumlah iterasi terbaik hasil pengujian sebelumnya, yaitu jumlah partikel sebesar 80 dan jumlah iterasi 70. Jumlah bobot inersia yang diuji adalah kelipatan 0,1 dimulai dari 0,1 sampai 0,9.

Tabel 6.3 Hasil Pengujian Batas Inersia Terbaik

		Inertia Min							
		0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
Inertia Max	0,2	5,035012	-	-	-	-	-	-	-
	0,3	5,097070	5,100097	-	-	-	-	-	-
	0,4	5,101759	5,101739	5,092063	-	-	-	-	-
	0,5	5,102022	5,100950	5,085575	5,102038	-	-	-	-
	0,6	5,102032	5,101744	5,102038	5,102040	5,102039	-	-	-
	0,7	5,102037	5,101993	5,100978	5,101881	5,101743	5,101724	-	-
	0,8	5,101922	5,102026	5,100280	5,102021	5,102033	5,100648	5,101217	-
	0,9	5,102038	5,101937	5,097968	5,101925	5,101897	5,101554	5,089894	5,100875

6.2.4 Hasil Pengujian Konvergensi

Pada sub-bab ini dilakukan pengujian terhadap jumlah populasi untuk mendapatkan jumlah populasi yang terbaik berdasarkan perancangan skenario pengujian 4.4.4. Untuk pengujian konvergensi data diuji menggunakan 5 daerah yang berbeda seperti yang tertera pada Tabel 6.4 di bawah ini.

Tabel 6.4 Data Daerah yang Diuji

Data ke- <i>i</i>	Daerah Tumpangsari	
	Kabupaten	Kecamatan
Data 1	Bojonegoro	Balen
Data 2	Lamongan	Babat
Data 3	Madiun	Balerejo
Data 4	Ngawi	Bringin
Data 5	Tulungagung	Bandung

Data yang digunakan dalam pengujian ini adalah tanaman jagung dan kacang kedelai, dengan komponen sosial yaitu $c1f$ sebesar 0,5, $c1i$ sebesar 2,5, $c2f$ sebesar 2,5 dan $c2i$ sebesar 0,5. Sedangkan jumlah partikel, jumlah iterasi dan bobot inersia yang digunakan merupakan jumlah partikel, jumlah iterasi dan bobot inersia terbaik hasil pengujian sebelumnya, yaitu jumlah partikel sebesar 80, jumlah iterasi 70 serta batas inersia yaitu w_{min} sebesar 0,4 dan w_{max} sebesar 0,6.

Tabel 6.5 Hasil Pengujian Konvergensi

Jumlah Iterasi	Nilai <i>Fitness</i>				
	Daerah 1	Daerah 2	Daerah 3	Daerah 4	Daerah 5
10	4,942515	5,088906	5,058375	5,087694	5,102041
20	5,049324	5,098773	5,093824	5,099813	5,102041
30	5,100351	5,098579	5,093806	5,101508	5,102041

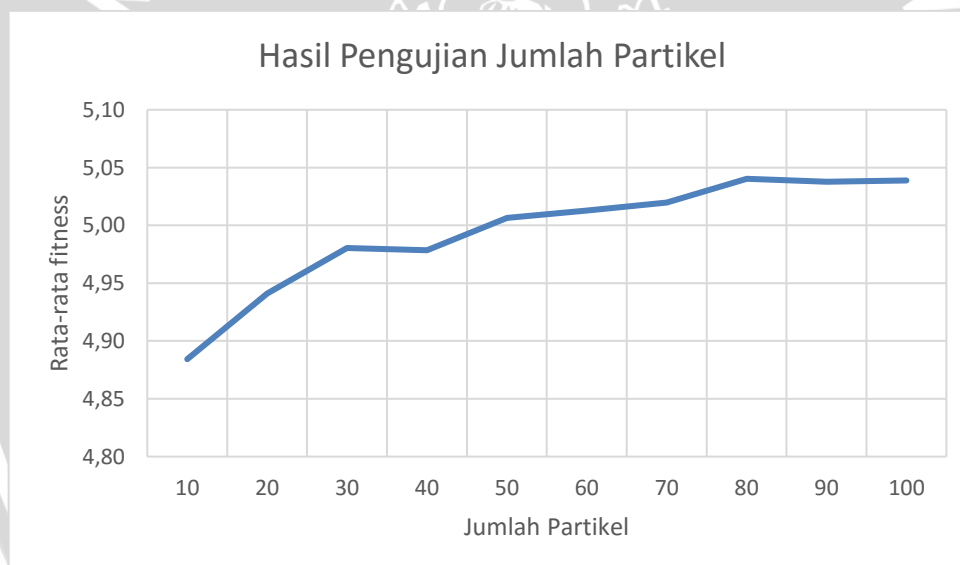
40	5,100190	5,102007	5,101545	5,102030	5,102041
50	5,101838	5,102037	5,100490	5,102037	5,102041
60	5,101964	5,102040	5,101940	5,102038	5,102041
70	5,101970	5,102040	5,102039	5,102041	5,102041
80	5,102040	5,102035	5,102035	5,102041	5,102041
90	5,102040	5,102041	5,102035	5,102041	5,102041
100	5,102031	5,102041	5,102032	5,102041	5,102041

6.3 Analisis Hasil Pengujian

Dari hasil pengujian yang dilakukan sebelumnya, maka pada sub-bab ini menjelaskan hasil analisis dari hasil pengujian yang telah dilakukan tersebut.

6.3.1 Analisis Hasil Pengujian Jumlah Partikel Terbaik

Berdasarkan Tabel 6.1, maka dapat dibuat grafik hasil pengujian terhadap jumlah populasi yang terbaik seperti yang ditunjukkan pada Gambar 6.1.



Gambar 6.1 Analisis Pengujian Jumlah Partikel

Berdasarkan Gambar 6.1, grafik menunjukkan bahwa semakin banyak jumlah partikel maka semakin besar nilai rata-rata *fitness* yang didapatkan. Hal ini dikarenakan semakin banyak jumlah partikel yang digunakan semakin beragam nilai atau solusi yang dihasilkan. Keragaman solusi ini membuat area pencarian pada *Particle Swarm Optimization* semakin besar sehingga mempermudah sistem untuk mendapatkan hasil yang lebih optimal. Percobaan dengan jumlah partikel 10 memiliki nilai rata-rata *fitness* yang paling rendah, yaitu sebesar 4,884249. Sedangkan nilai rata-rata *fitness* paling tinggi terjadi pada percobaan dengan jumlah populasi lebih dari 80 yaitu dengan *fitness* sebesar 5,040307.

6.3.2 Analisis Hasil Pengujian Jumlah Iterasi Terbaik

Berdasarkan Tabel 6.2, maka dapat dibuat grafik hasil pengujian terhadap ukuran iterasi terbaik seperti yang ditunjukkan pada Gambar 6.2.



Gambar 6.2 Analisis Pengujian Jumlah Iterasi Terbaik

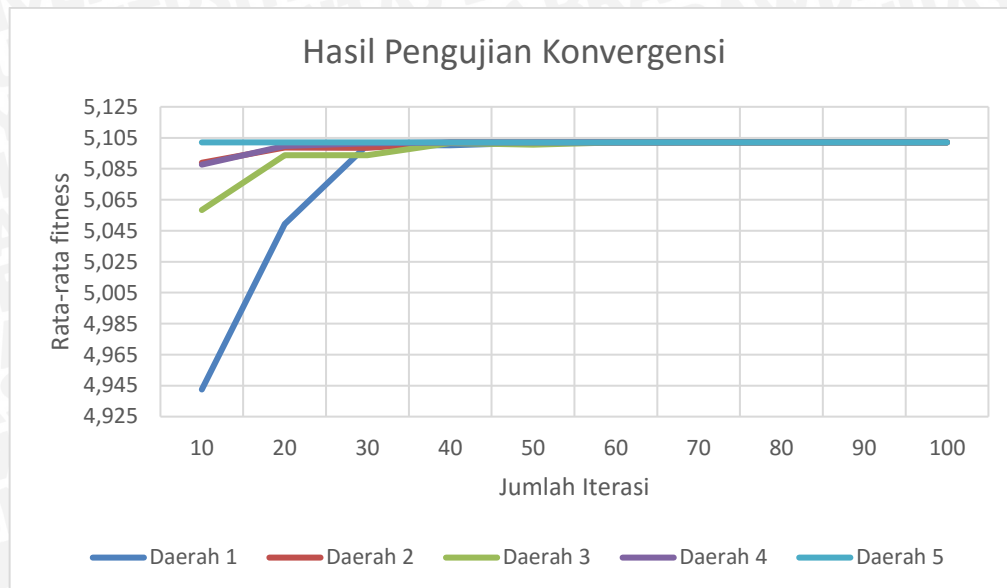
Berdasarkan Gambar 6.2, grafik menunjukkan bahwa semakin banyak jumlah iterasi maka semakin besar nilai rata-rata *fitness* yang didapatkan. Hal ini dikarenakan *Particle Swarm Optimization* merupakan algoritma yang secara iteratif mengoptimasi suatu permasalahan dengan memperbaiki kandidat solusinya. Semakin banyak jumlah iterasi yang digunakan, menambah jumlah proses optimasi pada *Particle Swarm Optimization* sehingga mendapatkan hasil yang didapat lebih optimal. Percobaan dengan jumlah iterasi 10 memiliki nilai rata-rata *fitness* yang paling rendah, yaitu sebesar 5,040307. Sedangkan nilai rata-rata *fitness* paling tinggi terjadi pada percobaan dengan jumlah iterasi lebih dari 70 yaitu dengan *fitness* sebesar 5,101224.

6.3.3 Analisis Hasil Pengujian Jumlah Bobot Inersia

Bobot inersia mempengaruhi pencarian pada *Particle Swarm Optimization*. Bobot inersia bernilai besar mempermudah pencarian global sedangkan bobot inersia bernilai kecil mempermudah pencarian lokal. Pada Tabel 6.3 menunjukkan bahwa batas inersia yaitu w_{min} sebesar 0,1 dan w_{max} sebesar 0,2 menghasilkan nilai *fitness* yang paling rendah, yaitu sebesar 5,08E-07. Sedangkan nilai rata-rata *fitness* paling tinggi terjadi pada percobaan dengan w_{min} sebesar 0,4 dan w_{max} sebesar 0,6 yaitu dengan *fitness* sebesar 5,102040.

6.3.4 Analisis Hasil Pengujian Konvergensi

Berdasarkan Tabel 6.5, maka dapat dibuat grafik hasil pengujian terhadap ukuran iterasi terbaik seperti yang ditunjukkan pada Gambar 6.3.



Gambar 6.3 Analisis Pengujian Konvergensi

Berdasarkan Gambar 6.3, grafik menunjukkan bahwa konvergensi terjadi pada jumlah iterasi sebesar 40. Percobaan dengan parameter Daerah 2, menghasilkan nilai *fitness* terbaik yaitu 5,102040 pada jumlah iterasi 60. Percobaan dengan parameter Daerah 1, 2 dan 3, menghasilkan nilai *fitness* terbaik pada jumlah iterasi 70 yaitu 5,101970, 5,102039 dan 5,102041. Sedangkan data Daerah 5 menghasilkan nilai *fitness* terbaik yaitu 5,102041 pada jumlah iterasi kurang dari 10.

6.3.5 Analisis Keseluruhan Hasil Pengujian

Pengujian keseluruhan sistem digunakan untuk mengetahui kualitas hasil optimasi yang dihasilkan sistem. Sistem melakukan optimasi pupuk pada Kabupaten Tulungagung Kecamatan Sumbergempol, jumlah iterasi yang digunakan adalah sebanyak 10 iterasi dan komponen sosial yaitu $c1f$ sebesar 0,5, $c1i$ sebesar 2,5, $c2f$ sebesar 2,5 dan $c2i$ sebesar 0,5. Sedangkan jumlah partikel dan jumlah iterasi yang digunakan merupakan jumlah partikel, jumlah iterasi dan batas inersia terbaik hasil pengujian sebelumnya, yaitu jumlah partikel sebesar 80, jumlah iterasi 70 serta batas inersia yaitu w_{min} sebesar 0,4 dan w_{max} sebesar 0,6.

Tabel 6.6 Perbandingan Komposisi Pupuk oleh Pakar dan Sistem

Rekomendasi	Rekomendasi Pupuk (Kg/Ha)						Penalti	Total Biaya
	Jagung			Kedelai				
	Urea	SP-36	KCL	Urea	SP-36	KCL		
Pakar	350	125	100	50	75	100	0	1960000
Sistem	350	123,89	99,79	50	72,82	87,21	162900	1880620
Selisih	0	1,11	0,21	0	2,18	12,79	162900	79380

Berdasarkan Tabel 6.6, total biaya yang dibutuhkan untuk komposisi pupuk oleh sistem sebesar Rp 1.880.620,00. Biaya ini lebih murah apabila dibandingkan dengan total biaya yang dibutuhkan untuk memenuhi rekomendasi komposisi pupuk dari pakar yaitu sebesar Rp 1.960.000,00. Dapat disimpulkan dalam pengujian tersebut terdapat penghematan biaya sebesar Rp 79.380,00 atau 4,05% dari biaya pakar dengan bobot penalti sebesar 162.900.

Tabel 6.7 Perbandingan Komposisi Pupuk oleh Petani dan Sistem

Rekomendasi	Rekomendasi Pupuk (Kg/Ha)						Penalti	Total Biaya
	Jagung			Kedelai				
	Urea	SP-36	KCL	Urea	SP-36	KCL		
Petani	420	165	132	66	90	132	2000000	2863200
Sistem	350	123,89	99,79	50	72,82	87,21	162900	1880620
Selisih	70	42,11	32,21	11	17,18	44,79	1837100	982580

Sedangkan Berdasarkan Tabel 6.7, total biaya yang dibutuhkan untuk komposisi pupuk oleh petani sebesar Rp 2.863.200 dengan bobot penalty 2.000.000. Biaya rekomendasi sistem lebih murah apabila dibandingkan dengan total biaya yang dibutuhkan untuk memenuhi rekomendasi komposisi pupuk dari petani dengan penghematan biaya sebesar Rp 982.580,00 atau 34,32% dan perbaikan bobot penalti sebesar 1.837.100.

BAB 7 PENUTUP

Pada bab ini memaparkan kesimpulan serta saran dari hasil penelitian yang telah dilakukan. Kesimpulan akan dijelaskan pada sub-bab 7.1 dan saran akan dijelaskan pada sub-bab 7.2

7.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan berdasarkan perancangan dan implementasi sistem, dapat ditarik kesimpulan sebagai berikut:

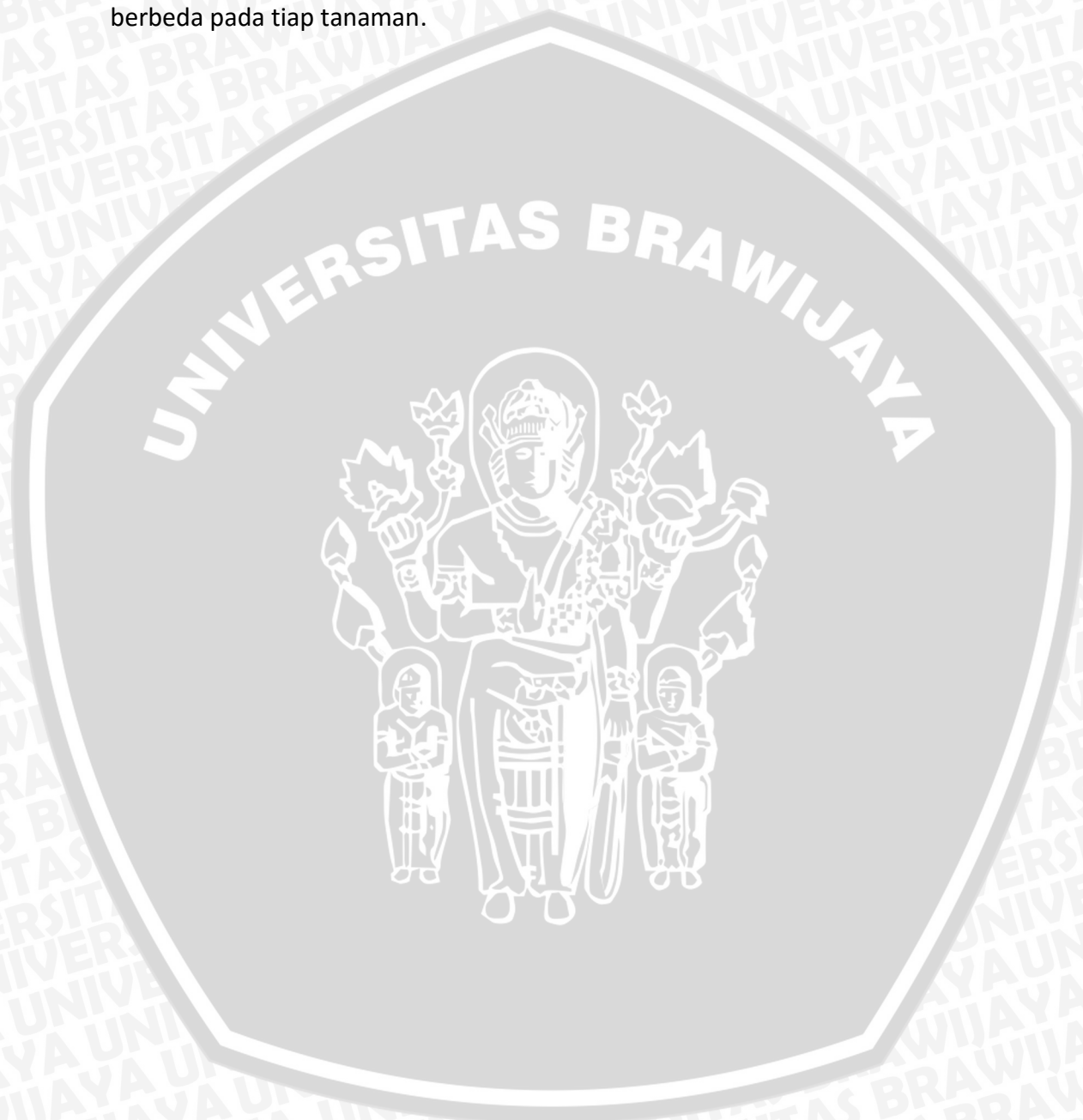
1. Implementasi sistem optimasi komposisi pupuk tanaman tumpangsari dengan menggunakan algoritma Particle Swarm Optimization, dimulai dengan membuat susunan partikel yang terdiri dari pupuk Urea, SP-36 tanaman 1, KCL tanaman 1, Urea tanaman 2, SP-36 tanaman 2 dan KCL tanaman 2. Untuk proses evaluasi, nilai *fitness* dihitung berdasarkan total harga dan selisih rekomendasi. Setelah fitness didapat dilakukan pencarian pBest dan gBest. Untuk melakukan perbaruan kecepatan bobot inersia yang digunakan merupakan *time varying inertia weight* (TVIW) dan *time varying acceleration coefficient* (TVAC). Kemudian partikel diperbarui berdasarkan kecepatan tersebut. Proses optimasi akan berakhir ketika telah mencapai generasi akhir dan menghasilkan kombinasi pupuk optimal.
2. Dari hasil pengujian Particle Swarm Optimazion pada data pupuk tanaman tumpangsari didapat parameter optimal sebagai berikut:
 - a. Jumlah partikel sebesar 80 dan iterasi sebesar 70.
 - b. Batas inersia yaitu w_{min} sebesar 0,4 dan w_{max} sebesar 0,6.
3. Dari perbandingan antara rekomendasi sistem dan rekomendasi pakar, rekomendasi sistem dapat menghemat total biaya yang digunakan untuk komposisi pupuk tanaman tumpangsari sebesar Rp 79.380,00 atau 4,05%. Sedangkan bila dibandingkan dengan komposisi pemupukan yang digunakan oleh petani, sistem dapat menghemat biaya sebesar Rp 982.580,00 atau 34,32% serta perbaikan penalti komposisi pupuk sebesar 1.837.100.

7.2 Saran

Untuk mengembangkan penelitian ini, saran dari penulis adalah dengan mengerjakan beberapa hal atau permasalahan yang belum sepenuhnya dikaji dalam penelitian ini di antaranya:

1. Pada pengujian yang telah dilakukan, partikel telah mencapai titik konvergen pada jumlah iterasi yang relatif sedikit yaitu pada iterasi 40 sehingga terdapat kemungkinan terjadi konvergensi dini. Pada penelitian selanjutnya dapat menggunakan *Random Injection* pada saat memperbarui posisi partikel sehingga mengurangi kemungkinan terjadinya konvergensi dini.
2. Optimasi pupuk yang digunakan dalam penelitian ini terbatas pada tanaman jagung dan kacang kedelai, sehingga pada penelitian selanjutnya dapat ditambahkan variasi tanaman lainnya yang sering digunakan pada pola tanam tumpangsari di Indonesia.

3. Dalam penelitian ini pemupukan masih dibedakan antar tanaman tumpangsari karena diharapkan jarak yang digunakan pada pola tanam tumpangsari cukup baik sehingga pupuk antar tanaman tidak saling mempengaruhi. Pada penelitian selanjutnya dapat menggabungkan pupuk tanaman tumpangsari menjadi satu sehingga dapat meningkatkan efisiensi pemupukan dikarenakan petani tidak perlu menyiapkan kadar pupuk yang berbeda pada tiap tanaman.



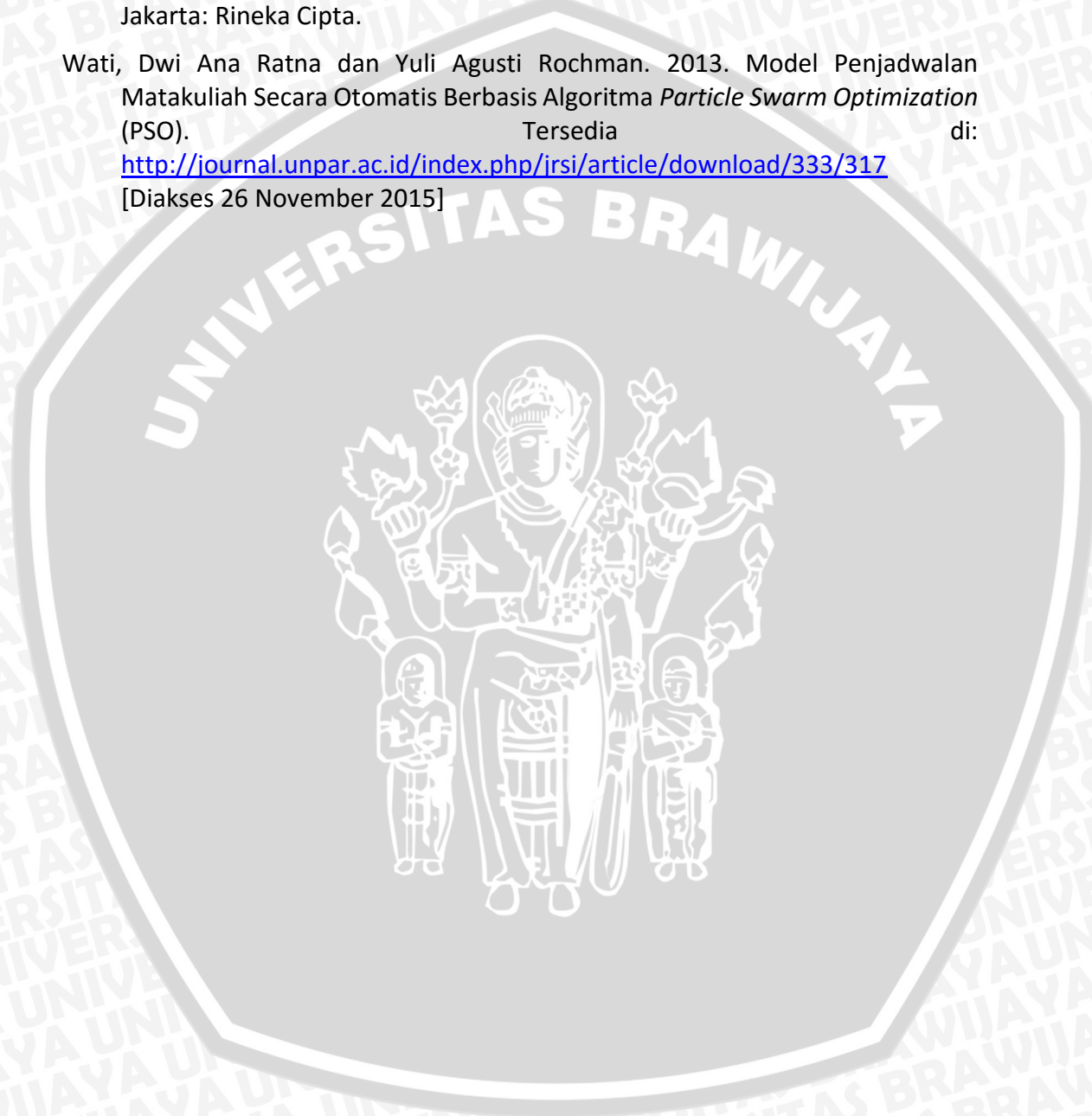
DAFTAR PUSTAKA

- AAK. 1993. Jagung, penerbit Kanisius – Yogyakarta.
- Aminah, Iin Siti, Rosmiah dan M. Haris Yahya. 2014. Efisiensi Pemanfaatan Lahan pada Tumpangsari Jagung (*Zea mays L.*) dan Kedelai (*Glycine Max L. Merrill*) di Lahan Pasang Surut. Tersedia di: www.pur-plso-unsri.org/dokumen/44_iin%20aminah_red.pdf [Diakses 26 November 2015]
- Ariani, Dian, Arna Fahriza dan Ira Prasetyaningrum. 2011. Optimasi Penjadwalan Mata Kuliah Di Jurusan Teknik Informatika PENS dengan Menggunakan Alogaritma *Particle Swarm Optimization* (PSO). Tersedia di: <http://repo.pens.ac.id/589/1/1235.pdf> [Diakses 15 September 2015]
- Badan Penelitian dan Pengembangan Petanian, Kementrian Pertanian. 2014. Kalender Tanam Terpadu Versi 2.0 Musim Hujan (MH) Oktober 2014 – Maret 2014.
- Bansal, J. C., dkk. 2011. *Inertia Weight Strategies in Particle Swarm Optimization*. Tersedia di: <http://ieeexplore.ieee.org/iel5/6083215/6089255/06089659.pdf> [Diakses 26 November 2015]
- Fadilah, Ayu Nur, Imam Cholissodin dan Wayan Firdaus Mahmudy. 2015. Implementasi *Analytical Hierarchy Process* (AHP) Dan Algoritma Genetika Untuk Rekomendasi Dan Optimasi Pemupukan Berimbang Tanaman Hortikultura. Tersedia di: filkom.ub.ac.id/doro/archives/detail/DR00139201506 [Diakses 15 September 2015]
- Putri, Marlina Perdana. 2011. Analisis Komparatif Usahatani Tumpangsari Jagung dan Kacang Tanah dengan Monokultur Jagung di Kabupaten Wonogiri. Tersedia di: <https://eprints.uns.ac.id/5992/> [Diakses 3 Oktober 2015]
- Indriati, Tri Retno, 2009. Pengaruh Dosis Pupuk Organik dan Populasi Tanaman Terhadap Pertumbuhan Serta Hasil Tumpangsari Kedelai (*Glycine Max L.*) dan Jagung (*Zea Mays L.*). Tersedia di: <http://eprints.uns.ac.id/8038/> [Diakses 3 Oktober 2015]
- Karimi, Maryam dan Homayoon Motameni. 2013. *Tasks Scheduling in Computational Grid using a Hybrid Discrete Particle Swarm Optimization*. Tersedia di: www.sersc.org/journals/IJGDC/vol6_no2/3.pdf [Diakses 18 November 2015]
- Krusienski, D. J. dan W. K. Jenkins. 2006. *A Particle Swarm Optimization – Least Mean Squares Algorithm for Adaptive Filtering*. Tersedia di: ieeexplore.ieee.org/iel5/9626/30419/01399128.pdf [Diakses 3 Oktober 2015]
- Soemarno. 2013. Pupuk dan Pemupukan Ramah Lingkungan. Malang: Fakultas Pertanian Universitas Brawijaya Malang.

Su, Te-Jen, Jui-Chuan Cheng dan Yang-De Sun. 2011. *Particle Swarm Optimization with Time-Varying Acceleration Coefficients Based on Cellular Neural Network for Color Image Noise Cancellation*. Tersedia di: https://www.thinkmind.org/download.php?articleid=icdt_2011_5_40_200_32 [Diakses 26 November 2015]

Sutedjo, Mul Mulyani. 1989. Pupuk dan Cara Pemupukan. Daerah Khusus Ibukota Jakarta: Rineka Cipta.

Wati, Dwi Ana Ratna dan Yuli Agusti Rochman. 2013. Model Penjadwalan Matakuliah Secara Otomatis Berbasis Algoritma *Particle Swarm Optimization* (PSO). Tersedia di: <http://journal.unpar.ac.id/index.php/jrsi/article/download/333/317> [Diakses 26 November 2015]



LAMPIRAN 1 SURAT IZIN PENELITIAN



PEMERINTAH KABUPATEN MALANG
BADAN KESATUAN BANGSA DAN POLITIK
Jalan KH. Agus Salim No. 7 Telp. (0341)366260 Fax. 366260
MALANG - 65119

SURAT KETERANGAN

Nomor : 072/657/421.205/2015

Untuk melakukan Survey / Research / Penelitian / KKN / PKL / Magang

Menunjuk : Ketua Program PTIIK UB.No;4744/UN10.36/36/AK/2015 tanggal;28 Des 2015
Perihal:Ijin

Dengan ini kami **TIDAK KEBERATAN** dilaksanakannya kegiatan **Permohonan Data** oleh :

Nama / Instansi : ROBBY YUDHA SAPUTRA/Mhs PTIIK UB Malang

Alamat : Jl. Veteran No;8 Malang

Thema/Judul/Survey/Research : Optimasi Nutrisi Pupuk Dalam Pertanian Tumpangsari
Dengan Algoritma Genetika

Daerah/tempat kegiatan : di BPTP Kec.Karang Ploso Kab.Malang,dan BALITKABI
Kec.Pakisaji Kab.Malang

Lamanya : 1 Bulan

Pengikut : Bayu Adi Prakoso dan Pasca Imanuddin Akbar

Dengan Ketentuan :

1. Mentaati ketentuan - ketentuan / Peraturan yang berlaku
2. Sesampainya ditempat supaya melapor kepada Pejabat setempat
3. Setelah selesai mengadakan kegiatan harap segera melapor kembali ke Bupati
Malang Cq. Kepala Badan Kesatuan Bangsa dan Politik Kabupaten Malang ;
4. Surat Keterangan ini tidak berlaku apabila tidak memenuhi ketentuan tersebut di atas

Malang 22 Desember 2015

A/n KEPALA BADAN KESATUAN BANGSA DAN POLITIK



Budiarto Hermawan SH.MSi

Pembina

NIP : 19671204 199303 1 007

TEMBUSAN :

Yth.

1. Sdr.Ketua Program PTIIK UB Malang
2. Sdr.Kepala BPTP Kec.Karang Ploso Kab.Malang
3. Sdr.Kepala BALITKABI Kec.Pakisaji kab.Malang
4. Sdr.Mhs/Ybs
5. Arsip

LAMPIRAN 2 HASIL WAWANCARA PAKAR

Pada lampiran ini, memaparkan hasil wawancara dengan peneliti tanaman jagung dan kacang kedelai dari Balai Pengkajian Teknologi Pertanian Propinsi Jawa Timur, Bapak Zainal Rabu, 6 Januari 2016.

1. *Range* pupuk yang ada pada kabupaten berasal dari mana?
 - Untuk *range* pupuk kabupaten diperoleh dari batas bawah dan atas yang ada di seluruh kecamatan yang ada pada kabupaten tersebut.
2. Untuk rekomendasi pupuk tanaman jagung dan kacang kedelai tersebut apakah bisa digunakan sebagai rekomendasi pupuk tumpangsari jagung dan kacang kedelai?
 - Rekomendasi pemupukan itu dapat digunakan untuk pemupukan tumpangsari jagung dan kacang kedelai.
3. Tumpangsari jenis apa yang cocok untuk jagung dan kacang kedelai?
 - Tumpangsari yang cocok untuk tanaman jagung dan kacang kedelai adalah *Strip Cropping*.
4. Apa yang harus diperhatikan dalam pertanian tumpangsari jagung dan kacang kedelai?
 - Jarak merupakan hal yang harus diperhatikan, agar tidak terjadi perebutan hara tanah dan cahaya matahari. Sehingga jarak yang ideal antar baris tanaman jagung adalah 70 cm x 40 cm, sedangkan jarak ideal antar baris tanaman kedelai adalah 15 cm x 40 cm, pada sela antar baris tanaman jagung terdapat 3 baris kacang kedelai.

LAMPIRAN 3 HASIL WAWANCARA PETANI

Pada lampiran ini, memaparkan hasil wawancara dengan salah satu petani jagung dan kacang kedelai pada daerah Kecamatan Sumbergempol Kabupaten Tulungagung, Bapak Supeno pada Rabu, 25 Mei 2016.

1. Dalam bertani jagung biasanya dikombinasikan dengan tanaman apa?
 - Biasanya jagung dikombinasikan dengan tanaman kacang-kacangan, bisa dengan kacang tanah atau dengan kacang kedelai, karena sama-sama tanaman yang tidak perlu banyak air.
2. Ukuran sawah dalam satu petak berapa?
 - Dalam satu petak itu ada 50 m x 50 m.
3. Pupuk yang digunakan apa saja?
 - Biasanya menggunakan Urea, ZA, PUSRI, SP-36, KCL, NPK, POSKA.
4. Takaran pupuk Urea, SP-36 dan KCL yang diberikan berapa?
 - Untuk tanaman jagung:
 - Urea diberikan 80 kg diawal penanaman dan 50 kg diakhir saat tongkol mulai muncul.
 - SP-36 total 50 kg diberikan diawal penanaman dan saat keluar bunga.
 - KCL diberikan 40 kg dicampur dengan urea disaat mulai keluar tongkol.
 - Untuk tanaman kacang kedelai:
 - Urea total 20 kg diberikan diawal penanaman dan pertengahan saat keluar bunga.
 - SP-36 total 30 kg diberikan diawal penanaman dan pertengahan.
 - KCL total 40 kg diberikan pada pertengahan dan saat mulai berbuah.

LAMPIRAN 4 HASIL PENGUJIAN

2.1 HASIL PENGUJIAN JUMLAH POPULASI

Ukuran Populasi	Nilai <i>Fitness</i>										Rata-rata <i>Fitness</i>
	Percobaan ke- <i>i</i>										
	1	2	3	4	5	6	7	8	9	10	
10	4,8891205	4,7778706	5,0046370	4,7275707	4,9102841	4,9728269	4,9496061	4,9054511	4,8626360	4,8424822	4,884249
20	4,7787748	4,9729592	4,8803624	5,0615008	4,8291206	4,8499375	4,9519995	5,0616170	5,0178667	5,0069274	4,941107
30	5,0477749	4,9731687	4,9380996	4,7594810	5,0152198	5,0436909	5,0420235	4,9302675	5,0313202	5,0233450	4,980439
40	4,9037123	4,9718075	4,9751606	4,9780616	5,0692631	5,0584432	4,9708682	4,8848010	5,0045028	4,9672824	4,978390
50	4,9928285	5,0430267	4,8124958	5,0612313	5,0323124	5,0295324	5,0893016	5,0019870	4,9611931	5,0396336	5,006354
60	4,9495664	5,0715169	4,9670526	5,0623297	4,9962392	5,0249859	4,9946468	5,0295276	5,0302861	5,0024315	5,012858
70	5,0911072	4,9167149	4,9546722	5,0444647	5,0718003	5,0290325	4,9806435	4,9923132	5,0537612	5,0629481	5,019746
80	5,0036750	5,0631055	4,9968085	5,0381898	5,0552096	5,0857055	4,9425154	5,0713228	5,0781540	5,0683818	5,040307
90	5,0386842	5,0781757	5,0416350	4,9971106	5,0031074	5,0568682	5,0662780	5,0095429	5,0662036	5,0206718	5,037828
100	5,0471878	5,0545701	5,0490090	5,0072909	5,0491277	5,0574160	5,0515838	5,0125496	5,0447538	5,0158663	5,038935

2.2 HASIL PENGUJIAN JUMLAH ITERASI

Jumlah Iterasi	Nilai <i>Fitness</i>										Rata-rata <i>Fitness</i>
	Percobaan ke- <i>i</i>										
	1	2	3	4	5	6	7	8	9	10	
10	5,0036750	5,0631055	4,9968085	5,0381898	5,0552096	5,0857055	4,9425154	5,0713228	5,0781540	5,0683818	5,040307
20	5,0979713	5,0909443	5,0891716	5,0958225	5,0972098	5,0703078	5,0493236	5,0933610	5,0946594	5,0777039	5,085648
30	5,0995523	5,0991298	5,0958892	5,0951991	5,1003442	5,1007207	5,1003510	5,1000237	5,0935641	5,0783689	5,096314
40	5,0996282	5,0980768	5,1012609	5,0991623	5,1018285	5,0947822	5,1001903	5,1013531	5,0896234	5,1019356	5,098784
50	5,1011297	5,1013916	5,1018066	5,1011090	5,1020198	5,1019923	5,0890552	5,1009841	5,1016635	5,1018541	5,100301
60	5,1012255	5,1020140	5,1007630	5,0996812	5,1020050	5,1003931	5,1019640	5,1014771	5,1009069	5,1017364	5,101217
70	5,1019272	5,1020385	5,0953627	5,1019562	5,1019562	5,1015080	5,1018238	5,1018879	5,1020275	5,1017568	5,101224
80	5,1019068	5,1019440	5,0765977	5,1019797	5,1019797	5,1019582	5,1019582	5,1029034	5,1019164	5,1019692	5,099511
90	5,1020380	5,1000378	5,0849009	5,1020355	5,1020355	5,1019997	5,0994273	5,1019997	5,1019997	5,1020356	5,099851
100	5,1020380	5,0914444	5,1020396	5,1014632	5,1020404	5,1020372	5,1020314	5,1020407	5,1019997	5,1020356	5,100917