

OPTIMASI KOMPOSISI MAKANAN PADA PENDERITA DIABETES MELITUS DAN KOMPLIKASINYA MENGGUNAKAN ALGORITMA GENETIKA

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Maryamah

NIM: 135150201111008



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2017**

PENGESAHAN

OPTIMASI KOMPOSISI MAKANAN PADA PENDERITA DIABETES MELITUS DAN
KOMPLIKASINYA MENGGUNAKAN ALGORITMA GENETIKA

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Maryamah

NIM: 135150201111008

Skripsi ini telah diuji dan dinyatakan lulus pada
20 April 2017

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Rekyan Regasari MP, S.T., M.T

NIK: 2011027704142001

Satrio Agung Wicaksono, S.Kom., M.Kom

NIP: 19860521 201212 1 001

Mengetahui

Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D

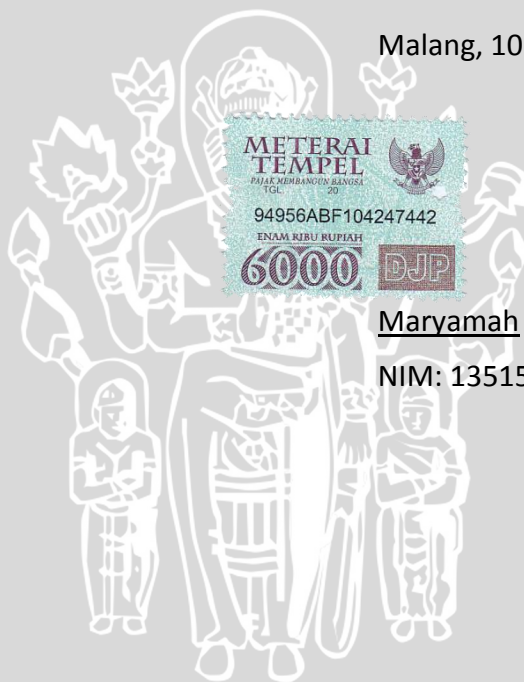
NIP. 19710518 200312 1 001

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, 10 April 2017



Maryamah

NIM: 135150201111008

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Optimasi Komposisi Makanan Pada Penderita Diabetes Melitus Dan Komplikasinya Menggunakan Algoritma Genetika” ini dapat terselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Ibu Rekyan Regasari MP, S.T., M.T dan Bapak Satrio Agung Wicaksono, S.Kom., M.Kom selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini serta dr.Asri Prameswari selaku pakar yang memberikan ilmu yang sangat membantu dalam menyelesaikan penelitian.
2. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku ketua Jurusan Informatika dan Bapak Agus Wahyu Widodo, S.T, M.Cs selaku ketua prodi Informatika.
3. Ayahanda dan Ibunda dan seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
4. Seluruh teman-teman seperjuangan Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 10 April 2017

Penulis

Maryamahfaisol02@gmail.com

ABSTRAK

Diabetes Melitus (DM) merupakan penyebab kematian no 4 terbesar didunia dan memiliki pertumbuhan terpesat serta menyebabkan kematian lebih banyak dibandingkan dengan penyakit lain sehingga penyakit diabetes diperlukan perhatian yang serius. Salah satu hal yang dapat dilakukan adalah dengan terapi diet yang dilakukan dengan menyusun menu makanan harian. Diet penderita diabetes dilakukan dengan memperhatikan kebutuhan energi, karbohidrat, protein, dan lemak dengan tepat dan akurat. Perhitungan energi pasien dapat dilakukan dengan manual atau dengan bantuan sistem yang menerapkan suatu algoritma. Jika dilakukan dengan proses manual akan membutuhkan waktu yang lama terutama apabila bahan makanan yang tersedia sangat banyak dan diet diabetes setiap komplikasi memiliki kebutuhan yang berbeda. Dengan jumlah bahan makanan yang banyak, cara manual akan mengalami kesulitan dan memiliki waktu relatif lama. Dengan bantuan sistem proses komputasi akan berlangsung dengan cepat dan jika diterapkan suatu algoritma perhitungan akan menghasilkan solusi yang lebih optimal, salah satu algoritma tersebut adalah algoritma genetika. Algoritma genetika merupakan metode heuristik yaitu suatu metode pencarian yang dalam pelaksanaannya terdapat aturan-aturan untuk memperoleh solusi yang lebih baik daripada solusi yang telah diperoleh sebelumnya serta dalam pelaksanaannya waktu yang digunakan untuk mencari solusi relatif cepat. Algoritma genetika banyak diterapkan pada berbagai macam permasalahan optimasi sehingga diharapkan juga dapat mengoptimalkan masalah pada optimasi komposisi makanan. Hasil yang didapatkan dari penelitian yang dilakukan adalah individu dalam populasi yang optimal berjumlah 250 individu dengan jumlah generasi 145 dan kombinasi cr dan mr paling optimal adalah 0.7 dan 0.3 dengan fitness 0.01857.

Kata kunci: Diabetes Melitus, Optimasi, Algoritma Genetika

ABSTRACT

Diabetes Mellitus (DM) is the 4th largest cause of death in the world and has the fastest growth and causes more deaths than any other disease so diabetes is needed serious attention. One of the things that can be done is with diet therapy that is done by preparing a daily diet. Diabetic diet is need attention in energy, carbohydrates, proteins, and fats with precise and accurate. The calculation of the patient's energy can be done manually or with the help of a system that implements an algorithm. If manual process will take a long time especially the available food is very much and every complication diabetic diet has different needs. With the amount of food that much, the manual way will have difficulty and have a relatively long time. With system computing process will take place quickly and if applied a calculation algorithm will yield more optimal solution, one of the algorithm is genetic algorithm. Genetic algorithm is a heuristic method that is a search method in the implementation there are rules to obtain a better solution than the solution that has been obtained previously and in the implementation of time used to find a solution relatively quickly. Genetic algorithm is widely applied to various optimization problems so it is also expected to optimize the problem on the optimization of food composition. The results obtained from the research conducted are individuals in the optimal population of 250 individuals with the number of generations of 145 and the combination of cr and mr most optimal is 0.7 and 0.3 with fitness 0.01857.

Keywords: *Diabetes Melitus, Optimization, Genetic Algorithm*



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang	1
1.2 Rumusan masalah.....	3
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Batasan masalah.....	3
1.6 Sistematika pembahasan.....	4
1.7 Jadwal Penelitian	5
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian Pustaka.....	6
2.2 Diabetes Melitus.....	6
2.2.1 Klasifikasi Diabetes Melitus.....	7
2.2.2 Komplikasi	8
2.3 Perhitungan Asupan Kalori	9
2.4 Diet Diabetes Melitus dan Komplikasinya	12
2.5 Algoritma Genetika.....	14
2.5.1 Kelebihan Algoritma Genetika	14
2.5.2 Proses Pencarian Solusi Algoritma Genetika	15
BAB 3 METODOLOGI	18

3.1 Studi Literatur	19
3.2 Pengumpulan Data	19
3.3 Perancangan	19
3.4 Implementasi	20
3.5 Pengujian	21
3.6 Kesimpulan	21
BAB 4 PERANCANGAN.....	22
4.1 Formulasi Permasalahan	22
4.2 Siklus Penyelesaian Masalah	24
4.2.1 Tahapan Algoritma Genetika	26
4.2.2 Desain Kromosom	36
4.2.3 Desain Fitnes	37
4.3 Manualisasi	40
4.3.3 Representasi Kromosom	41
4.3.4 Inisialisasi Populasi Awal	42
4.3.5 <i>Crossover</i>	42
4.3.6 Mutasi	43
4.3.7 Perhitungan Fitnes	43
4.3.8 Seleksi	44
4.3.9 Memilih Individu Terbaik	45
4.4 Perancangan Database	46
4.5 Perancangan UI	47
4.5.1 Data Input	47
4.5.2 Data Makanan	48
4.5.3 Algoritma Genetika	49
4.5.4 Individu Terbaik	50
4.5.5 Hasil	50
4.6 Perancangan Uji Coba dan Evaluasi	51
4.6.2 Pengujian Ukuran Individu Populasi	52
4.6.3 Pengujian Kombinasi Crossover Rate dan Mutation Rate	52
4.6.4 Pengujian Generasi	53
4.6.5 Pengujian Kromosom	54

BAB 5 IMPLEMENTASI	55
5.1 Struktur Kelas.....	55
5.2 Source Code	55
5.2.1 Representasi Kromosom	55
5.2.2 Crossover.....	56
5.2.3 Mutasi	58
5.2.4 Perhitungan Fitnes	59
5.2.5 Seleksi.....	61
5.3 Implementasi UI.....	64
5.3.1 Implementasi UI Beranda	64
5.3.2 Implementasi UI Data Makanan	65
5.3.3 Implementasi UI Algoritma Genetika.....	66
5.3.4 Implementasi UI Individu Terbaik	67
5.3.5 Implementasi UI Hasil	68
BAB 6 Pengujian	69
6.1 Pengujian Populasi.....	69
6.2 Pengujian Crossover Rate dan Mutation Rate	70
6.3 Pengujian Generasi.....	72
6.4 Pengujian Kromosom.....	74
6.5 Analisis Pengujian	76
BAB 7 Penutup	78
7.1 Kesimpulan	78
7.2 Saran	78
DAFTAR PUSTAKA.....	79
Lampiran	81

DAFTAR TABEL

Tabel 2.1 Kategori status gizi berdasarkan IMT	10
Tabel 2.2 Kategori status gizi berdasarkan Broca	10
Tabel 2.3 Kategori aktivitas	11
Tabel 2.4 Diet sesuai jumlah kalori harian	13
Tabel 4.1 Sumber Makanan	22
Tabel 4.2 Porsi bobot makanan	37
Tabel 4.3 Aturan Penalti.....	38
Tabel 4.4 Detail Kromosom.....	41
Tabel 4.5 Inisialisasi Populasi Awal	42
Tabel 4.6 Hasil Perhitungan Fitnes.....	43
Tabel 4.7 Hasil Seleksi	44
Tabel 4.8 Individu Terbaik.....	45
Tabel 4.9 Atribut jenis_makanan	46
Tabel 4.10 Atribut bahan_makanan	46
Tabel 4.11 Rancangan Pengujian Ukuran Populasi.....	52
Tabel 4.12 Rancangan Pengujian Crossover Rate (Cr) Dan Mutation Rate (Mr) ..	53
Tabel 4.13 Rancangan Pengujian Generasi.....	54
Tabel 6.1 Pengujian Populasi	69
Tabel 6.2 Pengujian Crossover rate dan Mutation rate.....	71
Tabel 6.3 Pengujian Generasi.....	72
Tabel 6.4 Hasil Menu Makanan.....	77

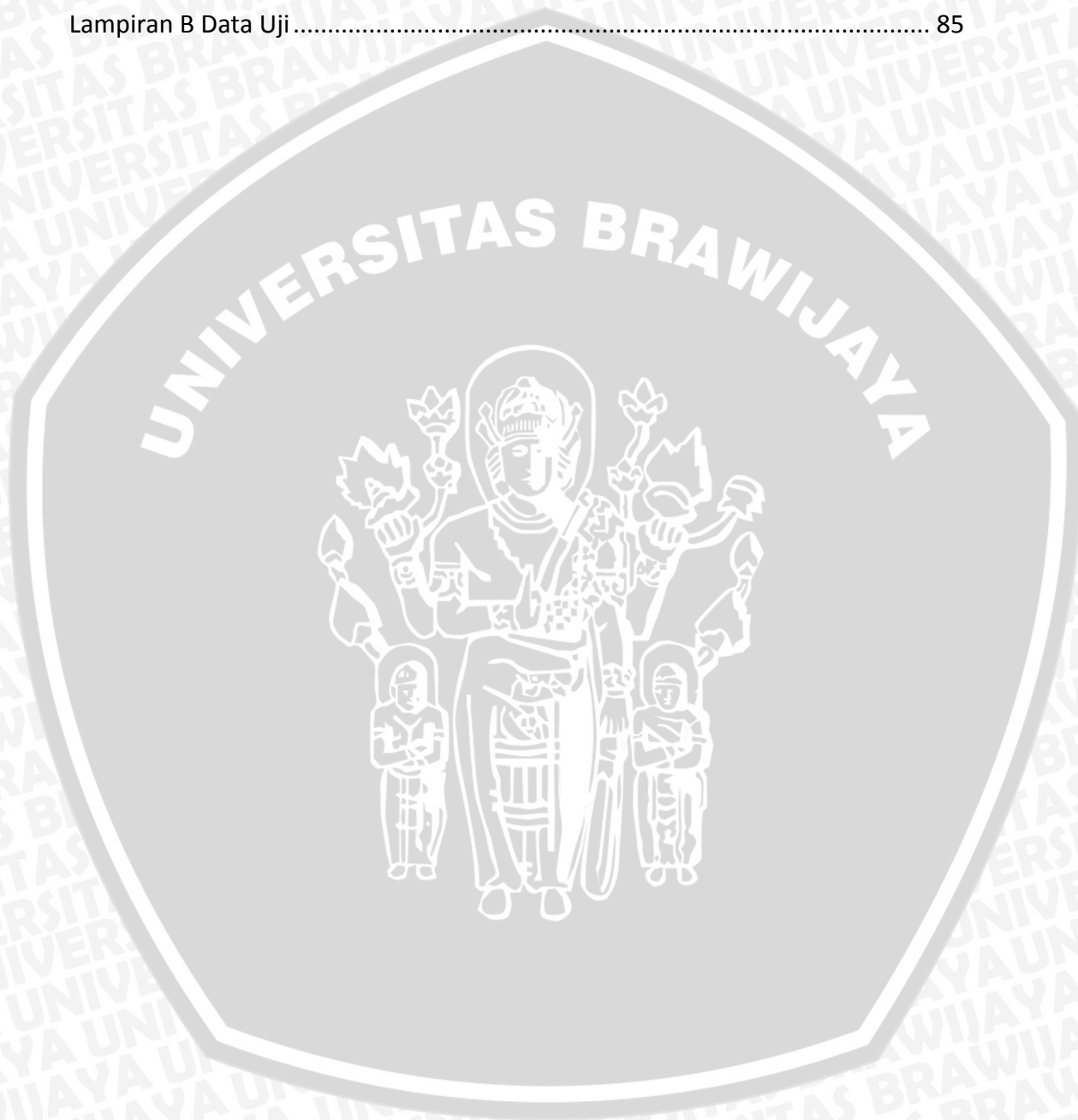
DAFTAR GAMBAR

Gambar 2.1 Representasi Kromosom	15
Gambar 2.2 Crossover One-cut Point	16
Gambar 2.3 Mutasi Resiprocal Exchange Mutation	16
Gambar 3.1 Langkah-langkah Penelitian	18
Gambar 3.2 Perancangan.....	20
Gambar 4.1 Flowchart Algoritma Genetika	25
Gambar 4.2 Flowchat inisialisasi populasi awal	26
Gambar 4.3 Flowchart Crossover.....	28
Gambar 4.4 Max Iterasi Crossover.....	29
Gambar 4.5 Flowchart Mutasi	31
Gambar 4.6 Perhitungan penalti.....	32
Gambar 4.7 Flowchart perhitungan fitness.....	33
Gambar 4.8 Flowchart Seleksi.....	35
Gambar 4.9 Random Injection	35
Gambar 4.10 CariGbest.....	36
Gambar 4.11 Representasi setiap bahan makanan	37
Gambar 4.12 Contoh Representasi Kromosom	37
Gambar 4.13 Contoh representasi kromosom individu	41
Gambar 4.14 Parent crossover 1	42
Gambar 4.15 Hasil child crossover 1	42
Gambar 4.16 Parent crossover 2	43
Gambar 4.17 Hasil child crossover 2	43
Gambar 4.18 Parent crossover 3	43
Gambar 4.19 Hasil child 3	43
Gambar 4.20 Parent mutasi 1	43
Gambar 4.21 Hasil child 1	43
Gambar 4.22 Parent mutasi 2	43
Gambar 4.23 Hasil child 2	43
Gambar 4.24 Perancangan Database.....	46

Gambar 4.25 Halaman Data Input	47
Gambar 4.26 Halaman Data Makanan	48
Gambar 4.27 Halaman Algoritma Genetika	49
Gambar 4.28 Halaman individu terbaik	50
Gambar 4.29 Halaman Hasil.....	51
Gambar 5.1 UI Beranda.....	64
Gambar 5.2 UI Data Makanan.....	65
Gambar 5.3 UI Algoritma Genetika	66
Gambar 5.4 UI Individu Terbaik	67
Gambar 5.5 UI Hasil.....	68
Gambar 6.1 Grafik Pengujian Populasi	70
Gambar 6.2 Grafik Pengujian kombinasi cr dan mr	71
Gambar 6.3 Pengujian Generasi	73
Gambar 6.4 Input Pengujian Kromosom.....	74
Gambar 6.5 Individu Terbaik Pengujian Kromosom	75
Gambar 6.6 Hasil Kromosom	75
Gambar 6.7 Hasil Pengujian	76

DAFTAR LAMPIRAN

Lampiran A Data Bahan Makanan	81
Lampiran B Data Uji	85



BAB 1 PENDAHULUAN

1.1 Latar belakang

Diabetes Melitus (DM) merupakan penyakit yang disebabkan oleh gangguan keseimbangan antara transportasi glukosa kedalam sel, yang disimpan dalam hati dan dikeluarkan oleh hati sehingga mengakibatkan kadar glukosa dalam darah meningkat (Tandra, 2013). Diabetes melitus adalah penyakit yang disebabkan oleh kelainan metabolis yang dapat disebabkan oleh beberapa faktor. Diabetes melitus memiliki gejala berupa hiperglikemia atau peningkatan kadar gula darah yang kronis. Hiperglikemia disebabkan oleh defisiensi sekresi hormon insulin atau defisiensi pengangkutan glukosa dalam tubuh. Selain itu gejala yang lainnya adalah memiliki gangguan metabolisme pada karbohidrat, lemak, dan protein (Susilo & Wulandari, 2011). Dengan adanya gangguan metabolisme, kondisi tubuh tidak dapat memproduksi insulin dengan benar atau tubuh mengalami kekurangan insulin. Sehingga glukosa dalam tubuh menumpuk (Sutanto, 2010). Selain itu diabetes melitus secara umum disebabkan oleh tiga faktor yaitu pola makan yang salah, gaya hidup yang modern dan obesitas (Susilo & Wulandari, 2011).

Penyakit diabetes melitus memiliki beberapa gejala yaitu sering buang air kecil, sering merasa haus, sering merasa lapar dan penurunan berat badan secara drastis (Susilo & Wulandari, 2011). Namun satu dari empat orang tidak menyadari bahwa mereka memiliki penyakit diabetes (Sutanto, 2010). Padahal diabetes merupakan penyebab kematian no 4 terbesar didunia dan merupakan penyakit yang memiliki pertumbuhan terpesat dan telah menyebabkan kematian lebih banyak dibandingkan dengan penyakit lain (Tandra, 2013). Pada tahun 2012 4,8 juta penduduk penderita diabetes melitus mengalami kematian dan setiap menit terdapat 6 orang yang meninggal diakibatkan penyakit diabetes (Tandra, 2013).

Jika penderita diabetes tidak mengontrol kadar gula dalam tubuhnya dengan teratur, tidak memperbaiki gaya hidup, diet yang tepat dan berolahraga dengan teratur dapat mengakibatkan penyakit diabetes mengalami komplikasi dengan penyakit mematikan. Adapun komplikasi tersebut adalah jantung, stroke, mata, ginjal, saraf, hipertensi, dan lambung (Prameswari, 2016). Gula darah yang terkontrol dengan baik, pengaturan diet yang benar dapat menghindarkan diabetes melitus dari berbagai komplikasi bahkan dapat mencegah dan mengobati penyakit diabetes (Tandra, 2013). Hal tersebut menjelaskan bahwa memperbaiki gaya hidup dan diet yang tepat dapat mengurangi komplikasi penyakit pada diabetes serta dapat meringankan kondisi penderita penyakit diabetes.

Hal yang perlu diperhatikan dalam diet yang tepat adalah menentukan komposisi makanan yang tepat yaitu bagaimana cara mengoptimalkan gizi dan nutrisi pada makanan yang dikonsumsi oleh penderita diabetes melitus. Cara menentukan komposisi makanan yang tepat dapat dilakukan manual (dilakukan oleh tenaga medis) dan menggunakan *software*. Perhitungan kalori dan kombinasi menu makanan yang dilakukan oleh tenaga medis memiliki banyak kelemahan

yaitu waktu penentuan membutuhkan waktu yang lama dan nutrisi yang terkandung belum tentu benar-benar optimal. Sedangkan dengan *software* penentuan komposisi makanan yang tidak menggunakan metode pendukung tidak membutuhkan waktu yang lama namun nutrisi yang dihasilkan juga belum tentu optimal. Sehingga diperlukan metode pendukung yang diimplementasikan pada *software* dan menghasilkan komposisi makanan yang cepat, nutrisi yang optimal dan harga makanan yang lebih murah.

Pada penelitian sebelumnya mengenai optimasi komposisi makanan yang dilakukan pada penderita penyakit diabetes melitus menggunakan algoritma genetika mendapatkan hasil kombinasi makanan yang dibutuhkan oleh penderita diabetes melitus dan biaya yang minimum. *Output* yang dihasilkan berupa list makan pagi, siang, dan malam (Rianawati & Mahmudy, 2015). Namun pada kenyataannya sangat jarang pasien yang menderita penyakit diabetes melitus saja tanpa komplikasi penyakit yang lain. Hal ini dikarenakan penyakit diabetes melitus menjadi sangat berbahaya karena penyakit tersebut menyebabkan komplikasi dengan penyakit lain (Prameswari, 2016). Sehingga sangat penting dilakukan penelitian mengenai optimasi komposisi makanan bagi penderita diabetes melitus dan komplikasinya.

Algoritma genetika banyak diterapkan pada berbagai macam permasalahan optimasi seperti pada penelitian sebelumnya mengenai optimasi bahan makanan pada penderita diabetes melitus (Rianawati & Mahmudy, 2015), optimasi lahan pertanian (Saputro & Mahmudy, 2015), optimasi barang dalam produksi (Ramuna & Mahmudy, 2015), optimasi penjadwalan (Sari & Mahmudy, 2015). Penelitian-penelitian tersebut dapat menghasilkan solusi yang mendekati optimum karena algoritma genetika merupakan metode heuristik yaitu suatu metode pencarian yang dalam pelaksanaannya terdapat aturan-aturan untuk memperoleh solusi yang lebih baik daripada solusi yang telah diperoleh sebelumnya. Karena sifat metode heuristik mendapatkan solusi yang lebih baik dari sebelumnya maka metode heuristik tidak selalu menghasilkan solusi optimum namun jika dibangun dengan baik kemungkinan solusi yang dihasilkan mendekati optimum. Serta dalam pelaksanaannya waktu yang digunakan untuk mencari solusi relatif cepat (Mahmudy, 2015). Dengan banyaknya penelitian yang menggunakan algoritma genetika pada berbagai permasalahan optimasi, algoritma genetika juga digunakan dalam penyelesaian permasalahan optimasi bahan makanan pada penderita penyakit diabetes melitus dan komplikasinya serta diharapkan dapat menghasilkan hasil yang optimum juga. Hasil dari penelitian dapat dikatakan optimum jika menu makanan yang dihasilkan tidak melebihi batasan kalori harian dari pasien penderita diabetes melitus.

Penelitian ini memiliki manfaat terutama bagi pasien penderita diabetes melitus yaitu pasien tidak kesulitan dalam menentukan makanan yang memenuhi gizi dengan harga yang terjangkau. Serta para penderita diabetes melitus harus hati-hati terhadap makanan yang dikonsumsi agar tidak memperparah kondisi dan terjangkit komplikasi yang lain. Berdasarkan uraian diatas, maka perlu dilakukan penelitian yang berjudul "Optimasi Komposisi Makanan pada Penderita Diabetes Melitus dan Komplikasinya menggunakan Algoritma Genetika".

1.2 Rumusan masalah

1. Apakah representasi kromosom yang dirancang sesuai dengan permasalahan optimasi komposisi makanan pada penderita penyakit diabetes dan komplikasinya?
2. Apakah komposisi makanan yang dihasilkan sudah optimal berdasarkan nilai fitnessnya?
3. Berapa nilai parameter dari jumlah individu dalam populasi, generasi, kombinasi *crossover rate* dan *mutation rate* yang optimal dari permasalahan optimasi komposisi makanan pada penderita penyakit diabetes dan komplikasinya?

1.3 Tujuan

1. Menentukan representasi kromosom yang dirancang sesuai dalam permasalahan optimasi komposisi makanan pada penderita penyakit diabetes dan komplikasinya.
2. Menentukan komposisi makanan yang dihasilkan optimal sesuai nilai fitness yang diperoleh.
3. Menerapkan metode algoritma genetika sehingga mendapatkan jumlah individu dalam populasi, generasi, kombinasi *crossover rate* dan *mutation rate* yang optimal dalam permasalahan optimasi komposisi makanan pada penderita penyakit diabetes dan komplikasinya.

1.4 Manfaat

1. Membantu para penderita penyakit diabetes melitus yang mengalami komplikasi dalam menentukan makanan yang optimal terutama pada penderita yang mengalami kebingungan dalam memilih makanan yang bernutrisi dan gizi tinggi.
2. Makanan memiliki biaya yang lebih terjangkau namun memiliki nutrisi dan gizi yang tinggi dan seimbang sesuai kebutuhan sehingga diharapkan penderita tidak kesulitan dalam mendapatkannya.
3. Tidak memperparah kondisi penyakit yang dihadapi karena makanan yang dihasilkan sesuai dengan makanan yang baik sesuai dengan penyakit yang dimiliki.
4. Bagi tenaga medis memudahkan dalam perumusan menu makanan yang optimal bagi pasien yang menderita penyakit diabetes melitus dan komplikasinya.

1.5 Batasan masalah

1. Membahas permasalahan pada komplikasi pasien penderita diabetes melitus tipe 2.
2. Komplikasi penyakit yang diteliti hanya komplikasi dengan penyakit jantung, stroke, ginjal, mata, lambung, dan saraf.

3. Harga dari makanan berasal dari daerah Malang pada bulan maret 2017.
4. Daftar menu yang dihasilkan digunakan pasien dalam waktu satu hari.

1.6 Sistematika pembahasan

Sistematika pembahasan dalam skripsi ini sebagai berikut :

Bab I Pendahuluan

Pada bab pendahuluan ini berisi mengenai latar belakang permasalahan yang dibahas, rumusan masalah, tujuan penelitian, manfaat dari penelitian yang akan di teliti terhadap orang lain, hipotesa dan batasan permasalahan dalam penelitian. Selain itu terdapat sistematika penulisan menjelaskan tentang struktur keseluruhan dari skripsi dan jadwal penelitian yang menampilkan susunan waktu dalam pengerjaan penelitian.

Bab II Landasan Kepustakaan

Pada bab ini akan dibahas mengenai teori-teori pendukung yang digunakan untuk membantu pada proses penelitian sehingga dapat menjadi referensi dalam melakukan penelitian.

Bab III Metodologi Penelitian

Pada bab ini akan membahas langkah-langkah yang digunakan dalam menyelesaikan penelitian. Metodologi penelitian terdiri dari metode pengambilan data, pengolahan data, proses perancangan sistem untuk menyelesaikan permasalahan, proses implelementasi yang akan dilakukan serta pengujian sistem.

Bab IV Perancangan

Berisi formulasi permasalahan yang akan diselesaikan serta data yang telah diperoleh untuk keperluan optimasi pemenuhan gizi dan nutrisi pada penderita penyakit diabetes melitus. Terdapat simulasi penyelesaian masalah sebagai Gambaran bagaimana tahapan-tahapan algoritma genetika dalam menyelesaikan permasalahan. Pada bab ini juga terdapat perancangan database, perancangan UI untuk merancang interface sistem yang akan dibuat serta perancangan uji coba dan evaluasi yang akan digunakan untuk mengetahui parameter algoritma genetika yang paling optimal sehingga menghasilkan solusi yang optimum.

Bab V Implementasi

Membahas mengenai pengimplementasian masalah menjadi sistem, menjelaskan mengenai cara implementasi algoritma genetika ke dalam program. Tahapan-tahapan yang dilakukan sama dengan tahapan dalam bab perancangan bagian simulasi penyelesaian masalah serta UI yang akan diimplementasikan juga sesuai dengan pada bab perancangan.

Bab VI Pengujian dan Analisis
Memuat tentang proses pengujian validasi sistem dan analisis hasil pengujian berupa akurasi sistem yang telah direalisasikan serta membahas detail sistem yang dibuat.

Bab VII Penutup
Memuat kesimpulan dari perancangan, implementasi, dan pengujian sistem yang telah dilakukan serta memberi saran untuk pengembangan penelitian yang lebih lanjut.

1.7 Jadwal Penelitian

Jadwal pelaksanaan skripsi sebagai berikut:

No	PROSES KEGIATAN :	BULAN DAN MINGGU KE:															
		Bulan X				Bulan XI				Bulan XII				Bulan I			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Studi Literatur																
2	Analisis Kebutuhan																
3	Pengumpulan Data																
4	Perancangan																
5	Implementasi																
6	Pengujian																

BAB 2 LANDASAN KEPUSTAKAAN

Terdapat beberapa uraian dan pembahasan yang dibutuhkan dalam penyelesaian skripsi ini. Teori – teori tersebut mengenai penjelasan diabetes melitus dan komplikasinya, penyakit komplikasi yang diderita oleh penderita diabetes melitus, serta penjelasan algoritma genetika sebagai metode yang digunakan dalam mengoptimalkan komposisi pada penderita penyakit diabetes melitus dan komplikasinya.

2.1 Kajian Pustaka

Terdapat beberapa penelitian sebelumnya yang menggunakan algoritma genetika seperti pada permasalahan optimasi biaya makanan dan gizi manula (Suci & Mahmudy, 2015). Penelitian tersebut lebih menekankan pada biaya dan hasil yang didapatkan dapat memberikan solusi biaya yang minimum pada total menu seimbang selama satu hari. Tidak terdapat penjelasan mengenai akurasi dari penelitian yang dilakukan. Namun, hasil yang didapatkan dari 500 generasi memiliki rata-rata fitness 110175.49, ukuran populasi memiliki rata-rata fitness 1198113.462 serta kombinasi antara *crossover* dan *mutation rate* mendapatkan hasil 0.3 dan 0.7 dengan rata-rata fitness 109795.06. Pada perolehan fitness 110175.49 generasi penelitian mengalami konvergensi sehingga proses iterasi dihentikan dan dapat disimpulkan bahwa hasil tersebut merupakan hasil yang mendekati solusi optimal.

Selanjutnya penelitian pada permasalahan optimasi bahan makanan penderita diabetes melitus menggunakan algoritma genetika (Rianawati & Mahmudy, 2015). Penelitian tersebut membahas bahan makanan pada penderita diabetes pada umumnya padahal keadaan nyata penderita diabetes selalu diikuti dengan penyakit komplikasi lain (Prameswari, 2016). Pada penelitian yang lain mengenai optimasi bahan makanan penderita kolesterol menggunakan algoritma genetika (Wahid & Mahmudy, 2015). Hasil yang diperoleh tidak berbeda jauh dengan penelitian yang telah dibahas sebelumnya hanya objek yang diteliti berbeda.

2.2 Diabetes Melitus

Diabetes melitus (DM) merupakan penyakit yang disebabkan oleh gangguan keseimbangan antara transportasi glukosa ke dalam sel, yang disimpan dalam hati dan dikeluarkan oleh hati sehingga mengakibatkan kadar glukosa dalam darah meningkat (Tandra, 2013). Diabetes melitus merupakan penyakit yang dapat dimiliki orang dalam jangka waktu yang panjang (penyakit menahun) karena tidak mudah untuk menurunkan kadar gula dalam darah sehingga membutuhkan penanganan yang tepat (Ernawati, 2013). Diabetes adalah kondisi tubuh tidak dapat memproduksi insulin dengan benar atau tubuh mengalami kekurangan insulin sehingga glukosa dalam darah menumpuk (Sutanto, 2010).

Menurut Tandra (2013) terdapat beberapa faktor yang menyebabkan seseorang mudah terkena diabetes diantaranya adalah

1. Keturunan. Jika terdapat sejarah keluarga yang pernah memiliki diabetes maka orang tersebut juga beresiko memiliki penyakit diabetes.
2. Ras tau etnis. Orang asia beresiko lebih tinggi terkena diabetes dan orang berkulit hitam beresiko dibandingkan orang berkulit putih.
3. Usia. Seiring bertambahnya usia terutama diatas 40 tahun resiko diabetes akan meningkat.
4. Obesitas. Timbunan lemak di perut akan menyebabkan insulit sulit berkerja dan kadar gula dalam darah akan meningkat.
5. Kurang gerak badan.
6. Kehamilan. 2-5 persen wanita hamil dapat menderita diabetes namun akan sembuh ketika wanita tersebut melahirkan.
7. Infeksi. Infeksi yang terjadi pada pankreas dan merusaknya dapat membuat pancreas tidak menghasilkan hormon insulin seperti biasanya sehingga menyebabkan gula dalam darah meningkat.
8. Stress. Hormon counter-insulin atau hormon yang bekerja berlawanan dengan insulin dihasilkan ketika seseorang mengalami stress. Jika hormon tersebut aktif dapat menghambat kerja insulin.
9. Obat-obatan. Beberapa obat dapat menyebabkan kadar gula dalam darah contohnya adalah obat antihipertensi dan obat penurun kolesterol.

2.2.1 Klasifikasi Diabetes Melitus

Menurut Susilo & Wulandari (2011) diabetes melitus dibagi menjadi tiga tipe yaitu

2.2.1.1 Diabetes Melitus Tipe 1

Diabetes melitus tipe satu merupakan diabetes yang diderita oleh anak-anak dan remaja. Diabetes ini terjadi karena rasio insulin dalam sirkulasi darah berkurang sehingga mengakibatkan sel penghasil insulin dalam pankreas hilang. Gejala yang ditimbulkan biasanya mendadak, dan apabila kondisi pasien parah dapat menyebabkan koma sehingga harus sesegera mungkin dibawa kedokter. Diet dan olahraga tidak dapat menyembuhkan diabetes tipe 1. Cara yang dilakukan untuk mengobatinya adalah dengan menyuntikkan insulin dan pada tahap awal dilakukan pergantian insulin dalam tubuh.

2.2.1.2 Diabetes Melitus Tipe 2

Diabetes melitus tipe 2 diderita oleh orang dewasa dan manula. Diabetes melitus tipe dua terjadi bukan rasio insulin dalam sirkulasi darah tetapi kelainan metabolisme yang disebabkan oleh beberapa faktor. Faktor tersebut adalah mutasi pada banyak gen, gangguan pengeluaran hormon insulin, disfungsi sel

jaringan yang menyebabkan resistansi sel pada insulin. 90% pasien diabetes melitus didunia disebabkan oleh obesitas dan faktor lain merupakan sejarah diabetes melitus pada keluarga. Diet dan olahraga dapat mengobati diabetes tipe 2 sehingga sangat penting dilakukan bagi pasien penderita diabetes melitus.

2.2.1.3 Diabetes Melitus Tipe 3

Diabetes melitus tipe 3 merupakan diabetes yang terjadi ketika hamil atau disebut juga diabetes gestasional. Karena diabetes tipe 3 terjadi ketika hamil maka setelah melahirkan dapat sembuh. Wanita hamil yang menderita diabetes ini terdapat kemungkinan merusak janin dan 20-50% wanita saja yang dapat bertahan hidup. Pengobatan yang dilakukan adalah menambah hormon insulin dan pemeriksaan yang rutin untuk mencegah terjadinya kerusakan pada janin (Tandra, 2013).

2.2.2 Komplikasi

Menurut Prameswari (2016) komplikasi-komplikasi yang dialami oleh penderita diabetes melitus dibagi menjadi dua yaitu

2.2.2.1 Mikrovaskular

Komplikasi *mikrovaskular* merupakan komplikasi dari penyakit diabetes melitus yang menyerang pembuluh darah arteri yang berukuran kecil. Komplikasi *mikrovaskular* terdiri dari empat jenis yaitu

1. Diabetes Retinopati. Diabetes retinopati adalah penyakit kerusakan yang terjadi pada retina. Penyebab terjadinya retinopati adalah menumpuknya glukosa pada pembuluh darah kapiler retina yang berukuran sangat kecil. Penyakit retinopati dibagi menjadi dua macam. Retinopati nonproliferatif dan retinopati proliferatif. Retinopati nonproliferatif disebabkan terjadinya pembengkakan pada retina bahkan terdapat sedikit kebocoran cairan serum dari pembuluh darah retina. Retinopati proliferatif disebabkan pendarahan pada pembuluh darah retina dan terbentuk pembuluh darah baru yang rapuh sehingga mudah berdarah. Selain retinopati terdapat dua komplikasi mata lain yang disebabkan oleh diabetes yaitu katarak dan glaukoma.
2. Diabetes Neuropati. Diabetes neuropati merupakan kerusakan saraf pada penderita diabetes biasanya menyerang daerah kaki pasien. Gangguan saraf yang terjadi adalah kaki tidak dapat merasakan panas, nyeri dan kesemutan sehingga pasien tidak menyadari jika terjadi luka. Luka yang diderita penderita diabetes sangat lama untuk disembuhkan dan beresiko terkena infeksi dan menyebabkan kaki harus diamputasi.
3. Diabetes Nefropati. Diabetes nefropati merupakan kerusakan yang terjadi pada ginjal. Kerusakan pada ginjal disebabkan oleh pengobatan yang dilakukan oleh pasien penderita diabetes. Dalam penyembuhan diabetes dokter hanya terfokus menurunkan kadar gula darah sehingga ginjal menjadi efek samping yang didapatkan. Padahal ginjal tidak dapat dianggap remeh karena ginjal adalah organ yang bertugas mengeluarkan racun dalam tubuh. Jika ginjal terjadi kerusakan, racun dalam tubuh tidak dapat dikeluarkan dan protein

sebagai bahan yang disaring ginjal untuk dimanfaatkan tubuh menjadi bocor keluar. Rata-rata 1 dari 10 orang penderita diabetes tipe 2 menderita nefropati (Tandra, 2013).

4. Diabetes Gastropati. Gastropati atau kerusakan lambung disebabkan oleh obat-obatan untuk mengobati diabetes yang diderita.

2.2.2.2 Makrovaskular

Komplikasi *makrovaskular* merupakan kebalikan dari *mikrovaskular* yaitu komplikasi dari penyakit diabetes melitus yang mengenai pembuluh darah arteri berukuran besar. Komplikasi *makrovaskuler* menyerang pembuluh darah yang berukuran besar dalam tubuh sesuai dengan letak pembuluh darah yang dapat menimbulkan masalah yang sangat serius (Romahorbo, 2014). Adapun macam-macam komplikasi *makrovaskular* adalah

1. Diabetes Kardiomiopati. Komplikasi diabetes yang paling banyak dalam masyarakat adalah komplikasi penyakit jantung. Hal ini dikarenakan glukosa menumpuk pada darah dan jantung sebagai organ yang mengatur peredaran darah tubuh bekerja lebih berat dan jika hal ini terus berlangsung dapat menyebabkan penyakit jantung. Penyakit jantung terdiri dari beberapa jenis diantaranya penyakit jantung koroner, serangan jantung dan gagal jantung. Penyakit jantung koroner merupakan gangguan yang terjadi pada aliran darah di pembuluh darah koroner sehingga jantung tidak berdetak dengan baik, melemah dan terjadi gagal jantung.
2. Stroke. Kemungkinan penderita diabetes 2-3 kali lipat lebih besar terkena stroke dibandingkan dengan orang yang tidak memiliki diabetes (Tandra, 2013). Hal ini dikarenakan gejala yang serupa dengan yang terjadi pada penyakit jantung. Pembuluh darah penderita diabetes menyempit sehingga saluran darah tidak lancar bahkan tersumbat. Tidak menutup kemungkinan aliran darah pada otak terhambat dan menyebabkan terjadinya stroke.

2.3 Perhitungan Asupan Kalori

Terdapat beberapa cara yang dilakukan dalam menentukan status gizi yang nantinya akan digunakan dalam menentukan asupan kalori yang dibutuhkan dalam diet penderita diabetes. Pertama langkah yang dilakukan adalah menghitung status gizi. Penentuan status gizi dapat dihitung berdasarkan perhitungan IMT dan perhitungan Broca (Ernawati, 2013).

Perhitungan status gizi berdasarkan IMT (Indeks Masa Tubuh) dapat dilihat pada persamaan 2.1

$$IMT = \frac{\text{Berat Badan}}{(\text{Tinggi Badan})^2} \quad (2.1)$$

Pada persamaan 2.1 berat badan menggunakan satuan kilogram dan tinggi badan menggunakan satuan meter. Hasil dari perhitungan yang didapatkan dapat dilihat pada kategori status gizi seperti tabel 2.1

Tabel 2.1 Kategori status gizi berdasarkan IMT

Kategori status gizi	Indeks Masa Tubuh
BB kurus	< 18.5
BB normal	18.5 – 22.9
BB lebih	> 23.0
Dengan resiko	23 - 24.9
Obesitas I	25 – 29.9
Obesitas II	≥ 30

Menurut rumus broca dengan menghitung BBI (Berat Badan Ideal) dapat menggunakan persamaan 2.2 dan persamaan 2.3

$$BBI = (Tinggi\ badan\ dalam\ cm - 100) - 10\% \quad (2.2)$$

$$BBI = 90\% * (Tinggi\ badan\ dalam\ cm - 100) * 1kg \quad (2.3)$$

Bagi laki-laki yang memiliki tinggi badan ≤ 160 cm dan perempuan ≤ 150 cm maka rumus yang digunakan adalah persamaan 2.4

$$BBI = (Tinggi\ badan\ dalam\ cm - 100) * 1kg \quad (2.4)$$

Hasil yang didapatkan dikelompokkan dalam beberapa kategori status gizi seperti pada tabel 2.1

Tabel 2.2 Kategori status gizi berdasarkan Broca

Kategori gizi	Berat badan ideal
BB Kurus	BB < 90% BBI
BB Normal	BB 90-110% BBI
BB Lebih	BB 110-120% BBI
Gemuk	BB > 120% BBI

Setelah mendapatkan nilai BBI, selanjutnya dilakukan perhitungan kalori harian yang dibutuhkan. Terdapat beberapa informasi yang dibutuhkan yaitu jenis kelamin, berat badan, umur dan aktivitas fisik yang dilakukan (Ernawati, 2013).

1. Jenis Kelamin

Perhitungan kalori sesuai dengan jenis kelamin disebut dengan kebutuhan basal atau AMB (Angka Metabolisme Basal). Menurut perhitungan rumus PERKENI atau Perkumpulan Endokrinologi Indonesia (Wahyuningsih, 2013). Rumus perhitungan sesuai dengan jenis kelamin terdapat pada persamaan 2.5 dan persamaan 2.6

$$AMB\ laki - laki = BBI\ (kg) * 30\ kalori \quad (2.5)$$

$$AMB\ perempuan = BBI\ (kg) * 25\ kalori \quad (2.6)$$

2. Umur

Perhitungan umur yang dilakukan menurut perhitungan rumus PERKENI atau Perkumpulan Endokrinologi Indonesi (Wahyuningsih, 2013) terdapat pada persamaan 2.7, 2.8 dan 2.9 sesuai dengan kategori umur.

$$\text{Umur } 40 - 59 \text{ tahun} = 5\% * \text{AMB Jenis Kelamin} \quad (2.7)$$

$$\text{Umur } 60 - 69 \text{ tahun} = 10\% * \text{AMB Jenis Kelamin} \quad (2.8)$$

$$\text{Umur } \geq 70 \text{ tahun} = 20\% * \text{AMB Jenis Kelamin} \quad (2.9)$$

3. Aktivitas Fisik

Aktivitas fisik dihitung sesuai dengan pekerjaan dan aktivitas yang dilakukan. Perhitungan menurut rumus PERKENI atau Perkumpulan Endokrinologi Indonesi (Wahyuningsih, 2013) terdapat pada persamaan 2.10 – 2.13 sesuai dengan kategori aktifitas fisik yang dijalani

$$\text{Keadaan istirahat} = 10\% * \text{AMB Jenis Kelamin} \quad (2.10)$$

$$\text{Keadaan ringan} = 20\% * \text{AMB Jenis Kelamin} \quad (2.11)$$

$$\text{Keadaan sedang} = 30\% * \text{AMB Jenis Kelamin} \quad (2.12)$$

$$\text{Keadaan berat dan sangat berat} = 50\% * \text{AMB Jenis Kelamin} \quad (2.13)$$

Pengelompokan kategori masing-masing aktivitas dapat dilihat pada tabel 2.3

Tabel 2.3 Kategori aktivitas

Aktifitas	Pekerjaan
Ringan	Pegawai kantor, pegawai toko, guru, ibu rumah tangga, sekretaris
Sedang	Pegawai industri ringan, mahasiswa
Berat	Pelaut, buruh, penari, atlit
Sangat Berat	Tukang becak, pandai besi, tukang gali

4. Berat badan

Perhitungan berat badan sesuai dengan kategori dapat dilihat pada persamaan 2.14 – 2.16

$$\text{BB gemuk} = -20\% * \text{AMB Jenis Kelamin} \quad (2.14)$$

$$\text{BB lebih} = 10\% * \text{AMB Jenis Kelamin} \quad (2.15)$$

$$\text{BB kurus} = 20\% * \text{AMB Jenis Kelamin} \quad (2.16)$$

Langkah selanjutnya adalah menghitung total kalori sesuai dengan persamaan 2.17

$$\text{Total kalori} = \text{AMB Jenis Kelamin} - \text{kalori umur} + \text{kalori aktivitas} + \text{kalori berat badan} \quad (2.17)$$

2.4 Diet Diabetes Melitus dan Komplikasinya

Pasien penderita penyakit diabetes melitus sangat disarankan oleh dokter untuk melakukan terapi diet. Karena kegunaan terapi diet pada penderita diabetes melitus sangat besar yakni memulihkan dan mempertahankan kadar gula yang terkandung dalam darah kembali menjadi normal sehingga tidak memperparah penyakit diabetes melitus yang diderita atau kemungkinan terburuk mengidap komplikasi. Kegunaan terapi yang lain adalah memberikan masukan jenis nutrisi yang cukup memadai untuk memperbaiki jaringan dalam tubuh serta memulihkan berat badan menjadi normal (Beck, 2011). Menurut Wahyuningsih (2013) terdapat beberapa hal yang perlu diperhatikan dalam pelaksanaan terapi diet bagi penderita diabetes yaitu

1. Kebutuhan protein adalah 10-15 % dari kebutuhan energi total
2. Kebutuhan lemak adalah 20-25 % dari kebutuhan energi total.
3. Kebutuhan karbohidrat 60-70 dari sisa kebutuhan energi total
4. Penggunaan gula murni tidak diperbolehkan. Jika kadar gula darah mulai normal boleh mengkonsumsi gula murni sampai 5 % dari kebutuhan energi total
5. Serat 25 gr/hari

Sesuai dengan beratnya penyakit diabetes, tipe pengobatan, kepribadian pasien, umur, berat badan dan gaya hidup pasien, terdapat tiga tipe diet bagi pasien penderita diabetes melitus menurut Beck (2011) yaitu

1. Diet rendah kalori

Diet rendah kalori digunakan dengan acuan berat badan dari pasien. Jika pasien mengalami obesitas maka sangat dianjurkan untuk melakukan diet rendah kalori. Pasien yang mengalami obesitas selain memiliki diabetes, rentan terserang penyakit lain sehingga perlu menurunkan berat badannya. Jika pasien mempunyai berat badan normal maka sedikit disarankan untuk melakukan diet rendah untuk menjaga berat badan pasien tetap normal.

2. Diet bebas gula

Diet bebas gula digunakan pada penderita pasien yang berusia lanjut usia dan tidak memerlukan suntikan insulin. Terdapat dua prinsip yang harus diterapkan dalam melakukan diet bebas gula. Prinsip tersebut adalah tidak memakan gula atau makanan yang mengandung gula dan mengkonsumsi makanan sumber hidraarang dalam sehari-hari.

3. Sistem penukaran hidraarang

Diet penukaran hidraarang digunakan pada pasien penderita diabetes melitus yang mendapat suntikan insulin atau obat-obat hipoglikemik oral yang memiliki dosis yang tinggi. Dalam pelaksanaan diet 55 % dari total energi harian harus ditukar dengan bentuk hidraarang.

Menurut Tjokroprawiro (2012) sesuai dengan jumlah kalori harian pasien. Diet pasien diabetes melitus dibagi menjadi beberapa tipe sesuai dengan tabel

Tabel 2.4 Diet sesuai jumlah kalori harian

Tipe	Besar Kalori
Diabetes Melitus I	1100
Diabetes Melitus II	1300
Diabetes Melitus III	1500
Diabetes Melitus IV	1700
Diabetes Melitus V	1900
Diabetes Melitus VI	2100
Diabetes Melitus VII	2300
Diabetes Melitus VIII	2500
Diabetes Melitus IX	2700
Diabetes Melitus X	2900
Diabetes Melitus XI	3100
Diabetes Melitus XII	3300

Diet sesuai waktu makan menurut Tjokroprawiro (2012) diberikan makanan utama dan tiga kali makanan pelengkap dengan selisih tiga jam. Contoh:

1. Pukul 06.30 makan pagi
2. Pukul 09.30 makan kecil atau buah
3. Pukul 12.30 makan siang
4. Pukul 15.30 makan kecil atau buah
5. Pukul 18.30 makan malam
6. Pukul 21.30 makan kecil atau buah

Menurut Prameswari (2016) diet diabetes dan komplikasinya dibagi menjadi dua yaitu

1. Komplikasi dengan penyakit jantung, ginjal dan stroke

Penyakit diabetes melitus yang berkomplikasi dengan penyakit jantung, ginjal atau stroke adalah komplikasi yang harus benar-benar melakukan diet dan mengontrol makanan yang masuk dalam tubuh. Hal ini dikarenakan komplikasi penyakit yang diderita sangat berbahaya dan dapat berujung pada kematian. Diet yang dilakukan adalah kandungan lemak yang masuk dalam tubuh $\pm 30\%$ dan $\pm 12\%$ kandungan protein dari total kalori harian serta sisanya untuk karbohidrat ($\pm 58\%$).

2. Komplikasi dengan penyakit mata, saraf dan lambung

Pada komplikasi diabetes melitus dengan penyakit mata, saraf, dan lambung, diet yang dilakukan hampir sama dengan diet pada penderita diabetes melitus yang tidak berkomplikasi. Komplikasi dengan beberapa penyakit ini dianggap tidak terlalu berbahaya namun harus tetap diperhatikan karena jika tidak melakukan diet kondisi penyakit komplikasi dan diabetes itu sendiri akan bertambah parah. Diet dilakukan dengan kebutuhan karbohidrat 50%, lemak 30% dan protein 20% dari total kalori harian.

Dan diet ini akan digunakan dalam menyelesaikan penelitian pada permasalahan optimasi bahan makanan ada penderita diabetes dan komplikasinya menggunakan algoritma genetika.

2.5 Algoritma Genetika

Algoritma genetika (*Genetic Algorithms, GAs*) merupakan salah satu jenis algoritma evolusi yang paling populer. Algoritma genetika berkembang seiring dengan perkembangan teknologi informasi yang sangat pesat. Hal ini dikarenakan kemampuan algoritma genetika yang umumnya untuk menyelesaikan berbagai masalah kompleks, algoritma ini banyak digunakan dalam berbagai bidang seperti bidang fisika, biologi, ekonomi, sosiologi dan lain-lain yang sering menghadapi masalah optimasi yang memiliki model matematika kompleks atau sulit untuk dibangun (Mahmudy 2015).

Algoritma genetika merupakan suatu metode heuristic yang berkembang berdasarkan prinsip genetika dan proses seleksi alamiah dari teori evolusi Darwin. Dalam teori evolusi Darwin menjelaskan bahwa suatu individu tercipta secara acak kemudian berkembangbiak melalui proses reproduksi sehingga membentuk sekumpulan individu sebagai suatu populasi (Zukhri, 2014). Menurut (Ganguly, et al., 2009) algoritma genetika merupakan metode yang digunakan mengoptimalkan permasalahan yang dipengaruhi oleh prinsip-prinsip seleksi alam serta genetika alami. Selain itu menurut (Paszkwicz, 2009) algoritma genetika berasal dari memecahkan permasalahan ilmiah dan teknis yang bersumber dari alam. Dalam pelaksanaannya menggunakan *global search* dan dapat mengatasi permasalahan yang memiliki kompleksitas yang tinggi.

2.5.1 Kelebihan Algoritma Genetika

Terdapat beberapa kelebihan algoritma genetika menurut para ahli. Menurut Gen dan Cheng (1997) dalam (Zukhri, 2014), Kelebihan algoritma genetika adalah:

1. Algoritma genetika menggunakan lebih sedikit perhitungan matematis dibandingkan dengan algoritma lain dalam menyelesaikan permasalahan yang berhubungan.
2. Algoritma genetika dapat mengendalikan fungsi objektif dan kendala yang ada, baik pada ruang pencarian diskrit maupun analog.
3. Dalam pencarian global algoritma genetika sangat efektif digunakan karena algoritma genetika memiliki operator-operator evolusi.

4. Memiliki tingkat fleksibilitas yang tinggi jika dihibrid dengan metode pencarian lain sehingga dapat mendapatkan hasil yang efektif.

Menurut Mahmudy (2015), Terdapat beberapa kelebihan dari algoritma genetika yaitu:

1. Algoritma genetika dapat menggunakan sub-populasi yang hanya dilakukan pada komputer sehingga dapat menjaga keragaman populasi dan peningkatan kualitas dalam pencarian.
2. Dalam penyelesaian masalah dengan banyak obyektif algoritma genetika dapat menghasilkan himpunan solusi yang optimal.
3. Beberapa penelitian membuktikan bahwa *hybrid Genetic Algorithm* dapat menghasilkan solusi yang lebih baik dan sangat efektif.

2.5.2 Proses Pencarian Solusi Algoritma Genetika

Menurut (Mahmudy, 2015) Langkah-langkah struktur dasar algoritma genetika yaitu:

1. Inisialisasi populasi
2. Reproduksi populasi
3. Evaluasi populasi
4. Seleksi populasi
5. Mengulang langkah pertama jika syarat perberhentian kromosom belum terpenuhi.

Penjelasan masing-masing proses tersebut adalah sebagai berikut:

2.5.2.2 Inisialisasi

Proses inisialisasi merupakan proses awal yang digunakan untuk membuat populasi awal dari jumlah kromosom. Proses ini dilakukan secara acak / random dan harus ditentukan berapa ukuran populasi (popsize). Menurut Mahmudy (2015) Inisialisasi kromosom dapat dilakukan dengan beberapa cara seperti biner, matriks dan integer. Inisialisasi kromosom integer umumnya digunakan dalam inisialisasi berbagai masalah dalam sehari-hari seperti pada penelitian sebelumnya yang dilakukan pada permasalahan optimasi bahan makanan (Rianawati & Mahmudy, 2015). Pada penelitian tersebut inisialisasi yang digunakan berbasis integer dan hasil yang diperoleh cukup optimal. Adapun contoh representasi kromosom berupa angka integer adalah berikut

$P1 = [2 \ 3 \ 4 \ 1 \ 5]$

Gambar 2.1 Representasi Kromosom

Pada Gambar 2.1 P1 menunjukkan *parent* yang pertama dan angka-angka integer merupakan nomor dari solusi permasalahan. Jika permasalahan yang dibahas adalah *TSP (Travelling Salesman Problem)* maka nomor tersebut merupakan jalur kota yang dimulai dari kota nomor 2 sampai kembali lagi ke

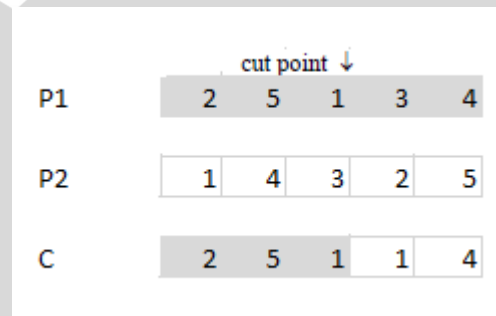
nomor 2. Dalam permasalahan optimasi bahan makanan maka nomor-nomor tersebut adalah nomor bahan makanan.

2.5.2.3 Reproduksi

Proses reproduksi merupakan proses untuk menghasilkan keturunan-keturunan individu dalam populasi. Dalam proses reproduksi yang dihasilkan akan ditampung dalam tempat penampungan *child*. Terdapat dua cara yang dilakukan dalam melakukan proses reproduksi yaitu

1. Crossover (Tukar Silang)

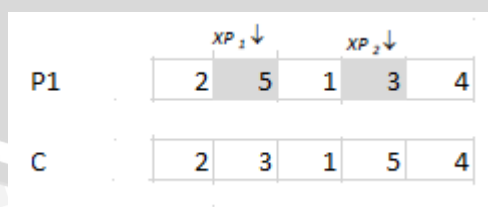
Metode *crossover* membutuhkan dua *parent* dalam melakukan proses reproduksi dan pemilihan *parent* dapat dilakukan secara acak. Metode *crossover* ada dua yaitu *single cut point (one-cut point)* dan *extended intermediate*. *Single cut point (one-cut point)* dilakukan dengan cara memotong titik dari kromosom dan digabung dengan *parent* yang lain yang telah dipilih sebelumnya. Contoh proses *crossover one-cut point* dapat dilihat pada Gambar 2.2



Gambar 2.2 Crossover One-cut Point

2. Mutasi

Metode mutasi hanya membutuhkan satu *parent* dan *child* yang dihasilkan juga satu dimana pemilihan *parent* dilakukan secara acak. Metode mutasi dapat dilakukan dengan metode *reciprocal exchange mutation* yaitu memilih dua titik acak (*XP /Exchange Point*) dari kromosom dan masing-masing titik akan menukar nilai gennya. Contoh mutasi *reciprocal exchange mutation* ditampilkan pada Gambar 2.3



Gambar 2.3 Mutasi Reciprocal Exchange Mutation

Setelah proses crossover dan mutasi selesai, child yang dihasilkan dimasukkan pada populasi dan dilanjutkan proses selanjutnya yaitu evaluasi.

2.5.2.4 Evaluasi

Proses evaluasi merupakan proses menghitung nilai fitness dari setiap kromosom. Semakin besar nilai fitness maka semakin baik kromosom untuk dijadikan calon keturunan baru. Cara perhitungan fitness tergantung dari masing-masing permasalahan. Menurut (Mahmudy, 2015) secara umum perhitungan fitness dibagi dua yaitu rumus maksimum dan rumus minimum. Rumus maksimum digunakan jika ingin mencari nilai fitness yang berbanding lurus dan rumus minimum adalah sebaliknya yaitu mencari nilai fitness yang berbanding terbalik. Adapun rumus maksimum dapat dilihat pada persamaan 2.18 dan rumus minimum dari fitness dapat menggunakan rumus pada persamaan 2.19

$$Fitness = f(X_1, X_2) \quad (2.18)$$

Keterangan:

$f(X_1, X_2)$ = fungsi obyektif dari permasalahan

$$Fitness = \frac{1}{f(x)} \quad (2.19)$$

Keterangan:

$f(x)$ = fungsi objektif dari permasalahan

Sesuai dengan permasalahan optimasi bahan makanan maka rumus yang digunakan adalah rumus minimum karena nilai yang dicari terdapat penalti dan harga yang memiliki nilai berbanding terbalik dengan hasil.

2.5.2.5 Seleksi

Seleksi merupakan proses pemilihan kromosom untuk digunakan pada generasi selanjutnya. Terdapat beberapa cara yang dilakukan dalam melakukan proses seleksi yaitu *elitism selection*, *roulette wheel*, *binary tournament* dan *replacement*. Pada penelitian ini akan menggunakan *elitism selection* yaitu mengurutkan nilai fitness dari yang terbaik sampai yang terendah lalu diambil sesuai ukuran populasi yang dibutuhkan. Seleksi *elitism* bersifat *deterministic* atau bekerja dengan menggunakan aturan yang tetap.

2.5.2.6 Pemberhentian Iterasi

Pemberhentian iterasi adalah kondisi dimana tahapan algoritma genetika harus dihentikan. Terdapat beberapa kondisi yang dapat digunakan dalam pemberhentian iterasi yaitu solusi yang dihasilkan memiliki konvergensi nilai dan ditentukan jumlah generasi yang akan dihitung. Pada penelitian ini pemberhentian iterasi yang digunakan adalah menentukan jumlah generasi yang akan dihitung. Kondisi tersebut digunakan karena pemberhentian kondisi lebih jelas dalam menjalankan prosesnya.

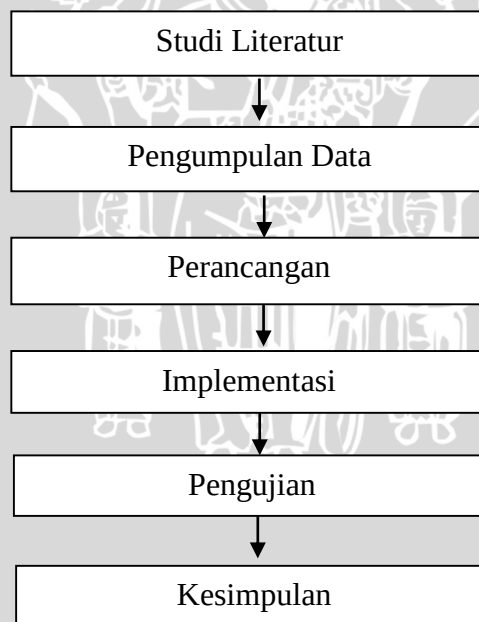
BAB 3 METODOLOGI

Metodologi penelitian menjelaskan mengenai langkah – langkah dalam penelitian untuk memperoleh data dan kemudian data di proses menjadi sebuah informasi masalah yang diteliti.

Berikut ini merupakan langkah-langkah yang dilakukan dalam penelitian ini :

1. Melakukan studi literatur mengenai metode algoritma genetika, kebutuhan nutrisi dan gizi pada penderita penyakit diabetes dan komplikasinya.
2. Melakukan pengumpulan data bahan makanan secara umum, data pasien penderita penyakit diabetes dan komplikasinya
3. Melakukan perancangan
4. Melakukan implementasi dari perancangan yang menghasilkan proses mengoptimalkan makanan pada penderita penyakit diabetes dan komplikasinya
5. Melakukan pengujian perangkat lunak.
6. Membuat kesimpulan.

Langkah – langkah penelitian yang akan dilakukan dapat dilihat pada Gambar 3.1



Gambar 3.1 Langkah-langkah Penelitian

3.1 Studi Literatur

Studi literatur merupakan tahapan pembelajaran yang dilakukan oleh peneliti yang berhubungan dengan permasalahan yang akan diteliti. Studi literature digunakan sebagai dasar dalam menentukan dasar teori dari permasalahan ini yang diteliti. Dengan mempelajari teori-teori tersebut diharapkan dapat digunakan sebagai pendukung untuk membantu penyelesaian masalah. Studi literatur dilakukan dengan menggunakan buku, jurnal, pakar, dan penelitian sebelumnya sebagai sumber media yang digunakan. Studi literatur dari pokok permasalahan meliputi :

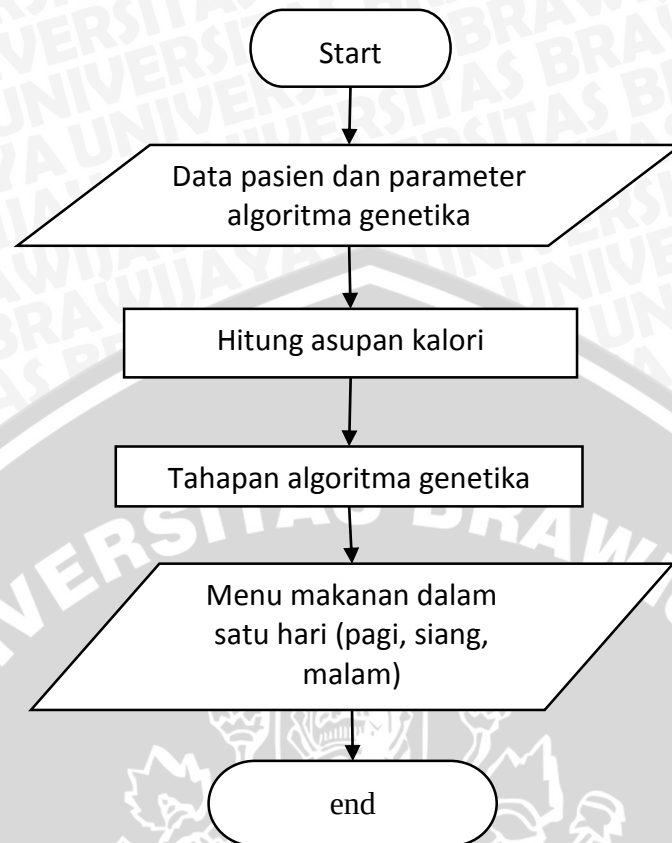
1. Metode Algoritma Genetika
2. Pemahaman tentang penyakit diabetes melitus dan kebutuhan gizi dan nutrisi pada penderita penyakit diabetes melitus.
3. Pemahaman tentang penyakit yang berkomplikasi dengan penderita penyakit diabetes mellitus.
4. Diet yang tepat bagi penderita diabetes melitus dan komplikasinya.

3.2 Pengumpulan Data

Pengumpulan data merupakan tahapan penelitian dimana data-data yang akan digunakan dikumpulkan dan dianalisis sesuai kebutuhan. Data-data yang digunakan dalam penelitian adalah data bahan penakar makanan dari (Republik Indonesia. Peraturan Menteri Kesehatan Nomor 41 Tahun 2014). Data yang dibutuhkan meliputi data gizi dan nutrisi yang sesuai dengan kebutuhan zat gizi baik kebutuhan zat gizi makro atau zat gizi mikro terutama data gizi dan nutrisi pada penderita penyakit diabetes melitus dan komplikasinya.

3.3 Perancangan

Perancangan merupakan perencanaan yang digunakan menunjukkan tahapan-tahapan penelitian yang dilakukan. Perancangan meliputi beberapa tahapan yaitu perancangan algoritma, perancangan *database*, perancangan *user interface* dan perancangan uji coba. Perancangan algoritma yang terdiri dari formulasi masalah, siklus penyelesaian dan manualisasi. Formulasi permasalahan membahas mengenai alasan pentingnya masalah untuk diteliti. Siklus penyelesaian meliputi tahapan yang dilakukan mulai dengan melakukan proses input data berupa data diri dari pengguna meliputi nama, umur, berat badan, jenis kelamin, tinggi badan, aktivitas dan penyakit komplikasi yang diderita. Kemudian data diri pengguna akan dihitung untuk mendapatkan total asupan kalori harian dari penderita berdasarkan rumus 2.1 sampai 2.17. Total asupan kalori akan diproses sesuai langkah-langkah dalam algoritma genetika (mulai dari inialisasi sampai seleksi) sehingga menghasilkan output data makanan yang harus dimakan oleh pengguna dalam waktu satu hari (makan pagi, makan siang, makan malam) dengan masing-masing terdapat lima bahan makanan penyusun seperti sumber makanan pokok, sumber nabati, sumber hewani, sumber sayuran dan pelengkap (buah-buahan). Siklus penyelesaian dapat dilihat pada Gambar 3.2.



Gambar 3.2 Perancangan

3.4 Implementasi

Implementasi merupakan proses penerapan hasil perancangan menjadi bahasa program, dalam masalah ini menggunakan bahasa pemrograman java. Dalam tahapan implementasi terdapat beberapa kebutuhan yang diperlukan baik dari sisi *hardware* maupun *software*. Kebutuhan *hardware* yang diperlukan adalah PC dengan spesifikasi RAM 10 GB, *processor* intel core i3. Kebutuhan *software* yang diperlukan meliputi windows 8 digunakan sebagai sistem operasi, bahasa pemrograman yang digunakan adalah java, editor pemrograman yang digunakan adalah Netbeans IDE dan mysql sebagai *database* atau media penyimpanan data bahan makanan. Implementasi dilakukan dengan beberapa tahap yang terdiri dari:

1. Membuat *database* data makanan yang digunakan untuk menyimpan data-data makanan mulai dari makanan sumber pokok, sumber nabati, sumber hewani, sumber sayuran dan sumber pelengkap.
2. Merancang algoritma genetika sesuai permasalahan optimasi komposisi makanan pada penderita diabetes melitus dan komplikasinya.
3. Membuat *user interface* yang digunakan sebagai antarmuka dengan *user*.
4. Menerapkan perancangan algoritma genetika menjadi program.
5. Mendapatkan *output* berupa menu makanan dalam waktu satu hari.

3.5 Pengujian

Pengujian dilakukan untuk menentukan nilai parameter dari algoritma genetika yang terdiri jumlah individu dalam populasi, generasi, kombinasi *crossover* dan mutasi yang paling optimal sehingga diharapkan mendapatkan hasil menu makanan yang optimal bagi penderita diabetes melitus dan komplikasinya. Data pengujian didapatkan dari data pasien diabetes melitus RS Permata Bunda. Selain menentukan nilai parameter perlu menentukan tingkat *error* dari program yang dibuat. Tingkat eror diperoleh dari menu yang dihasilkan oleh program. Menu tersebut dilakukan pengecekan sesuai kebutuhan seimbang dari data pasien yang diinputkan. Jika total kebutuhan dalam menu sesuai dengan kebutuhan seimbang maka tidak terjadi *error* namun jika kurang atau lebih dari kebutuhan seimbang maka dianggap terdapat *error*. Tingkat eror yang dihasilkan oleh program mempengaruhi kriteria menu makanan yang optimal atau tidak. Menu makanan yang optimal harus memiliki *fitnes* yang tinggi dan tidak terjadi tingkat eror.

3.6 Kesimpulan

Kesimpulan merupakan gagasan yang didapatkan dari penelitian yang dilakukan. Kesimpulan berisi tentang hasil akhir dari semua tahap-tahapan yang telah dilakukan sebelumnya. Kesimpulan berisi jawaban dari rumusan masalah dari penelitian atau hasil yang ingin diketahui dari penelitian yang dilakukan. Dalam penelitian ini kesimpulan berisi representasi kromosom dari permasalahan dan nilai parameter dalam algoritma yang paling optimal.

BAB 4 PERANCANGAN

4.1 Formulasi Permasalahan

Diabetes melitus merupakan penyakit yang sangat mematikan terutama penyakit tersebut telah berkomplikasi dengan penyakit lain terutama penyakit yang mematikan lain seperti penyakit jantung dan stroke. Menurut (Prameswari, 2016) pasien yang menderita penyakit diabetes perlu memperbaiki gaya hidup dan melakukan diet yang tepat sehingga dapat mengurangi komplikasi penyakit pada diabetes serta dapat meringankan kondisi penderita penyakit diabetes. Diet penderita diabetes dilakukan dengan memperhatikan kebutuhan energi pasien. Kebutuhan energi didapatkan dengan perhitungan asupan karbohidrat, protein, dan lemak dengan tepat dan akurat. Perhitungan energi pasien dapat dilakukan dengan manual atau dengan bantuan sistem yang menerapkan suatu algoritma. Jika dilakukan dengan proses manual akan membutuhkan waktu yang lama terutama apabila bahan makanan yang tersedia sangat banyak dan diet diabetes setiap komplikasi memiliki kebutuhan yang berbeda.

Pada bahan makanan keseluruhan berjumlah 156 yang terdiri dari 22 bahan makanan pokok, 12 bahan makanan nabati, 48 bahan makanan hewani, 36 bahan makanan sayuran dan 38 bahan makanan pelengkap. Sebagian daftar bahan makanan serta gizi yang terkandung dalam makanan dapat dilihat pada Tabel 4.1 namun daftar lengkap bahan makanan dapat dilihat pada Lampiran A.

Tabel 4.1 Sumber Makanan

No	Nama Pangan	Id	Kalori	Protein	Karbohidrat	Lemak	harga
			Kkal	gram	gram	Gram	
Sumber makanan pokok (150 gram)							
1	Bihun	1	525	12	120	0	2700
2	Biskuit	2	656.25	15	150	0	2400
3	Havermut	3	583.333	13	133	0	3600
4	Jagung Segar	4	210	5	48	0	900
5	Kentang	5	125	3	29	0	1800
Sumber makanan nabati (50 gram)							
1	Kacang Hijau	23	160	16	12	6	1000
2	Kacang Kedelai	24	160	16	12	6	600
3	Kacang Mete	25	266.66667	26.66667	20	10	5000
4	Kacang Merah	26	160	16	12	6	1200
5	Kacang Tanah	27	200	20	15	7.5	1250

No	Nama Pangan	Id	Kalori	Protein	Karbohidrat	Lemak	harga
			Kkal	gram	gram	Gram	
Sumber makanan hewani (80 gram)							
1	Daging sapi	35	114.2857	0	7	2	8800
2	Daging ayam	36	100	0	7	2	2400
3	Hati Sapi	37	80	0	7	2	6400
4	Ikan Asin	38	266.66667	0	7	2	2400
5	Ikan Teri Kering	39	200	0	7	2	4000
Sumber makanan sayuran (200 gram)							
1	Bayam	83	50	10	2	0	1000
2	Kapri muda	84	50	10	2	0	4800
3	Brokoli	85	50	10	2	0	2400
4	Kembang kol	86	50	10	62	0	3000
5	Kemangi	87	50	10	2	0	3333
Sumber makanan pelengkap (100 gram)							
1	Alpokat	119	100	20	0	0	1000
2	Anggur	120	30.30303	6.06060	0	0	2000
3	Apel merah	121	58.82352	11.7647	0	0	2200
4	Apel malang	122	66.66667	13.3333	0	0	1200
5	Belimbing	123	40	8	0	0	1000

Dengan jumlah bahan makanan yang relatif banyak, perhitungan manual oleh tenaga medis akan mengalami kesulitan dan memiliki waktu relatif lama. Dengan bantuan perhitungan sistem, proses komputasi akan berlangsung relatif lebih cepat. Untuk mendapatkan menu optimal sistem perlu diterapkan suatu algoritma sehingga menghasilkan solusi yang lebih optimal. Salah satu algoritma yang dapat mengatasi masalah optimasi yang memiliki permasalahan kompleks adalah algoritma genetika. Selain memperhatikan kebutuhan energi yang diperlukan pasien, makanan yang dihasilkan diharapkan memiliki biaya yang murah sehingga tidak semakin membebaskan pasien. Dengan beberapa pertimbangan tersebut perhitungan dengan sistem menggunakan algoritma genetika sangat disarankan agar solusi yang dihasilkan tepat dan memiliki biaya yang minimum.

4.2 Siklus Penyelesaian Masalah

Terdapat beberapa proses yang dilakukan dalam penyelesaian masalah optimasi komposisi makanan pada penderita diabetes melitus dan komplikasi dengan menggunakan algoritma genetika. Proses pertama yang dilakukan adalah inisialisasi parameter awal. Parameter awal yang diperlukan adalah data pasien yang bersangkutan dan parameter yang digunakan untuk melakukan tahapan algoritma genetika. Setelah itu sistem menghitung kalori harian pasien dan melakukan proses sesuai tahapan dari algoritma genetika. Tahapan algoritma genetika meliputi representasi kromosom, reproduksi yang terdiri dari *crossover* dan mutasi, menghitung fitness dan seleksi.

Tahapan yang digunakan dalam menyelesaikan permasalahan optimasi makanan pada penderita diabetes melitus dan komplikasinya adalah:

1. Inisialisasi Parameter Awal

Inisialisasi parameter awal dibagi menjadi dua. Data pasien dan parameter algoritma genetika.

- a. Data pasien meliputi nama, umur, jenis kelamin, tinggi badan, berat badan, jenis aktivitas yang dilakukan dan komplikasi penyakit yang diderita.
- b. Parameter algoritma genetika meliputi jumlah generasi, *popsize*, *crossover rate* dan *mutation rate*.

2. Menghitung kalori harian

Perhitungan kalori harian yang dilakukan sesuai dengan data pasien yang diinputkan. Setelah mendapatkan total kalori akan dibagi menjadi porsi karbohidrat, protein, dan lemak sesuai yang dibutuhkan.

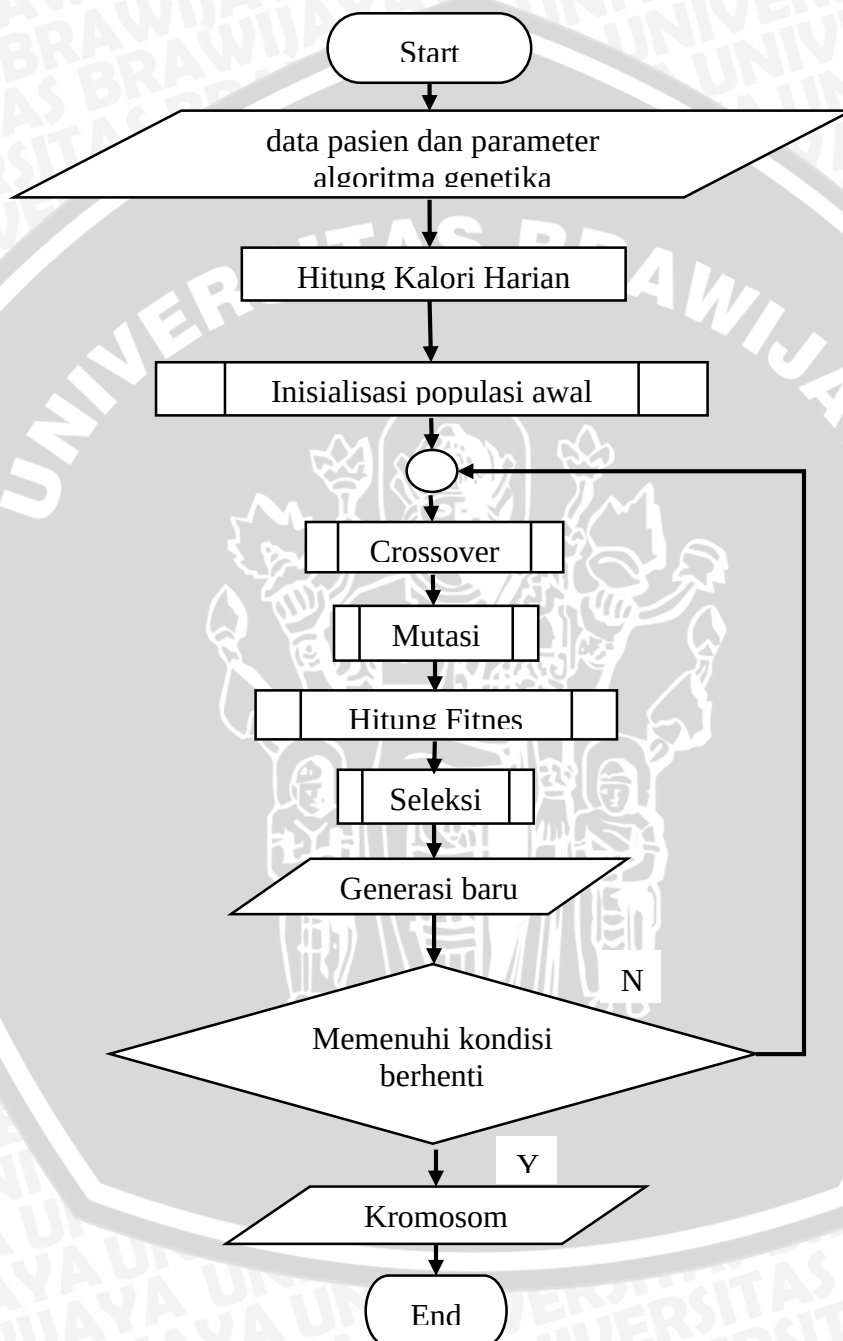
3. Membangkitkan populasi awal sesuai *popsize* dan generasi yang diinputkan

4. Melakukan tahap-tahap penyelesaian sesuai dengan algoritma genetika

- a. Melakukan reproduksi. Reproduksi dibagi menjadi dua yaitu *crossover* dan mutasi.
 1. Proses *crossover* dilakukan dengan metode *one-cut point* yakni memilih *random* individu pada populasi awal dan menghasilkan *child* sesuai *crossover rate* (*cr*) yang diinputkan.
 2. Proses mutasi menggunakan metode *reciprocal exchange mutation* yang menghasilkan *child* sesuai *mutation rate* (*mr*) yang diinputkan.
- b. Menghitung nilai fitness. Perhitungan nilai fitness dilakukan kepada semua individu termasuk *child* yang dihasilkan dari proses reproduksi.
- c. Melakukan seleksi *elitism* sesuai nilai fitness yang diperoleh sebelumnya. Seleksi *elitism* dilakukan dengan memilih individu yang memiliki nilai fitness terbaik. Jumlah individu yang dipilih sesuai dengan jumlah *popsize* yang sebelumnya diinputkan. Individu yang terpilih akan menjadi populasi pada generasi selanjutnya.

- d. Pada generasi selanjutnya proses yang dilakukan sama dengan proses sebelumnya sampai kondisi berhenti terpenuhi. Setelah kondisi terpenuhi individu terbaik pada iterasi terakhir dijadikan sebagai solusi terbaik.

Flowchart tahapan algoritma genetika dalam menyelesaikan masalah optimasi makanan pada penderita diabetes melitus dan komplikasinya ditunjukkan pada Gambar 4.1



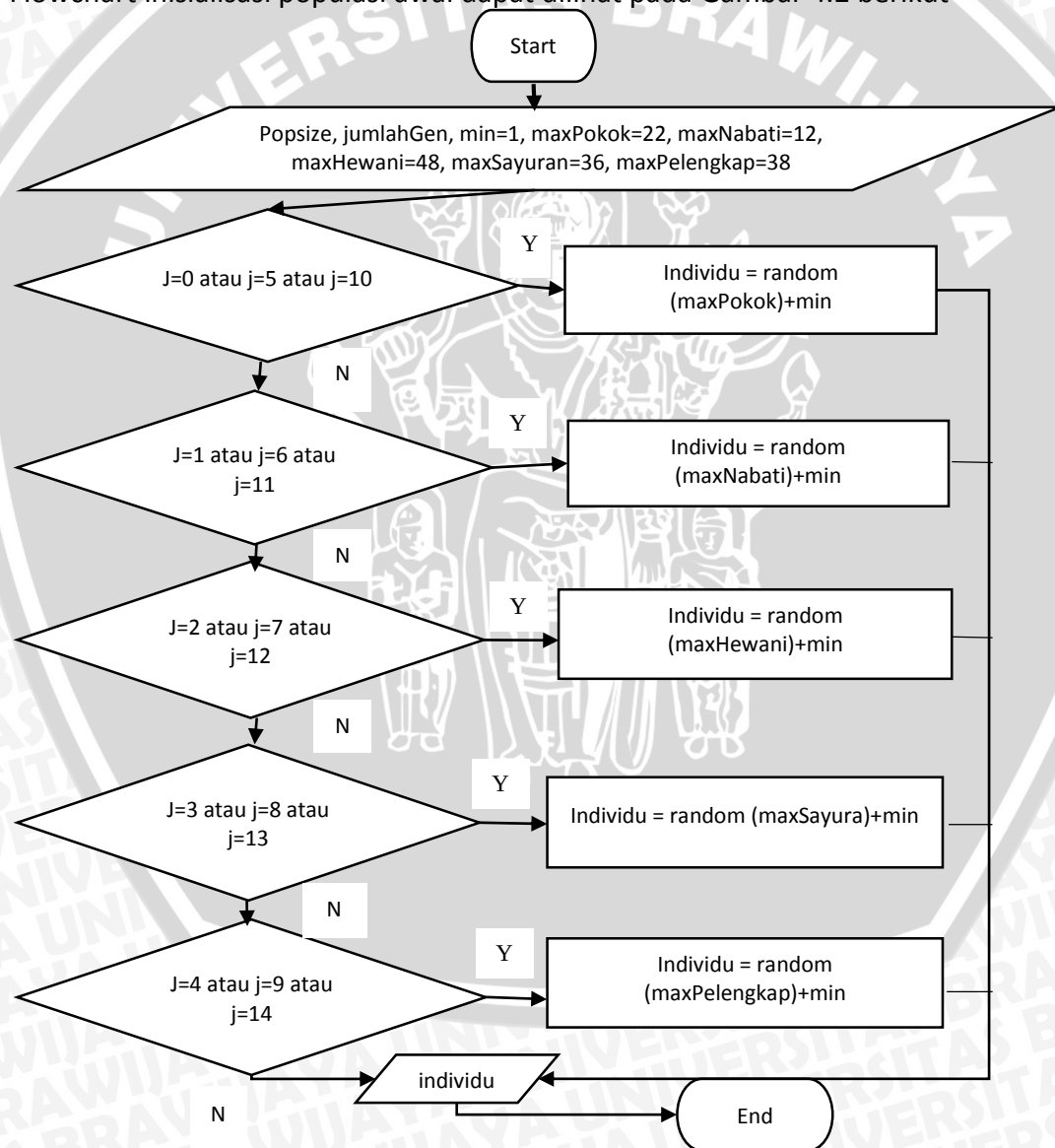
Gambar 4.1 Flowchart Algoritma Genetika

4.2.1 Tahapan Algoritma Genetika

Tahapan algoritma genetika dimulai dengan melakukan inisialisasi populasi awal sesuai jumlah populasi (*popsi*), proses *crossover* dan mutasi, evaluasi (perhitungan *fitnes*) dan seleksi sampai didapatkan individu terbaik.

4.2.1.1 Inisialisasi Populasi Awal

Populasi merupakan kumpulan individu yang saling berkelompok. Dalam hal ini populasi yang dimaksud adalah kumpulan individu yang akan menjadi solusi dari permasalahan. Jumlah populasi awal atau *popsi* ditentukan sesuai data yang diinputkan. Setelah mengetahui jumlah *popsi* akan dibentuk kromosom masing-masing individu dalam *popsi* yang berjumlah 15 gen dan setiap gen dalam kromosom dipilih acak namun dibatasi sesuai jumlah bahan makanan. Flowchart inisialisasi populasi awal dapat dilihat pada Gambar 4.2 berikut



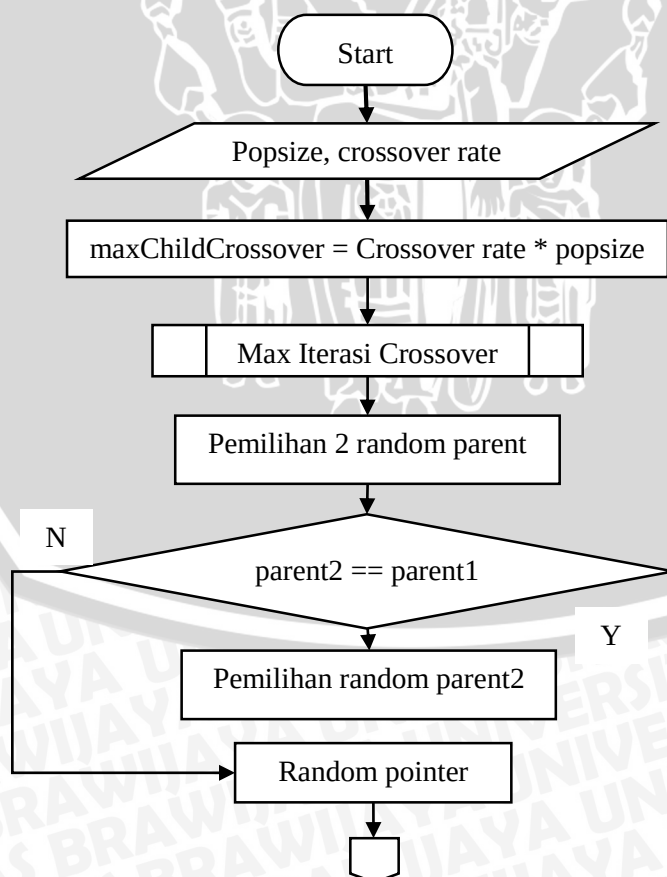
Gambar 4.2 Flowchat inisialisasi populasi awal

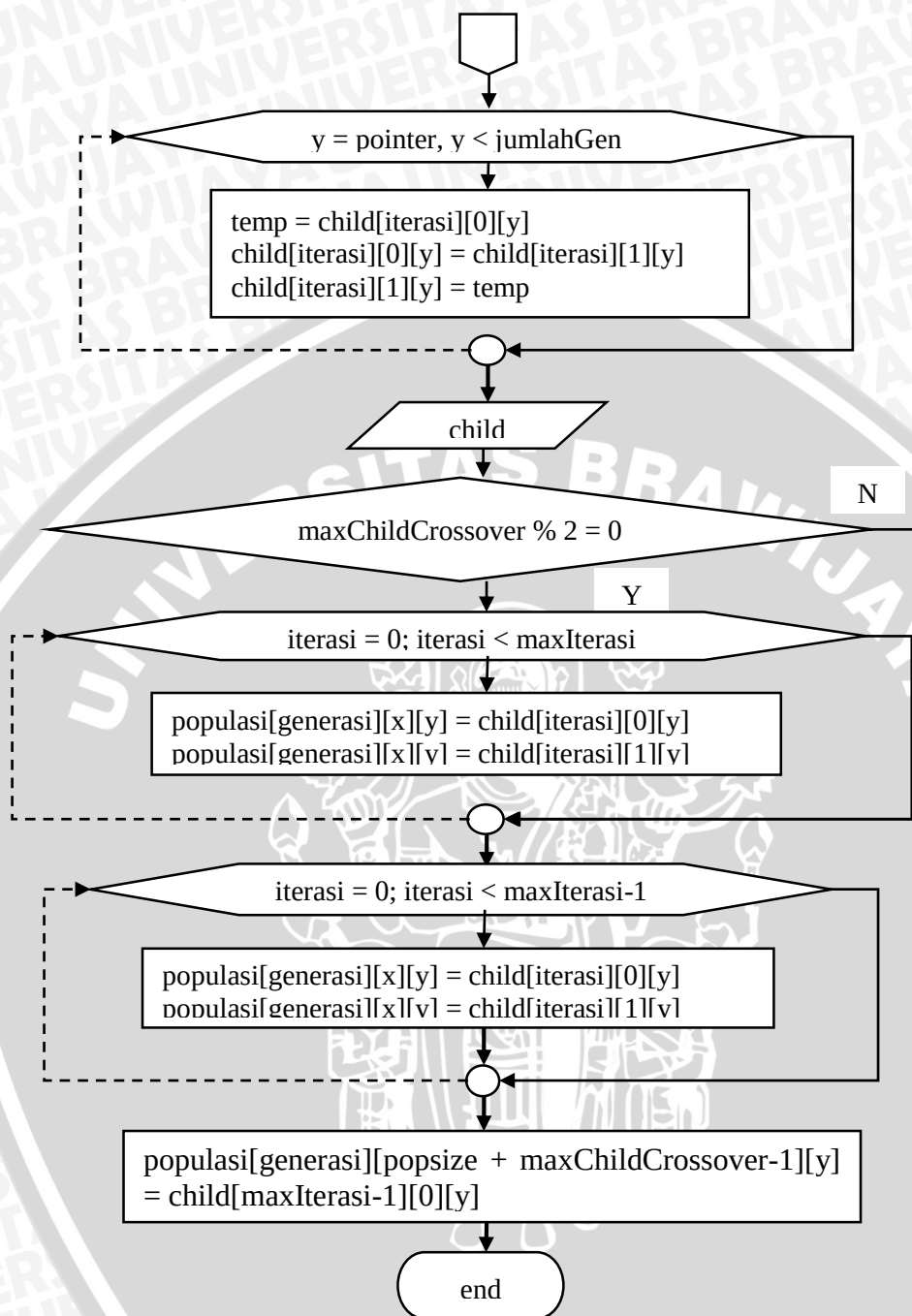
4.2.1.2 Crossover

Setelah melakukan inialisasi awal langkah yang selanjutnya adalah proses reproduksi yang dibagi menjadi dua yakni *crossover* dan mutasi. *Crossover* dilakukan dengan memilih individu yang dikenal dengan parent pada inialisasi populasi awal. Sebelum melakukan *crossover* harus menentukan besar *crossover rate* (*cr*) yang berguna sebagai berapa banyak anak yang dihasilkan pada proses *crossover*. Cara mengetahui banyak anak yang dihasilkan dilakukan dengan perkalian antara *crossover rate* dan *popsiz*e. Metode *crossover* yang dilakukan pada penelitian adalah *one-cut point*. Langkah-langkah metode *crossover one-cut point* adalah

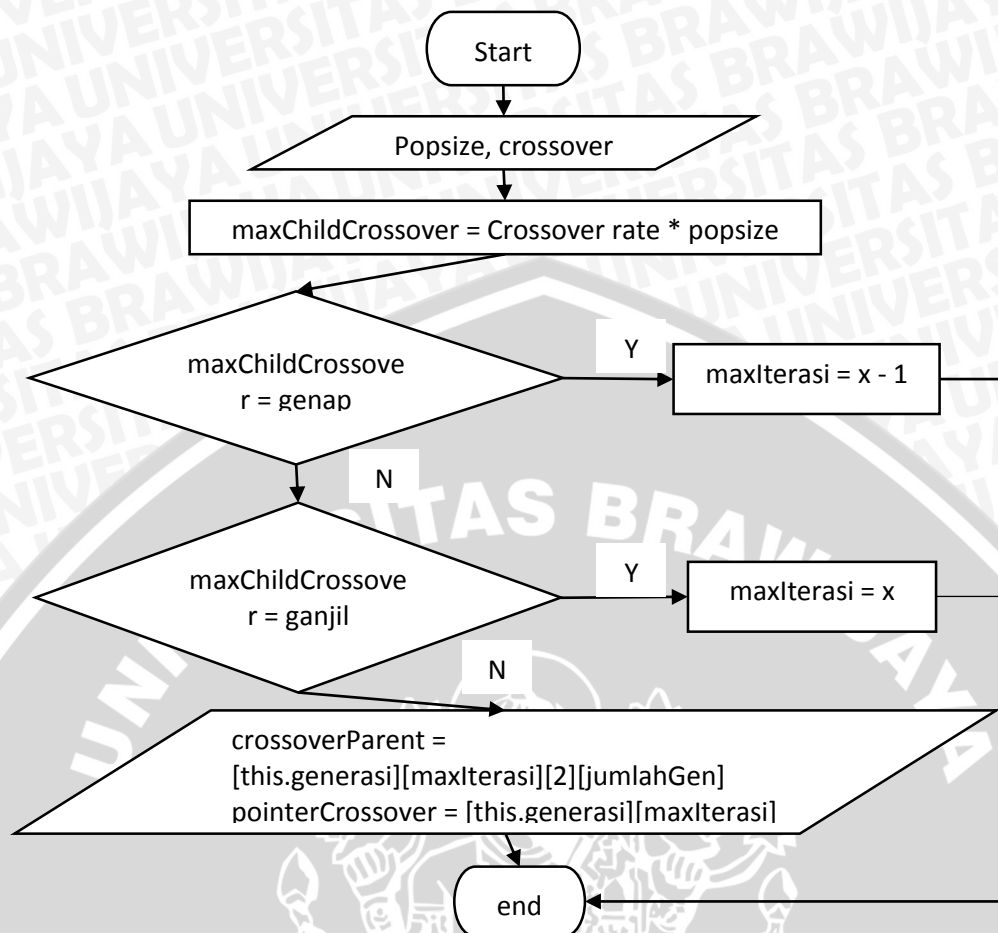
1. Memasukkan jumlah populasi (*popize*) dan *crossover rate*
2. Menghitung jumlah *child crossover* yang akan dihasilkan
3. Memilih 2 individu dari populasi awal secara random dan masing-masing *parent* tidak boleh sama
4. Menentukan titik potong *crossover* secara *random*
5. Setelah mendapatkan titik potong *crossover*, menukar gen-gen yang dibatasi titik potong secara silang dan digabungkan pada *child*.
6. Proses dilanjutkan sampai memperoleh jumlah *child* yang sebelumnya telah dihitung.

Flowchart langkah-langkah metode *crossover one-cut point* dapat dilihat pada Gambar 4.3 dan Gambar 4. 4





Gambar 4.3 Flowchart Crossover



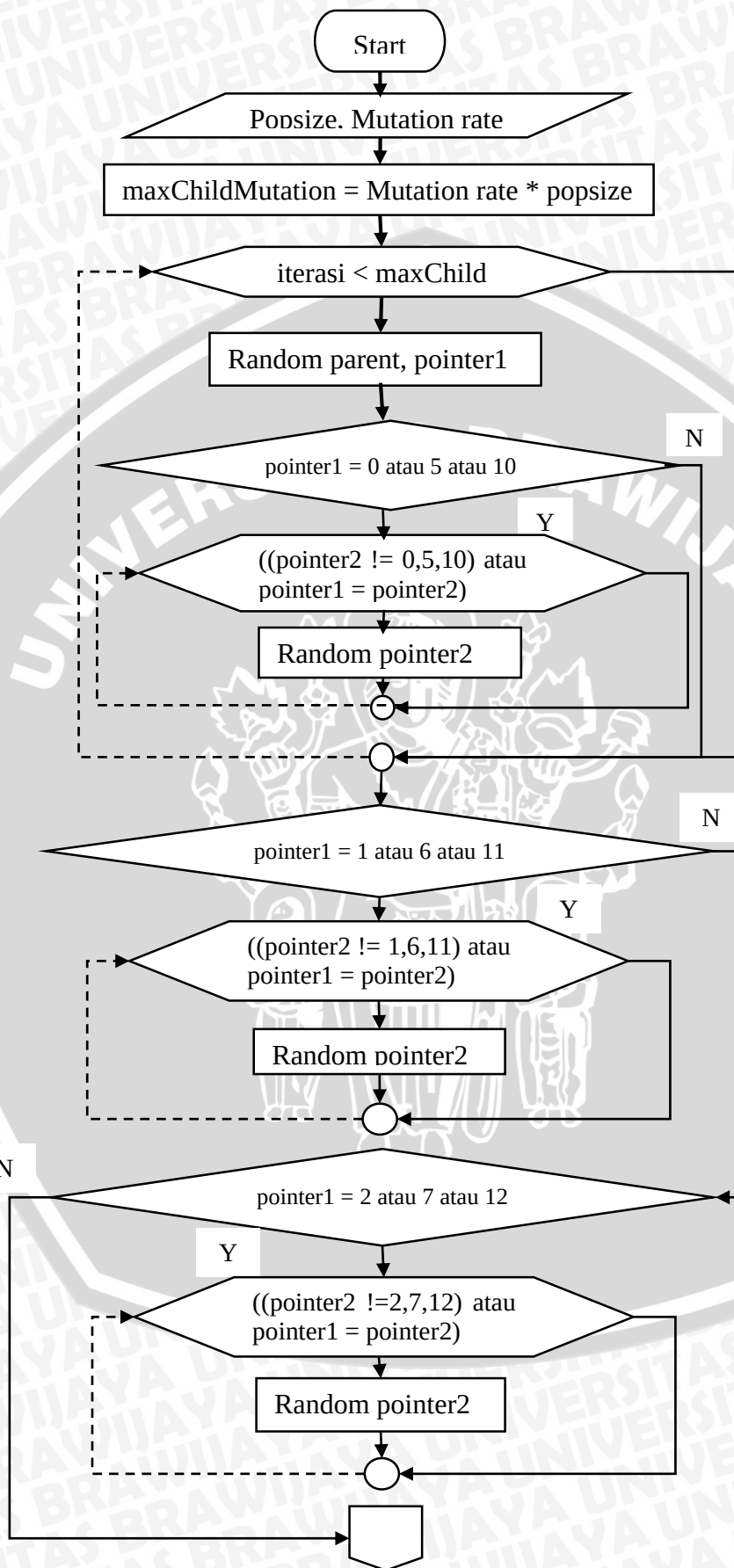
Gambar 4.4 Max Iterasi Crossover

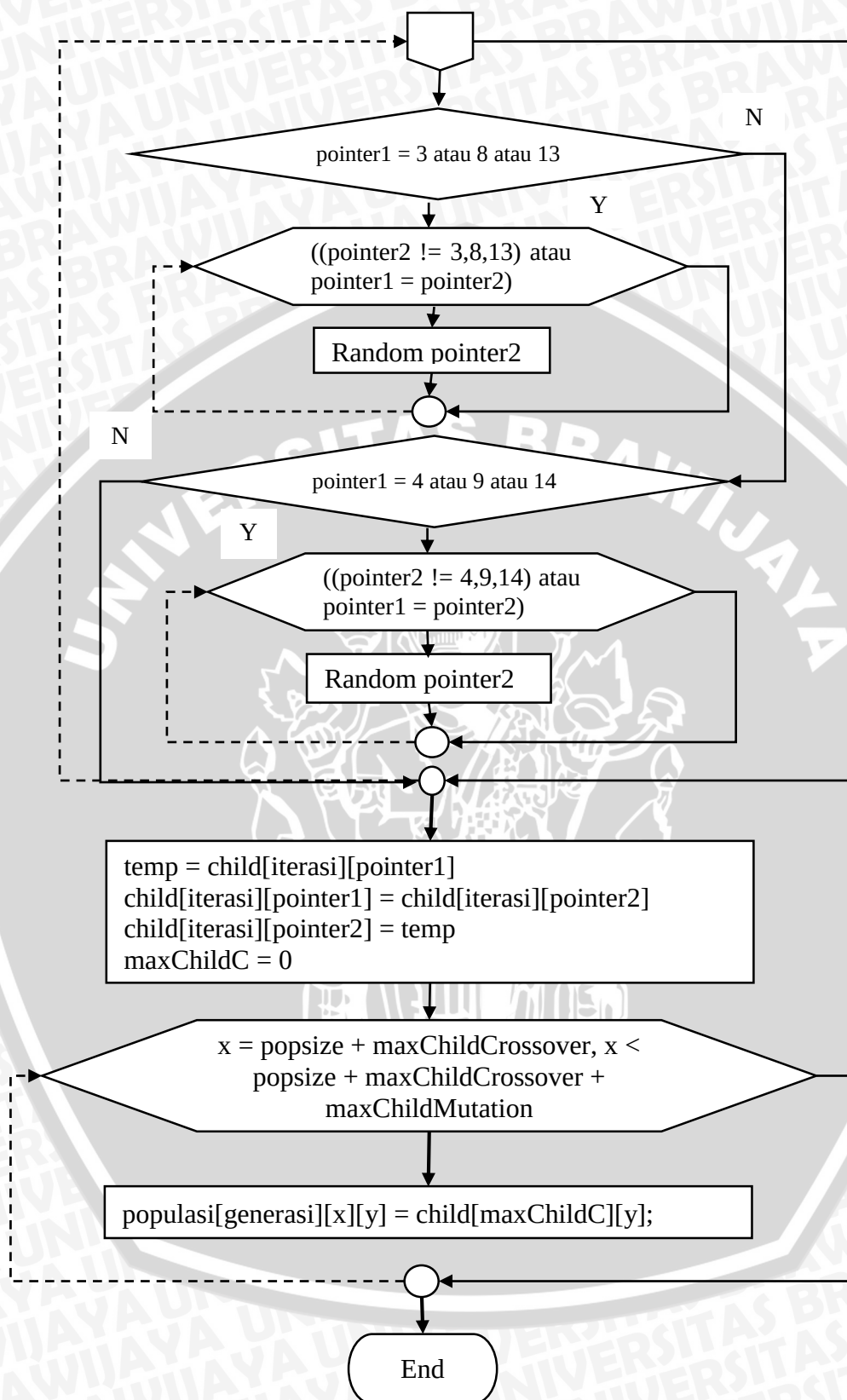
4.2.1.3 Mutasi

Langkah selanjutnya adalah reproduksi mutasi. Mutasi dilakukan dengan memilih satu *parent* untuk melakukan reproduksi. Sebelum melakukan mutasi harus menentukan *mutation rate (mr)* yang digunakan untuk mengetahui jumlah yang dihasilkan pada proses mutasi. Metode mutasi yang dilakukan pada penelitian adalah *reciprocal exchange mutation* bersyarat. Langkah-langkah metode mutasi *reciprocal exchange mutation* bersyarat adalah

1. Memasukkan jumlah populasi dan *mutation rate* lalu menghitung jumlah *child* yang akan dihasilkan.
2. Memilih 1 individu dari populasi awal secara *random*
3. Menentukan 2 titik mutasi secara *random*. Pada titik *random 2* memiliki syarat harus terletak satu jenis bahan makanan yang sama dengan *random* titik 1. Jika titik 1 berada di titik makanan pokok maka titik kedua juga harus berada di makanan pokok namun nilai tidak boleh sama. Jika *random* titik sama maka *child* yang dihasilkan akan sama dengan *parent*.
4. Setelah mendapatkan titik *random*, menukar 2 gen dan hasil disimpan pada *child*. Proses diulang sesuai jumlah *child* yang telah dihitung,

Flowchart langkah-langkah metode metode *reciprocal exchange mutation* dapat dilihat pada Gambar 4.5

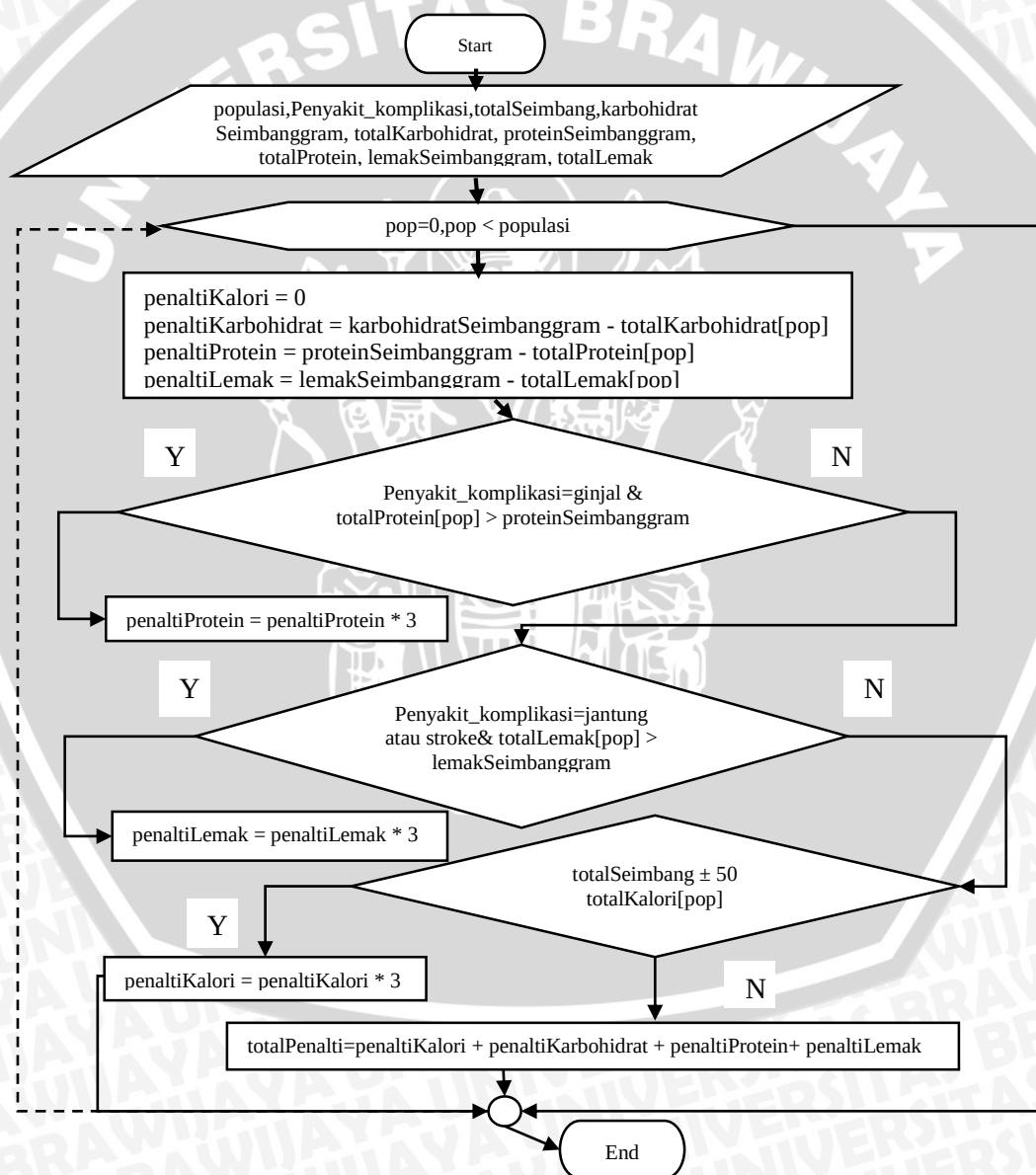




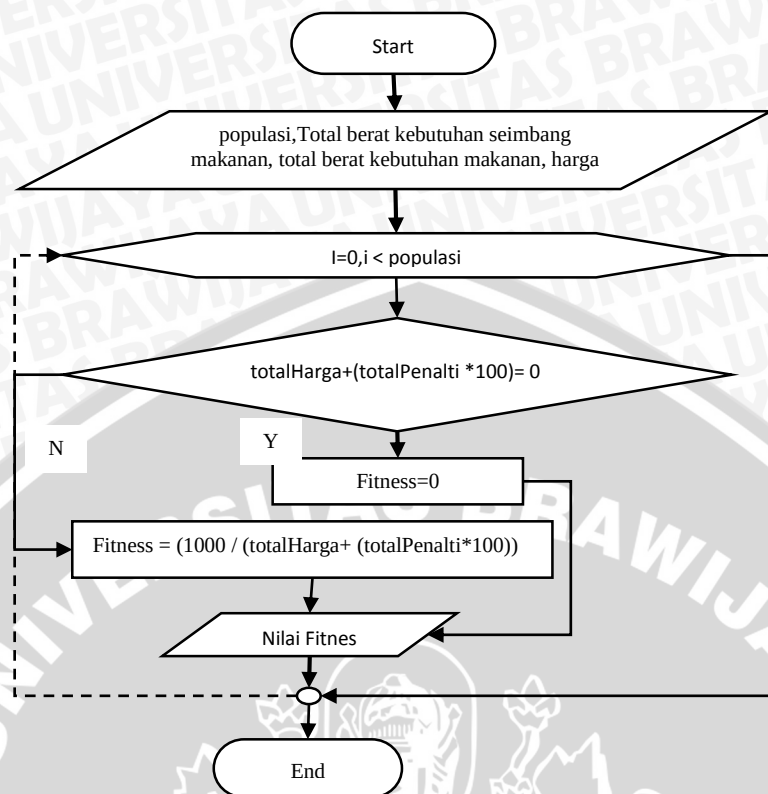
Gambar 4.5 Flowchart Mutasi

4.2.1.4 Evaluasi

Langkah selanjutnya setelah melakukan proses reproduksi *crossover* dan mutasi adalah evaluasi. Proses evaluasi merupakan tahapan algoritma genetika dalam menghitung kebugaran (fitnes). Semakin besar nilai fitnes maka semakin baik solusi yang dihasilkan. Sebelum melakukan perhitungan fitnes terdapat hal yang harus dilakukan yaitu perhitungan penalti. Dalam permasalahan optimalisasi makanan pada penderita diabetes melitus dan komplikasinya penalti adalah kondisi dimana total kandungan bahan makanan tidak sesuai dengan total kandungan bahan makanan seimbang yang telah dihitung sebelumnya. Setelah melakukan perhitungan total penalti dilanjutkan dengan menghitung nilai fitnes setiap individu. Flowchat perhitungan penalti dan fitnes dapat dilihat pada Gambar 4. 6 dan 4.7 berikut



Gambar 4.6 Perhitungan penalti



Gambar 4.7 Flowchart perhitungan fitness

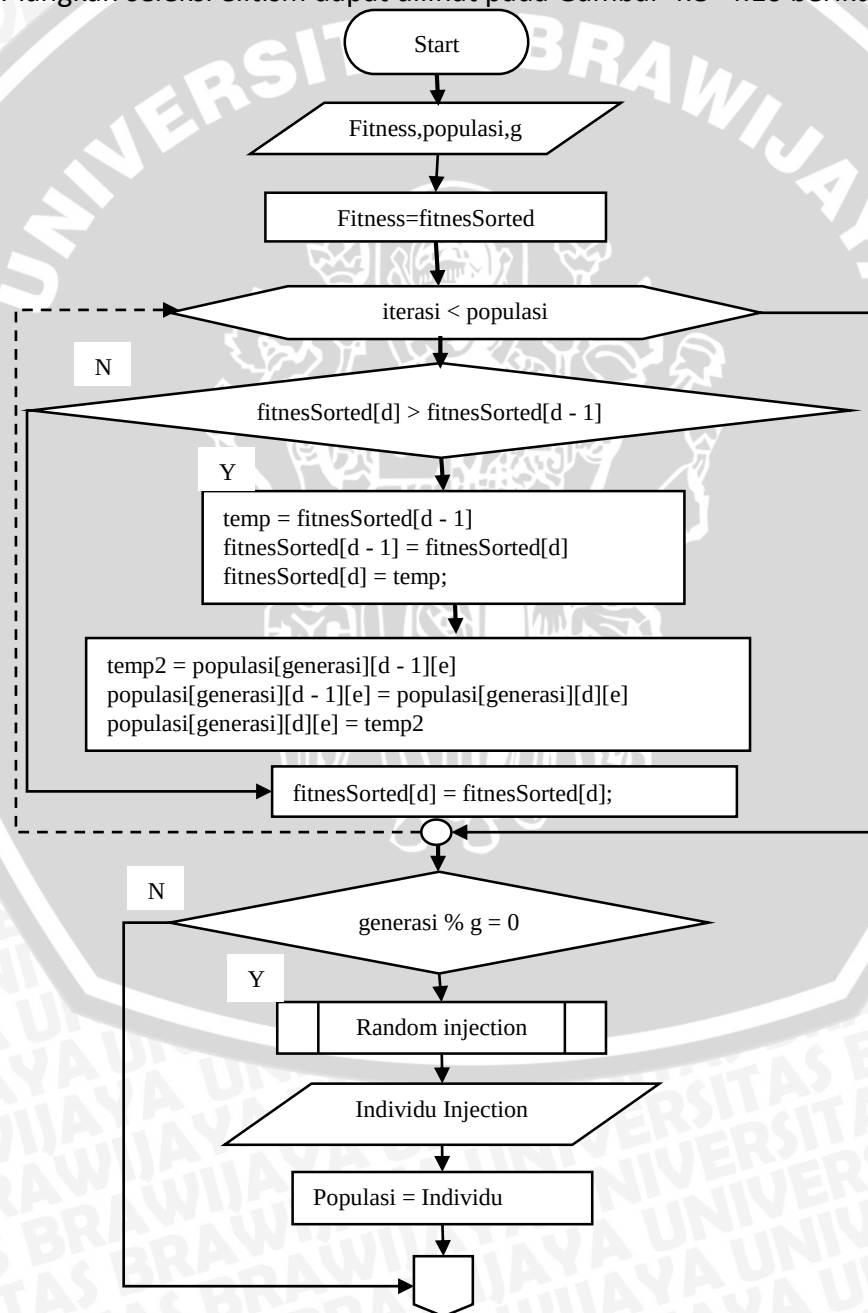
4.2.1.5 Seleksi

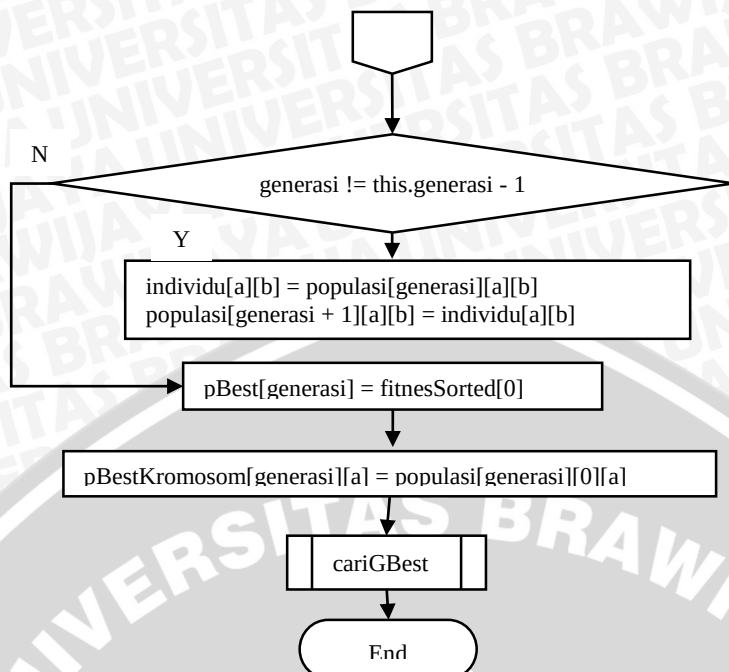
Proses seleksi merupakan proses pemilihan individu yang nantinya akan digunakan pada generasi selanjutnya. Proses seleksi tidak terlepas dari nilai fitness yang telah dihitung pada proses sebelumnya. Semakin besar nilai fitness peluang individu tersebut terpilih akan semakin besar. Semakin besar nilai fitness maka solusi semakin baik dan generasi selanjutnya akan menghasilkan generasi yang lebih baik (Mahmudy, 2015). Terdapat berbagai metode dalam melakukan proses seleksi salah satunya adalah seleksi *elitism*. Seleksi *elitism* disebut juga seleksi deterministic karena individu yang terpilih sudah pasti mengambil individu dengan nilai fitness terbaik saja namun seleksi *elitism* memiliki kekurangan yaitu konvergensi dini. Konvergensi dini merupakan kondisi dimana hampir semua individu memiliki nilai sama atau mengalami perubahan yang tidak signifikan sebelum tercapainya nilai optimum. Penanganan konvergensi sangat penting dilakukan untuk mendapatkan hasil solusi yang optimal. Salah satu metode untuk mengatasi permasalahan konvergensi dini adalah dengan melakukan *random injection*. *Random injection* merupakan proses pembangkitan individu secara *random* seperti pada langkah awal inisialisasi sejumlah n individu. Menurut Mahmudy (2015) $n = 0,1 \times$ jumlah individu biasanya cukup memadai. Proses *random injection* biasanya dilakukan dengan pengecekan nilai fitness setiap generasi jika terdapat konvergensi akan dilakukan *random injection*, namun untuk menghemat waktu komputasi dilakukan proses percobaan. Percobaan tersebut dilakukan dengan mencoba beberapa kali dengan parameter yang sama dan pada

hasil yang didapatkan berupa jumlah generasi yang biasanya terjadi konvergensi nilai. Sesuai jumlah generasi tersebut *random injection* akan dilakukan setiap interval g generasi. Langkah-langkah dalam melakukan seleksi *elitism* adalah sebagai berikut.

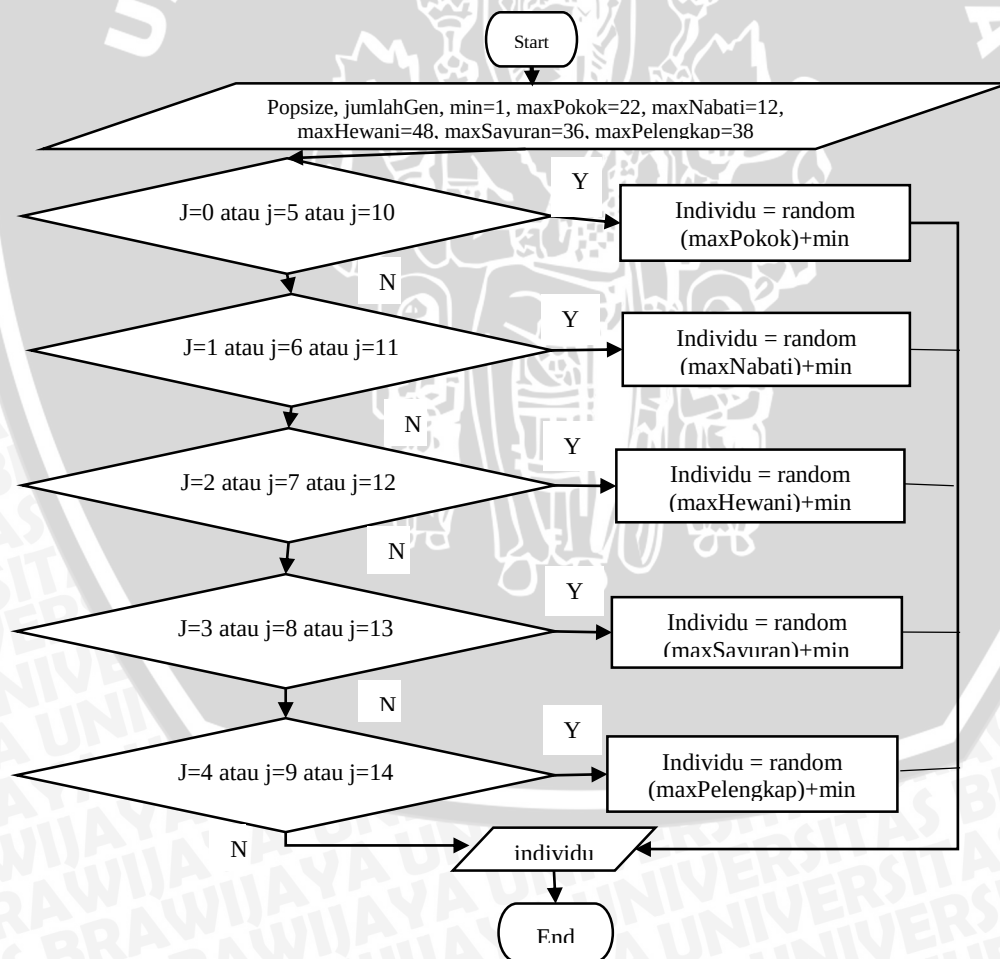
1. Memasukkan nilai *fitnes* yang telah dihitung sebelumnya dan jumlah populasi.
2. Mengurutkan nilai *fitnes* terbaik sampai terendah.
3. Pada generasi kelipatan g dilakukan proses *Random Injection* yaitu memasukkan individu sejumlah n pada generasi. Nilai n didapatkan dengan mengalikan $0.1 \times$ jumlah populasi.
4. Mengambil beberapa individu sesuai dengan jumlah *popsiz*e yang memiliki nilai tertinggi.

Langkah-langkah seleksi elitism dapat dilihat pada Gambar 4.8- 4.10 berikut.

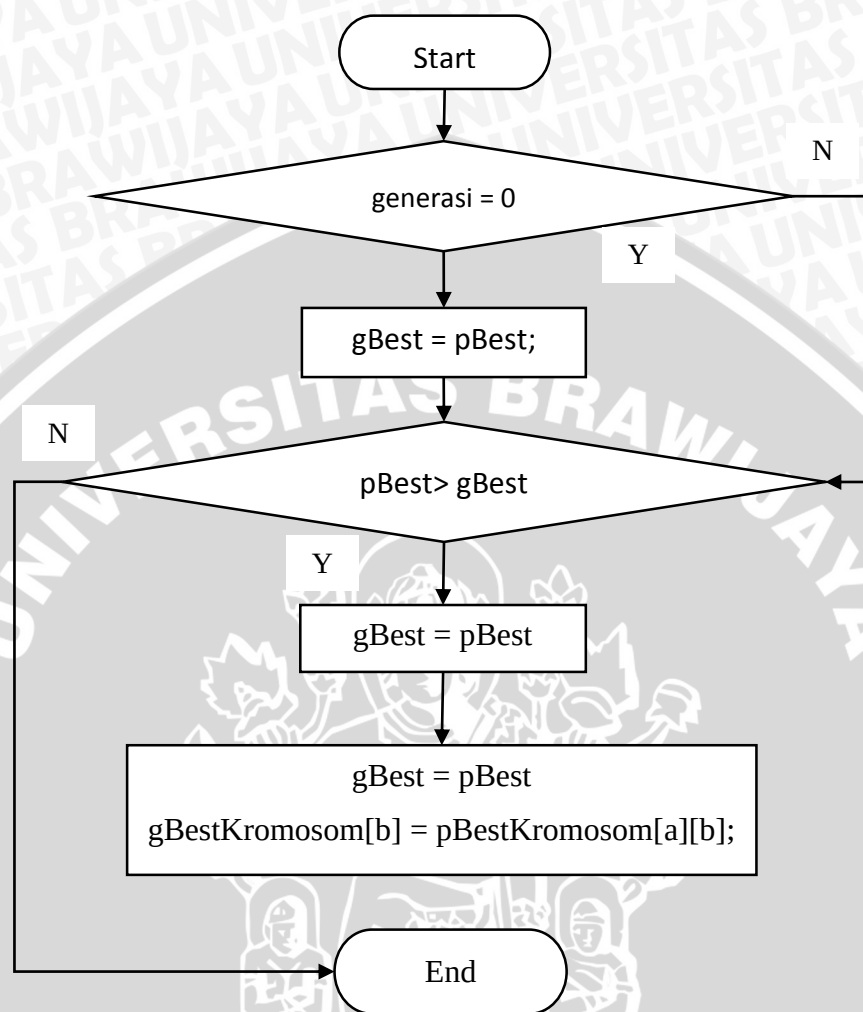




Gambar 4.8 Flowchart Seleksi



Gambar 4.9 Random Injection



Gambar 4.10 CariGbest

4.2.1.6 Memilih Individu Terbaik

Setelah melakukan tahapan-tahapan algoritma genetika sampai kondisi berhenti, langkah selanjutnya yang dilakukan adalah memilih individu yang terbaik. Individu terbaik biasanya memiliki fitness yang tinggi dan individu terbaik nantinya akan menjadi solusi yang dianggap paling optimal.

4.2.2 Desain Kromosom

Desain atau representasi kromosom yang digunakan pada penelitian ini adalah representasi kromosom bilangan integer berbasis nomor makanan yang dibagi sesuai dengan jenis makanannya. Terdapat 15 gen penyusun kromosom setiap individu. 15 gen tersebut mewakili 5 gen untuk makan pagi, 5 gen untuk makan siang dan 5 gen makan malam. Masing-masing 5 gen terdiri dari sumber

makanan pokok, sumber nabati, sumber hewani, sumber sayuran dan pelengkap dengan porsi bobot berbeda yang dapat dilihat pada tabel 4.2.

Tabel 4.2 Porsi bobot makanan

Jenis Makanan	Bobot Makanan (gram)
Sumber Makanan Pokok	150
Sumber Nabati	50
Sumber Hewani	80
Sumber Sayuran	200
Pelengkap	100

Representasi kromosom dilakukan pada bahan makanan dalam waktu makan pagi, makan siang, makan malam. Setiap bahan makanan mengandung karbohidrat, protein dan lemak. Adapun representasi kromosom pada permasalahan optimasi makanan penderita diabetes melitus dan komplikasi dapat dilihat pada Gambar 4.11 berikut

SP				SN				SH				SS				P			
E1	Kb1	Pr1	L1	E2	Kb2	Pr2	L2	E3	Kb3	Pr3	L3	E4	Kb4	Pr4	L4	E5	Kb5	Pr5	L5

Gambar 4.11 Representasi setiap bahan makanan

Keterangan:

SP = Sumber Makanan Pokok

SS = Sumber Sayuran

SN = Sumber Nabati

P = Pelengkap

L1, L2, L3, L4, L5 = Berat bobot lemak

SH = Sumber Hewani

E1, E2, E3, E4, E5 = Berat Energi

Kb1, Kb2, Kb3, Kb4, Kb5 = Berat bobot karbohidrat

Pr1, Pr2, Pr3, Pr4, Pr5 = Berat bobot protein

Contoh representasi kromosom permasalahan optimasi komposisi bahan makanan pada penderita diabetes melitus dan komplikasinya menggunakan algoritma genetika ditampilkan pada Gambar 4.12

Makan pagi					Makan Siang					Makan Malam				
SP	SN	SH	SS	P	SP	SN	SH	SS	P	SP	SN	SH	SS	P
22	1	13	5	4	14	5	40	36	16	12	3	32	24	1

Gambar 4.12 Contoh Representasi Kromosom

4.2.3 Desain Fitnes

Sebelum membahas desain fitnes, terdapat perhitungan penalti yang nantinya akan digunakan dalam perhitungan fitnes. Cara melakukan perhitungan penalti adalah menghitung selisih total kandungan kalori, karbohidrat, protein,

dan lemak dengan kandungan kalori, gizi seimbang karbohidrat, protein dan lemak. Cara perhitungan penalti dapat dilihat pada persamaan 4.1 berikut.

$$\text{Total penalti} = \text{penalti}_1 + \text{penalti}_2 + \text{penalti}_3 + \text{penalti}_4 \quad (4.1)$$

Keterangan:

penalti_1 = Penalti kalori

penalti_2 = Penalti karbohidrat

penalti_3 = Penalti protein

penalti_4 = Penalti lemak

Penalti merupakan kondisi dimana terdapat nilai yang tidak sesuai dengan aturan. Penalti sangat penting dalam melakukan perhitungan diet diabetes melitus dan komplikasinya. Penalti digunakan untuk menghindari hasil makanan yang melebihi asupan energi dari pasien. Terdapat beberapa aturan yang harus diterapkan dalam permasalahan optimasi makanan pada penderita diabetes melitus dan komplikasinya adalah kandungan kalori, karbohidrat, protein dan lemak tidak kurang atau lebih dari kandungan gizi seimbang kalori, karbohidrat, protein dan lemak. Aturan penalti dapat dilihat pada tabel 4.3

Tabel 4.3 Aturan Penalti

No	Aturan	Penalti
1	Total kebutuhan kalori pada makanan, lebih atau kurang dari kebutuhan seimbang kalori	Penalti kalori = (kebutuhan seimbang kalori – total kebutuhan kalori makanan) * α_1
2	Total kebutuhan karbohidrat pada makan pagi, siang dan malam, lebih atau kurang dari kebutuhan seimbang karbohidrat	Penalti karbohidrat = (kebutuhan seimbang karbohidrat – total kebutuhan karbohidrat makanan) * α_2
3	Total kebutuhan protein pada makan pagi, siang dan malam, lebih atau kurang dari kebutuhan seimbang protein	Penalti protein = kebutuhan seimbang protein – total kebutuhan protein makanan) * α_3
4	Total kebutuhan lemak pada makan pagi, siang dan malam, lebih atau kurang dari kebutuhan seimbang lemak	Penalti lemak = kebutuhan seimbang lemak – total kebutuhan lemak makanan) * α_4

Keterangan

α_1 = Bobot penalti kalori

α_2 = Bobot penalti karbohidrat

α_3 = Bobot penalti protein

α_4 = Bobot penalti lemak

Bobot penalti kalori, karbohidrat, protein, lemak mempunyai bobot yang berbeda. Bobot penalti kalori memiliki bobot terbesar karena kalori merupakan penalti yang sangat serius. Namun bobot karbohidrat, protein dan lemak memiliki bobot yang berbeda sesuai dengan penyakit komplikasi yang diderita. Hal ini berdasarkan pengetahuan pakar bahwa terdapat kondisi dimana penyakit komplikasi harus memenuhi kebutuhan wajib. Kondisi bobot penalti setiap komplikasi adalah

1. Komplikasi penyakit ginjal

Pada komplikasi dengan penyakit ginjal, kebutuhan protein merupakan kebutuhan yang sangat penting untuk diperhatikan. Hal ini dikarenakan asupan protein seimbang dari pasien yang memiliki penyakit komplikasi dengan ginjal harus memenuhi kebutuhan seimbang jika melebihi akan memperparah kondisi pasien. Sehingga nilai penalti dari pasien yang memiliki penyakit komplikasi ginjal adalah nilai $\alpha_1 = 3$, $\alpha_2 = 1$, $\alpha_3 = 3$ dan $\alpha_4 = 1$.

2. Komplikasi penyakit jantung dan stroke

Pada komplikasi penyakit jantung dan stroke, kebutuhan makanan yang harus diperhatikan adalah lemak. Total lemak harus memenuhi kebutuhan lemak seimbang. Sangat tidak dianjurkan pasien penderita jantung dan stroke untuk mengkonsumsi lemak seimbang untuk tidak memperparah penyakit yang diderita oleh pasien. Sehingga penalti penderita penyakit komplikasi jantung dan stroke adalah nilai $\alpha_1 = 3$, $\alpha_2 = 1$, $\alpha_3 = 1$ dan $\alpha_4 = 3$.

3. Komplikasi penyakit mata, syaraf, lambung dan tidak komplikasi

Penyakit diabetes melitus yang berkomplikasi dengan penyakit mata, syaraf dan lambung serta yang tidak memiliki komplikasi hanya melakukan aturan total kalori tidak melebihi total kalori seimbang. Sehingga penalti pada penyakit komplikasi mata, syaraf, lambung, dan tidak berkomplikasi adalah $\alpha_1 = 3$, $\alpha_2 = 1$, $\alpha_3 = 1$ dan $\alpha_4 = 3$.

Setelah menghitung nilai penalti, selanjutnya menghitung nilai fitness dengan rumus

$$\text{Fitness} = \frac{1000}{(\text{Penalti} \times 100) + \sum \text{harga}} \quad (4.2)$$

Pada rumus fitness terdapat angka 1000 yang artinya untuk menyetarakan harga dari bahan makanan berkisar antara ribuan. Pada penalti dikalikan 100 untuk memperbesar nilai dari penalti sehingga nilai fitness yang dihasilkan akan kecil dan kemungkinan solusi terpilih akan kecil. Angka 100 berasal dari percobaan yang dilakukan oleh peneliti untuk mendapatkan nilai fitness yang memberatkan solusi agar tidak terpilih namun tetap menjaga agar nilai keseluruhan fitness tidak terlalu kecil. Selain itu terdapat jumlah harga yang diperoleh dari total harga semua bahan makan pada setiap kromosom.

4.3 Manualisasi

Contoh kasus implementasi algoritma genetika pada permasalahan optimasi makanan pada penderita diabetes melitus dan komplikasi sebagai berikut.

1. Inisialisasi Parameter Awal

a. Data Pasien

Nama pasien : Abdullah
 Jenis kelamin : Laki-laki
 Usia : 52 Tahun
 Berat badan : 65 kg
 Tinggi badan : 165 cm
 Aktivitas : Keadaan istirahat
 Komplikasi : Ginjal

b. Parameter Algoritma Genetika

Jumlah generasi = 1
 Ukuran populasi (*popsize*) = 10
 Crossover rate (*cr*) = 0,5
 Mutation rate (*mr*) = 0,2

2. Menghitung Kalori harian

BBI = $0,9 * (165-100) * 1 \text{ kg} = 58,50 \text{ kg}$
 IMT = $65 / (1,65)^2 = 23,875115 \text{ kg}$
 AMB Pria = $58,50 \text{ kg} \times 30 \text{ Kkal/kg} = 1755 \text{ Kkal}$
 AMB Umur = $- 5 \% \times 1755 \text{ Kkal} = 87,75 \text{ Kkal}$
 TEE ringan = $10\% \times 1755 \text{ Kkal} = 175,5 \text{ Kkal}$
 AMB Berat badan = $20 \% \times 1755 \text{ Kkal} = 351 \text{ Kkal}$

Sehingga total kalori harian yang dibutuhkan oleh pasien diabetes melitus adalah

Total Kalori = $1755 - 87,75 + 175,5 + 351 = 2193,75 \text{ Kkal}$

Karbohidrat (Kkal) = $58\% \times 2193,75 \text{ Kkal} = 1272,375 \text{ Kkal} = 318,094 \text{ gr}$

Lemak (Kkal) = $30\% \times 2193,75 \text{ Kkal} = 658,125 \text{ Kkal} = 164,5313 \text{ gr}$

Protein (Kkal) = $12\% \times 2193,75 \text{ Kkal} = 263,25 \text{ Kkal} = 29,22075 \text{ gr}$

Dari perhitungan kalori harian diperoleh kebutuhan karbohidrat 372,9375 gram, Lemak 164,5313 gram, dan protein 29,22075 gram. Setelah menghitung kalori harian sistem melakukan proses penyelesaian masalah optimasi makanan

pada penderita diabetes melitus sesuai dengan tahapan algoritma genetika sebagai berikut.

4.3.3 Representasi Kromosom

Contoh representasi kromosom dapat dilihat pada Gambar 4.13 berikut

Makan pagi					Makan Siang					Makan Malam				
SP	SN	SH	SS	P	SP	SN	SH	SS	P	SP	SN	SH	SS	P
22	1	13	5	4	14	5	40	36	16	12	3	32	24	1

Gambar 4.13 Contoh representasi kromosom individu

Dari contoh representasi kromosom individu Gambar 4.13 dapat dilihat bahwa kromosom dibagi menjadi 15 gen penyusun yang terdiri dari 5 gen makan pagi, 5 gen makan siang dan 5 gen makan malam. Masing-masing 5 gen mewakili sumber makanan pokok, sumber nabati, sumber hewani, sumber sayuran dan pelengkap atau buah-buahan. Penomoran pada Gambar 4.13 mewakili nomor bahan makanan. Penomoran tersebut dibatasi sesuai dengan jumlah bahan makanan yang tersedia yaitu sumber makanan pokok berjumlah 22 sehingga batas nomor untuk sumber makanan pokok memiliki batas antara 1-22. Sumber nabati 1-12, sumber hewani 1-47, sumber sayuran 1-35 dan pelengkap 1-37. Pada menu makan pagi kolom pertama bernomor 22 merepresentasikan sumber makanan pokok berupa ubi jalar kuning dengan bobot pada Tabel 4.4, nomor 1 merepresentasikan sumber nabati kacang hijau, nomor 13 merepresentasikan sumber hewani gabus kering. Nomor 5 merepresentasikan sumber sayuran kemangi, dan nomor 4 merepresentasikan pelengkap apel malang.

Tabel 4.4 Detail Kromosom

Individu	Bahan makanan	Energi (Kkal)	Karbohidrat (gram)	Protein (gram)	Lemak (gram)	Harga (Rp)
Makan pagi	Ubi jalar kuning	194.444	4	44	0	750
	Kacang hijau	160	16	12	6	1000
	Ikan gabus kering	400	0	7	2	2400
	Kemangi	50	10	2		3333
	Apel malang	66.6667	13.3333	0	0	1200
Makan siang	Nasi ketan putih	262.5	6	60	0	3000
	Kacang tanah	200	20	15	7.5	1250
	Daging bebek	266.66667	0	7	13	2880
	Daun singkong	100	20	6	0	1000
	Kedondong	50	10	0	0	800
Makan malam	Beras giling hitam	262.5	6	60	0	3300
	Kacang mete	266.66667	26.66667	20	10	5000

Individu	Bahan makanan	Energi (Kkal)	Karbohidrat (gram)	Protein (gram)	Lemak (gram)	Harga (Rp)
Makan malam	Udang segar	114.28571	0	7	2	5600
	Labu siam	50	10	2	0	1400
	alpokat	100	20	0	0	1000

4.3.4 Inisialisasi Populasi Awal

Inisialisasi populasi awal atau pembangkitan kromosom dilakukan dengan membuat individu sejumlah *popsiz*e dan nilai gen dalam individu tidak boleh melebihi jumlah bahan makanan. Contoh inisialisasi populasi awal pada kasus dengan *popsiz*e 10 pada tabel 4.5

Tabel 4.5 Inisialisasi Populasi Awal

individu	Kromosom														
	Makan Pagi					Makan siang					Makan malam				
	SP	SN	SH	SS	P	SP	SN	SH	SS	P	SP	SN	SH	SS	P
P1	7	4	35	20	25	19	9	21	8	30	15	7	13	8	3
P2	1	3	21	33	5	22	9	11	14	29	7	6	44	12	2
P3	21	0	12	4	3	13	4	39	35	15	11	2	31	23	0
P4	15	12	5	35	15	2	10	35	17	31	6	4	23	14	22
P5	17	4	1	7	15	19	3	26	29	3	11	9	25	7	38
P6	8	6	9	4	20	3	2	15	11	7	1	10	16	34	13
P7	20	9	45	18	14	4	6	43	27	25	5	11	34	23	38
P8	9	10	29	3	38	13	4	15	16	1	20	12	22	17	9
P9	2	5	36	28	32	11	8	3	11	5	19	1	48	25	20
P10	3	11	14	19	10	7	7	48	21	4	22	5	8	2	21

Keterangan

Individu = nomor dari inisialisasi populasi awal

4.3.5 Crossover

Proses *crossover* sesuai contoh kasus bahwa *crossover rate* adalah 0.5 sehingga *child* yang dihasilkan adalah $0.5 \times 10 = 5$ *child*. Dengan memilih dua *parent* secara acak yaitu P4 dan P7 seperti Gambar 4.14 menghasilkan *child* 1 dan 2 yakni Gambar 4.15, P1 dan P9 menghasilkan *child* 3 dan 4 pada Gambar 4.16 dan 4.17, P3 dan P10 menghasilkan *child* 5 seperti pada Gambar 4.18 dan 4.19.

P4	15	12	5	35	15	2	10	35	17	31	6	4	23	14	22
P7	20	9	45	18	14	4	6	43	27	25	5	11	34	23	38

Gambar 4.14 Parent crossover 1

Child 1	15	12	5	35	15	2	10	43	27	25	5	11	34	23	38
Offspring 2	20	9	45	18	14	4	6	35	17	31	6	4	23	14	22

Gambar 4.15 Hasil child crossover 1

P1	7	4	35	20	25	19	9	21	8	30	15	7	13	8	3
P9	2	5	36	28	32	11	8	3	11	5	19	1	48	25	20

Gambar 4.16 Parent crossover 2

Child 3	7	4	35	20	25	19	9	21	8	30	15	1	48	25	20
Offspring 4	2	5	36	28	32	11	8	3	11	5	19	7	13	8	3

Gambar 4.17 Hasil child crossover 2

P3	22	1	13	5	4	14	5	40	36	16	12	3	32	24	1
P10	3	11	14	19	10	7	7	48	21	4	22	5	8	2	21

Gambar 4.18 Parent crossover 3

Offspring 5	22	1	14	19	10	7	7	48	21	4	22	5	8	2	21
-------------	----	---	----	----	----	---	---	----	----	---	----	---	---	---	----

Gambar 4.19 Hasil child 3

4.3.6 Mutasi

Proses mutasi sesuai contoh kasus, *mutation rate* adalah 0.2 sehingga *child* yang dihasilkan adalah $0.2 \times 10 = 2$. Dengan memilih *parent* pada populasi awal secara acak yaitu P2 seperti pada Gambar 4.20 menghasilkan *child* 6 yakni Gambar 4.21, P5 menghasilkan *child* 7 pada Gambar 4.22 dan 4.23.

P2	1	3	21	33	5	22	9	11	14	29	7	6	44	12	2
----	---	---	----	----	---	----	---	----	----	----	---	---	----	----	---

Gambar 4.20 Parent mutasi 1

C6	1	3	29	33	5	22	9	11	14	21	7	6	44	12	2
----	---	---	----	----	---	----	---	----	----	----	---	---	----	----	---

Gambar 4.21 Hasil child 1

P5	17	4	1	7	15	19	3	26	29	3	11	9	25	7	38
----	----	---	---	---	----	----	---	----	----	---	----	---	----	---	----

Gambar 4.22 Parent mutasi 2

C7	17	4	1	7	11	19	3	26	29	3	15	9	25	7	38
----	----	---	---	---	----	----	---	----	----	---	----	---	----	---	----

Gambar 4.23 Hasil child 2

4.3.7 Perhitungan Fitnes

Sesuai contoh kasus sebelumnya, hasil perhitungan fitnes menggunakan persamaan 4.2 yang sebelumnya dihitung terlebih dahulu total penalti sesuai persamaan 4.1 mendapatkan hasil seperti pada Tabel 4.6 berikut.

Tabel 4.6 Hasil Perhitungan Fitnes

Hasil	Total kandungan				Total pinalti	Total harga	fitnes
	Energi	Karbohidrat	Protein	Lemak			
P1	2656.93	154.57	336.76	28	1070.784	28650	0.000349
P2	2701.09	415.39	100.44	37.5	802.902	36450	0.000274
P3	2543.73	310.44	94.44	40.5	539.235	33913	0.000295
P4	2661.099	422.85	87.99	23.31	772.103	35950	0.000278
P5	1807.143	272.490	84.80	25	627.321	29650	0.000337
P6	2630.958	411.640	101.33	27	740.399	32650	0.000306
P7	2042.436	234.987	68.66	43	395.388	23980	0.000417

Hasil	Total kandungan				Total pinalti	Total harga	fitnes
	Energi	Karbohidrat	Protein	Lemak			
P8	2381.544	342.813	79.62	20.31	407.135	24530	0.000408
P9	2599.154	396.444	102.80	41	680.865	26700	0.000374
P10	2549.101	414.278	94.78	39	642.623	32500	0.000308
C1	2436.988	364.271	81.05	29.81	475.965	32050	0.000312
C2	2266.548	293.571	75.60	36.5	198.933	26980	0.000371
C3	2415.180	381.590	86.37	35	471.608	26350	0.000379
C4	2840.907	397.765	106.80	34	934.938	33800	0.000296
C5	2280.212	337.389	94.89	43.5	205.995	30350	0.000329
C6	2738.200	405.672	100.44	37.5	830.283	35650	0.000280
C7	1938.161	301.908	87.37	25	469.456	29350	0.000341

4.3.8 Seleksi

Sesuai dengan contoh kasus sebelumnya, setelah mendapatkan hasil evaluasi dan melakukan seleksi dengan metode *elitism* yakni memilih individu sejumlah populasi awal yang memiliki fitness tertinggi. Hasil dari seleksi *elitism* dalam dilihat pada Tabel 4.7

Tabel 4.7 Hasil Seleksi

individu	Kromosom															Fitnes
	Makan Pagi					Makan siang					Makan malam					
	SP	SN	SH	SS	P	SP	SN	SH	SS	P	SP	SN	SH	SS	P	
P7	20	9	45	18	14	4	6	43	27	25	5	11	34	23	38	0.000417
P8	9	10	29	3	38	13	4	15	16	1	20	12	22	17	9	0.000408
C3	7	4	35	20	25	19	9	21	8	30	15	1	48	25	20	0.000379
C2	20	9	45	18	14	4	6	35	17	31	6	4	23	14	22	0.000371
P1	7	4	35	20	25	19	9	21	8	30	15	7	13	8	3	0.000349
C7	17	4	1	7	11	19	3	26	29	3	15	9	25	7	38	0.000341
P5	17	4	1	7	15	19	3	26	29	3	11	9	25	7	38	0.000337
C5	22	1	14	19	10	7	7	48	21	4	22	5	8	2	21	0.000329
C1	15	12	5	35	15	2	10	43	27	25	5	11	34	23	38	0.000312
P10	3	11	14	19	10	7	7	48	21	4	22	5	8	2	21	0.000308

4.3.9 Memilih Individu Terbaik

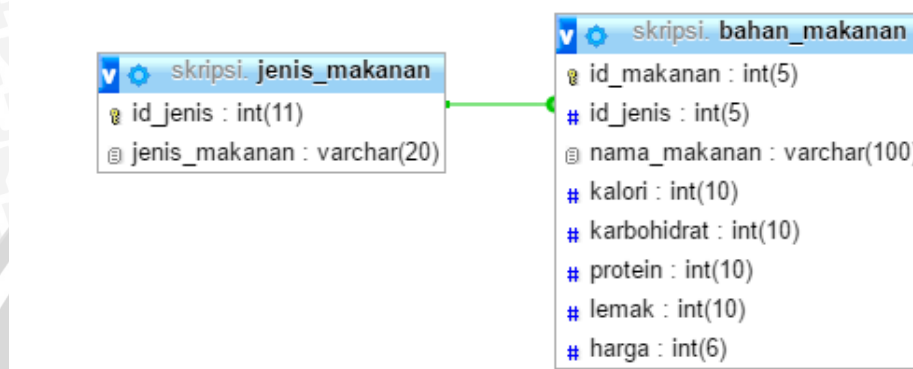
Dari hasil seleksi dapat dilihat bahwa individu terbaik merupakan individu P7 yang memiliki nilai fitness 0.000417. Individu P7 terdiri dari gen yang mewakili bahan makanan pada Tabel 4.8 berikut

Tabel 4.8 Individu Terbaik

Individu	Bahan makanan	Energi (Kkal)	Karbohidrat (gram)	Protein (gram)	Lemak (gram)	Harga (Rp)
Makan pagi	Tape Beras Ketan	262.5	60	6	0	480
	Oncom	80	8	6	3	400
	Sardencis	342.86	0	7	13	2400
	Genjer	50	10	2	0	1200
	Jeruk manis	50	10	0	0	1000
Makan siang	Jagung Segar	210	48	4.8	0	900
	Kacang Toto	200	20	15	7.5	1500
	Ayam dengan kulit	300	0	7	13	2400
	Nangka muda	100	20	6	0	1400
	Nanas	58.82	11.76	0	0	700
Makan malam	Kentang	125	28.57	2.86	0	1800
	Tahu	40	4	3	1.5	300
	Daging kambing	150	0	7	5	6400
	Pepaya muda	50	10	2	0	600
	Strawberry	23.25581	4.651163	0	0	2500

4.4 Perancangan Database

Database merupakan media yang digunakan untuk menyimpan data-data berukuran besar yang akan diolah agar nantinya dalam berjalannya proses perhitungan meringankan kerja sistem serta proses dapat menghasilkan solusi yang tepat dan cepat. Dalam penelitian ini, database digunakan untuk menyimpan seluruh data bahan makanan. *Database* yang akan digunakan terdiri dari 1 tabel yang digunakan menyimpan semua sumber makanan seperti sumber makanan pokok, sumber sayuran, sumber hewani, sumber nabati dan pelengkap. *Database* dapat dilihat pada Gambar 4.24 berikut.



Gambar 4.24 Perancangan Database

Penjelasan masing-masing tabel data bahan makanan dapat dilihat pada Tabel 4.9 dan 4.10 berikut.

Tabel 4.9 Atribut jenis_makanan

Nama atribut	Tipe data	Keterangan
Id_jenis	<i>Integer</i>	<i>Primari key</i> tabel
jenis_makanan	<i>Varchar</i>	Menyimpan jenis makanan

Tabel 4.10 Atribut bahan_makanan

Nama atribut	Tipe data	Keterangan
Id_makanan	<i>Integer</i>	<i>Primari key</i> tabel
nama_makanan	<i>Varchar</i>	Menyimpan nama makanan
jenis_makanan	<i>Varchar</i>	Menyimpan jenis makanan
Kalori	<i>Double</i>	Menyimpan nilai kalori
Karbohidrat	<i>Double</i>	Menyimpan nilai karbohidrat
Protein	<i>Double</i>	Menyimpan nilai protein
Lemak	<i>Double</i>	Menyimpan nilai lemak
harga	<i>Integer</i>	Menyimpan nilai harga

4.5 Perancangan UI

Perancangan *User Interface (UI)* pada permasalahan optimasi komposisi makanan pada penderita diabetes dan komplikasinya terdiri dari beberapa halaman yaitu halaman input, halaman data bahan makanan, halaman proses, halaman hasil.

4.5.1 Data Input

Data input merupakan halaman utama sistem karena halaman ini meminta pasien untuk melengkapi data-data yang diperlukan untuk melakukan proses selanjutnya. Halaman data input dapat dilihat pada Gambar 4.25

Optimasi Komposisi Makanan pada Penderita Diabetes Melitus dan Komplikasinya
Menggunakan Algoritma Genetika

Data Input | Data Makanan | Algoritma Genetika | Individu Terbaik | Hasil

Data Input

Data Pasien

Nama: 1

Umur: 2

Tinggi Badan: 3

Berat Badan: 4

Jenis Kelamin: 5

Aktifitas: 6

Komplikasi: ☐ 7 ☐

Generasi: 8

Popsize: 9

Crossover rate: 10

Mutation rate: 11

12

Maryamah 135150201111008 TIF-13

Gambar 4.25 Halaman Data Input

Keterangan:

1. No 1 merupakan *field* mengisi nama pasien
2. No 2 merupakan *field* mengisi umur pasien
3. No 3 merupakan *field* untuk mengisi tinggi badan pasien
4. No 4 merupakan *field* untuk mengisi berat badan pasien
5. No 5 merupakan *field* untuk memilih jenis kelamin
6. No 6 merupakan *field* untuk memilih aktifitas yang dilakukan pasien

7. No 7 merupakan *field* untuk memilih komplikasi yang diderita pasien
8. No 8 merupakan *field* untuk mengisi jumlah generasi
9. No 9 merupakan *field* untuk mengisi *popsize*
10. No 10 merupakan *field* untuk mengisi *crossover rate*
11. No 11 merupakan *field* untuk mengisi *mutation rate*
12. No 12 merupakan tombol sistem melakukan prosesnya

4.5.2 Data Makanan

Pada halaman data makanan, sistem menampilkan data – data bahan makanan yang disimpan dalam database dan dipanggil sesuai dengan jenis sumber data makanan. Data – data makanan terdiri dari sumber makanan pokok, sumber nabati, sumber hewani, sumber sayuran, dan pelengkap. Halaman data makanan dapat dilihat pada Gambar 4.26

Optimasi Komposisi Makanan pada Penderita Diabetes Mellitus dan Komplikasinya
Menggunakan Algoritma Genetika

Data Input
Data Makanan
Algoritma Genetika
Individu Terbaik
Hasil

Data Bahan Makanan

Pilih Data 1

2

Title 1	Title 2	Title 3	Title 4	Title 5	Title 6	Title 7	Title 8

Maryamah
135150201111008
TIF-13

Gambar 4.26 Halaman Data Makanan

Keterangan:

1. No 1 merupakan *combo box* untuk memilih data makanan
2. No 2 merupakan tabel untuk menampilkan bahan makanan

[illegible]

1. No 1 merupakan tabel menampilkan aktifitas algoritma genetika setiap generasi

4.5.4 Individu Terbaik

Halaman individu terbaik menampilkan hasil generasi terbaik dari proses sebelumnya. Halaman individu terbaik dapat dilihat pada Gambar 4.28 berikut.

Optimasi Komposisi Makanan pada Penderita Diabetes Melitus dan Komplikasinya
Menggunakan Algoritma Genetika

Data Input Data Makanan Algoritma Genetika Individu Terbaik Hasil

1

Kromosom	Fitness

Maryamah 135150201111008 TIF-13

Gambar 4.28 Halaman individu terbaik

Keterangan:

1. No 1 merupakan tabel menampilkan individu terbaik

4.5.5 Hasil

Halaman hasil menampilkan beberapa data hasil dari proses perhitungan algoritma genetika serta beberapa data lain. Data lain yang ditampilkan adalah data pasien, data kebutuhan gizi pasien deskripsi bahan makanan serta detail bahan makanan optimal dalam satu hari yang harus menjadi menu pagi, siang dan malam bagi pasien. Halaman hasil dapat dilihat pada Gambar 4.29 berikut.

Optimasi Komposisi Makanan pada Penderita Diabetes Melitus dan Komplikasinya
Menggunakan Algoritma Genetika

Data Input Data Makanan Individu Terbaik Hasil

Data Pasien

Nama

Umur

Tinggi Badan

Berat Badan 1

Jenis Kelamin

Aktifitas

Komplikasi 3

Kebutuhan Gizi

Energi

Karbohidrat 2

Protein

Lemak

4

Title 1	Title 2	Title 3	Title 4	Title 5	Title 6	Title 7	Title 8	Title 9

Maryamah

135150201111008

TIF-13

Gambar 4.29 Halaman Hasil

Keterangan:

1. No 1 merupakan *field* menampilkan data pasien
2. No 2 merupakan *field* menampilkan kebutuhan gizi pasien
3. No 3 merupakan deskripsi hasil komposisi makanan
4. No 4 merupakan tabel menampilkan komposisi makanan

4.6 Perancangan Uji Coba dan Evaluasi

Untuk mendapatkan hasil solusi yang optimal, diperlukan perancangan uji coba pada beberapa hal berikut:

1. Pengujian untuk menentukan ukuran individu dalam populasi yang paling optimal
2. Pengujian untuk menentukan kombinasi *crossover rate* dan *mutation rate* dalam proses reproduksi yang paling optimal
3. Pengujian untuk menentukan ukuran generasi yang paling optimal
4. Pengujian Kromosom untuk menentukan apakah kromosom yang dirancang sudah sesuai atau tidak.

4.6.2 Pengujian Ukuran Individu Populasi

Pengujian individu populasi digunakan untuk mendapatkan ukuran individu dalam populasi yang paling optimal. Hal ini dilakukan dengan cara memasukkan beberapa jumlah populasi sehingga nantinya akan diketahui berapa individu paling optimal dalam satu generasi. Pada pengujian ukuran individu dalam populasi nilai *crossover rate* dan *mutation rate* dibuat sama yaitu $cr=0.5$ $mr=0.5$ dan generasi=30. Dengan perlakuan tersebut nantinya individu dalam populasi akan menghasilkan fitness dan dibandingkan ukuran populasi berapa yang memiliki fitness lebih besar. Tabel rancangan pengujian genetika dapat dilihat pada tabel 4.11 berikut.

Tabel 4.11 Rancangan Pengujian Ukuran Populasi

Popsize	Nilai Fitnes					Nilai fitnes rata-rata
	Percobaan ke					
	1	2	3	4	5	
10						
40						
70						
100						
130						
160						
190						
220						
250						
280						
310						
340						

4.6.3 Pengujian Kombinasi Crossover Rate dan Mutation Rate

Pengujian kombinasi *crossover rate* (cr) dan *mutation rate* (mr) dilakukan untuk mengetahui kombinasi cr dan mr yang paling optimal. Pada pengujian kombinasi cr dan mr , ukuran generasi dan ukuran individu dalam populasi yang digunakan sama. Untuk pengujian ukuran generasi dilakukan pada generasi berjumlah tetap sesuai pengujian populasi sebelumnya dan ukuran populasi menggunakan jumlah individu yang memiliki rata-rata fitness terbaik. Tabel rancangan kombinasi *crossover rate* dan *mutation rate* dapat dilihat pada 4.12 berikut.

Tabel 4.12 Rancangan Pengujian Crossover Rate (Cr) Dan Mutation Rate (Mr)

No	Kombinasi		Nilai Fitnes					Nilai fitnes rata- rata
			Percobaan ke					
	Cr	Mr	1	2	3	4	5	
1	1	0						
2	0.9	0.1						
3	0.8	0.2						
4	0.7	0.3						
5	0.6	0.4						
6	0.5	0.5						
7	0.4	0.6						
8	0.3	0.7						
9	0.2	0.8						
10	0.1	0.9						
11	0	1						

4.6.4 Pengujian Generasi

Pengujian generasi dilakukan untuk mendapatkan ukuran generasi yang optimal. Hal ini dilakukan dengan menguji coba beberapa ukuran generasi yang menghasilkan solusi yang optimal. Pengujian generasi dilakukan karena sifat algoritma genetika yang memiliki proses awal memilih individu, kromosom, dan gen didalamnya dengan melakukan random sehingga hipotesa yang didapatkan adalah semakin besar ukuran genetika maka semakin optimal solusi yang dihasilkan. Pengujian generasi terakhir setelah pengujian populasi dan pengujian kombinasi cr dan mr dilakukan. Pengujian generasi dilakukan dengan 20 ukuran generasi sesuai pada Tabel 4.13. Dengan perlakuan ukuran generasi berbeda, jumlah populasi, crossover rate dan mutation rate yang diinputkan dalam hasil sebelumnya jumlah populasi, kombinasi cr dan mr yang paling optimal. Dengan perlakuan tersebut nantinya tiap generasi akan menghasilkan masing-masing fitnes dan dibandingkan generasi berapa yang memiliki fitnes lebih besar. Tabel rancangan pengujian genetika dapat dilihat pada tabel 4.13 berikut.

Tabel 4.13 Rancangan Pengujian Generasi

Ukuran iterasi	Nilai fitness					Nilai fitness rata-rata
	Percobaan ke					
	1	2	3	4	5	
10						
25						
40						
55						
70						
85						
100						
115						
130						
145						
160						
175						
190						
205						
220						
235						
250						
265						
280						
295						

4.6.5 Pengujian Kromosom

Pengujian kromosom merupakan pengujian yang dilakukan untuk menguji kromosom yang sudah dirancang sesuai atau tidak dengan permasalahan yang diteliti. Kromosom dikatakan sesuai jika jumlah gen dalam kromosom berjumlah 15 dan masing-masing nilai gen tidak melebihi batas dari jumlah makanan.

BAB 5 IMPLEMENTASI

5.1 Struktur Kelas

Struktur kelas pada permasalahan optimasi komposisi makanan pada penderita penyakit diabetes dan komplikasinya menggunakan algoritma genetika terdiri dari dua kelas yaitu

1. Koneksi. Koneksi digunakan untuk menghubungkan sistem dengan *database* sehingga dalam proses algoritma genetika lebih tepatnya ketika menghitung total kalori sesuai representasi kromosom dapat mengambil nilai yang berasal dari *database*.
2. Proses. Proses digunakan untuk menyimpan semua metode-metode dalam penyelesaian masalah menggunakan algoritma genetika.

5.2 Source Code

Sesuai tahapan algoritma genetika terdapat beberapa tahapan penting dalam penyelesaian permasalahan. Tahapan-tahapan tersebut adalah representasi kromosom, *crossover* dan mutasi, perhitungan fitness dan seleksi. Tahapan tersebut diimplementasikan pada subbab *source code* dibawah ini

5.2.1 Representasi Kromosom

Source code representasi kromosom berfungsi dalam pembuatan kromosom sesuai dengan jumlah populasi yang diinputkan dan jumlah gen 15. Gen berisi makanan pokok, nabati, hewani, sayuran, pelengkap dan masing-masing gen dilakukan random sesuai jumlah bahan makanan yang tersedia pada *database*. *Source code* representasi kromosom dapat dilihat pada *Source code 1*

```
int maxSP = 22, maxN = 12, maxH = 48, maxS = 36, maxP = 38, min = 1;
for (int j = 0; j < jumlahGen; j++) {
    if (j == 0 || j == 5 || j == 10) {
        individu[a][j] = rand.nextInt(maxSP) + min;
    } else if (j == 1 || j == 6 || j == 11) {
        individu[a][j] = rand.nextInt(maxN) + min;
    } else if (j == 2 || j == 7 || j == 12) {
        individu[a][j] = rand.nextInt(maxH) + min;
    } else if (j == 3 || j == 8 || j == 13) {
        individu[a][j] = rand.nextInt(maxS) + min;
    } else if (j == 4 || j == 9 || j == 14) {
        individu[a][j] = rand.nextInt(maxP) + min;
    }
    populasi[0][a][j] = individu[a][j]; } }
```

Source Code 1 Representasi Kromosom

5.2.2 Crossover

Crossover merupakan bagian dari reproduksi yang berfungsi menghasilkan anak. Pada proses *crossover* menggunakan metode *one cut point* dimana memilih satu titik potong dan hasil akan dimasukkan ke anak. *Source code crossover* dapat dilihat pada *Source code 2*

```
public void maxIterasiCrossover() {
    //      untuk menentukan berapa banyak iterasi crossovernya
    for (int x = 1; x <= popsize; x++) {
        if ((2 * x) > maxChildCrossover && (2 * x) - maxChildCrossover
        == 2) {
            maxIterasi = x - 1;
            break;
        } else if ((2 * x) > maxChildCrossover && (2 * x) -
        maxChildCrossover == 1) {
            maxIterasi = x;
            break;
        }
    }
    crossoverParent = new
    int[this.generasi][maxIterasi][2][jumlahGen];
    pointerCrossover = new int[this.generasi][maxIterasi];
}

public void crossover(int maxchild, int generasi) { //one cut point
    int parent1, parent2, pointer;
    int child[][][] = new int[maxIterasi][2][jumlahGen];
    int counter_child = 0;
    for (int iterasi = 0; iterasi < maxIterasi; iterasi++) {
        parent1 = rand.nextInt((popsize - 1));
        do {
            parent2 = rand.nextInt((popsize - 1));
        } while (parent2 == parent1);

        pointer = rand.nextInt((jumlahGen - 1) + 1);

        //      setting nilai child seperti nilai parent
        for (int y = 0; y < jumlahGen; y++) {
            child[iterasi][0][y] = getIndividuKe(parent1, y);
            child[iterasi][1][y] = getIndividuKe(parent2, y);
            crossoverParent[generasi][iterasi][0][y] =
            getIndividuKe(parent1, y);
            crossoverParent[generasi][iterasi][1][y] =
            getIndividuKe(parent2, y);
        }
    }
}
```



```

    }

    pointerCrossover[generasi][iterasi] = pointer + 1;
    //proses crossover
    for (int y = pointer; y < jumlahGen; y++) {
        int temp = child[iterasi][0][y];
        child[iterasi][0][y] = child[iterasi][1][y];
        child[iterasi][1][y] = temp;
    }
    for (int x = 0; x < 2; x++) {
        for (int y = 0; y < jumlahGen; y++) {
        }
        counter_child++;
    }
}

//    memilih child mana yang masuk sebagai populasi
int childKe = 0, crosKe = 0;
if (maxChildCrossover % 2 == 0) {
    int x = popsize;
    do {
        for (int iterasi = 0; iterasi < maxIterasi; iterasi++) {
            for (int y = 0; y < jumlahGen; y++) {
                populasi[generasi][x][y] = child[iterasi][0][y];
            }
            x++;
            for (int y = 0; y < jumlahGen; y++) {
                populasi[generasi][x][y] = child[iterasi][1][y];
            }
            x++;
        }
    } while (x < maxchild);
} else {
//    jika max child crossover tidak kelipatan 2 maka
    int x = popsize;
    do {
        for (int iterasi = 0; iterasi < maxIterasi - 1; iterasi++){
            for (int y = 0; y < jumlahGen; y++) {
                populasi[generasi][x][y] = child[iterasi][0][y];
            }
            x++;
            for (int y = 0; y < jumlahGen; y++) {

```

```

        populasi[generasi][x][y] = child[iterasi][1][y];
    }
    x++;
}
//      slot populasi tersisa 1 maka diisi dengan 'hasil child pertama
pada iterasi crossover yang terakhir'
    for (int y = 0; y < jumlahGen; y++) {
        populasi[generasi][popsize + maxChildCrossover - 1][y] =
child[maxIterasi-1][0][y];
    }
} while (x < maxchild);
}

```

Source Code 2 Crossover

5.2.3 Mutasi

Hampir sama dengan *crossover*, mutasi juga merupakan bagian dari reproduksi yang berfungsi menghasilkan *child*. Pada proses mutasi menggunakan metode *reciprocal exchange mutation* dimana memilih dua titik, masing-masing nilai ditukar dan hasil akan dimasukkan ke *child*. *Source code* mutasi dapat dilihat pada *Source code 3*.

```

int parent, pointer1, pointer2;

maxChildCrossover = (int) Math.ceil(cr * popsize);
maxChildMutation = (int) Math.ceil(mr * popsize);
mutasiTerpilih = new int[generasi][maxChildMutation][jumlahGen];
int child[][] = new int[maxChildMutation][jumlahGen];
for (int iterasi = 0; iterasi < maxChild; iterasi++) {
    parent = rand.nextInt((popsize - 1));
    pointer1 = rand.nextInt((jumlahGen - 1));
    if (pointer1 == 0 || pointer1 == 5 || pointer1 == 10) {
        do {
            pointer2 = rand.nextInt((jumlahGen - 1));
        } while ((pointer2 != 0 && pointer2 != 5 && pointer2 !=
10) || pointer1 == pointer2);
    }
    if (pointer1 == 1 || pointer1 == 6 || pointer1 == 11) {
        do {
            pointer2 = rand.nextInt((jumlahGen - 1));
        } while ((pointer2 != 1 && pointer2 != 6 && pointer2 !=
11) || pointer1 == pointer2);
    }
    if (pointer1 == 2 || pointer1 == 7 || pointer1 == 12) {
        do {
            pointer2 = rand.nextInt((jumlahGen - 1));

```

```

        } while ((pointer2 != 2 && pointer2 != 7 && pointer2 !=
12) || pointer1 == pointer2);
    }
    if (pointer1 == 3 || pointer1 == 8 || pointer1 == 13) {
        do {
            pointer2 = rand.nextInt((jumlahGen - 1));
        } while ((pointer2 != 3 && pointer2 != 8 && pointer2 !=
13) || pointer1 == pointer2);
    }
    if (pointer1 == 4 || pointer1 == 9 || pointer1 == 14) {
        do {
            pointer2 = rand.nextInt((jumlahGen - 1));
        } while ((pointer2 != 4 && pointer2 != 9 && pointer2 !=
14) || pointer1 == pointer2);
    }
    for (int y = 0; y < jumlahGen; y++) {
        child[iterasi][y] = getIndividuKe(parent, y);
    }
    //proses mutation
    int temp = child[iterasi][pointer1];
    child[iterasi][pointer1] = child[iterasi][pointer2];
    child[iterasi][pointer2] = temp;
//    child mutasi telah terbentuk
//    jadikan child mutasi sebagai populasi
    int maxChildC = 0;
    for (int x = popsize + maxChildCrossover; x < popsize +
maxChildCrossover + maxChildMutation; x++) {
        for (int y = 0; y < jumlahGen; y++) {
            populasi[generasi][x][y] = child[maxChildC][y];
            //child[maxChildC][y]
        }
        maxChildC++;
    }
}

```

Source Code 3 Mutasi

5.2.4 Perhitungan Fitnes

Sebelum perhitungan fitnes, wajib mengitung penalti dari kebutuhan karbohidrat, protein, dan lemak. Setelah itu dijumlah total pinalti dan dimasukkan ke perhitungan fitnes. Dalam perhitungan fitnes juga membutuhkan total harga sesuai kromosom yang terbentuk. Setelah total penalti dan total harga didapatkan perhitungan fitnes dapat dilakukan. Adapaun source code perhitungan fitnes dapat dilihat pada *Source code 4*.

```

//menghitung penalti
public void penalti(int generasi) {

```



```

for (int pop = 0; pop < populasi[0].length; pop++) {
    double penaltiKalori = Math.abs(total - totalKalori[pop]);
    if (penaltiKalori <= 100) {
        penaltiKalori = 0;
    }
    double penaltiKarbohidrat = Math.abs(karbohidratSeimbanggram - totalKarbohidrat[pop]);
    if (penaltiKarbohidrat <= 12.5) {
        penaltiKarbohidrat = 0;
    }
    double penaltiProtein = Math.abs(proteinSeimbanggram - totalProtein[pop]);
    if (penaltiProtein <= 5.55) {
        penaltiProtein = 0;
    }
    double penaltiLemak = Math.abs(lemakSeimbanggram - totalLemak[pop]);
    if (penaltiLemak < 12.5) {
        penaltiLemak = 0;
    }
    if (k_ginjal.isSelected() && penaltiProtein > 0) {
        penaltiProtein = penaltiProtein * 3;
    }
    if (k_jantung.isSelected() && penaltiLemak > 0 || k_stroke.isSelected() && penaltiLemak > 0) {
        penaltiLemak = penaltiLemak * 3;
    }
    if (penaltiKalori > 0) {
        penaltiKalori = penaltiKalori * 3;
    }
    totalPenalti[pop] = Math.abs(penaltiKalori + penaltiKarbohidrat + penaltiProtein + penaltiLemak);
}

//menghitung fitnes
public void fitnes(int generasi) {
    for (int i = 0; i < populasi[0].length; i++) {
        double semua = (totalHarga[i] + (totalPenalti[i] * 100));
        if (semua == 0) {
            fitnes[generasi][i] = 0;
        } else {
            fitnes[generasi][i] = (1000 / semua);
        }
    }
}

```

Source Code 4 Perhitungan Fitnes

5.2.5 Seleksi

Proses seleksi yang digunakan adalah seleksi *elitism* dimana hanya fitnes terbaik saja yang terpilih. Setelah mendapatkan fitnes dari proses sebelumnya, fitnes diurutkan sesuai nilai fitnes dan dipilih sesuai jumlah *popsi*. Hasil dari seleksi akan digunakan kembali pada generasi selanjutnya. Proses seleksi dapat dilihat pada *source code* 5.

```
public int[][] randomInjection(int generasi) {
    int n = (int) Math.ceil(0.1 * popsize);
    int maxSP = 22, maxN = 12, maxH = 48, maxS = 36, maxP = 38, min = 1;
    for (int a = populasi[0].length - n, b = 0; a <
        populasi[0].length; a++, b++) {
        for (int j = 0; j < jumlahGen; j++) {
            if (j == 0 || j == 5 || j == 10) {
                randomInj[b][j] = rand.nextInt((maxSP - min) + 1) + min;
            } else if (j == 1 || j == 6 || j == 11) {
                randomInj[b][j] = rand.nextInt((maxN - min) + 1) + min;
            } else if (j == 2 || j == 7 || j == 12) {
                randomInj[b][j] = rand.nextInt((maxH - min) + 1) + min;
            } else if (j == 3 || j == 8 || j == 13) {
                randomInj[b][j] = rand.nextInt((maxS - min) + 1) + min;
            } else if (j == 4 || j == 9 || j == 14) {
                randomInj[b][j] = rand.nextInt((maxSP - min) + 1) + min;
            }
            populasi[generasi][a][j] = randomInj[b][j];
        }
    }
    return randomInj;
}

//melakukan seleksi elitism
public void seleksi(int generasi) {
    int individuBaru[][] = new int[popsi][jumlahGen];
    int n = (int) Math.ceil(0.1 * popsize);
    //    sorting nilai fitnes,penalti,harga
    double fitnesSorted[] = fitnes[generasi];
    double HargaSorted[] = totalHarga;
    double PenaltiSorted[] = totalPenalti;
    for (int iterasi = 0; iterasi < populasi[0].length - n; iterasi++) {
        for (int d = 1; d < populasi[0].length; d++) {
            if (fitnesSorted[d] > fitnesSorted[d - 1]) {
                double tempH = HargaSorted[d - 1];
```

```

        HargaSorted[d - 1] = HargaSorted[d];
        HargaSorted[d] = tempH;

        double tempP = PenaltiSorted[d - 1];
        PenaltiSorted[d - 1] = PenaltiSorted[d];
        PenaltiSorted[d] = tempP;

        double temp = fitnessSorted[d - 1];
        fitnessSorted[d - 1] = fitnessSorted[d];
        fitnessSorted[d] = temp;
        for (int e = 0; e < jumlahGen; e++) {
            int temp2 = populasi[generasi][d - 1][e];
            populasi[generasi][d-1][e] =
populasi[generasi][d][e];
            populasi[generasi][d][e] = temp2;
        }
    } else {
        fitnessSorted[d] = fitnessSorted[d];
        HargaSorted[d] = HargaSorted[d];
        PenaltiSorted[d] = PenaltiSorted[d];
    }
}
}
if ((generasi + 1) % 15) == 0) {
    for (int a = 0; a < popsize - n; a++) {
        for (int b = 0; b < jumlahGen; b++) {
            individu[a][b] = populasi[generasi][a][b];
            populasi[generasi][a][b] = individu[a][b];
        }
    }
    int IndividuInjection[][];
    IndividuInjection = randomInjection(generasi);
    for (int i = popsize - n, k = 0; i < popsize; i++, k++) {
        for (int j = 0; j < jumlahGen; j++) {
            populasi[generasi][i][j] = IndividuInjection[k][j];
        }
        totalPenalti[i] = 0;
        fitness[generasi][i] = 0;
        totalHarga[i] = 0;
    }
}

```



```

        for (int a = 0; a < popsize; a++) {
            for (int b = 0; b < jumlahGen; b++) {
                individu[a][b] = populasi[generasi][a][b];
                populasi[generasi][a][b] = individu[a][b];
            }
        }
    } else {
        // id untuk tiap individu dalam populasi
// set barang menjadi nilai dari individu yang memiliki fitness
// tertinggi
        if (generasi != this.generasi - 1) {
            for (int a = 0; a < popsize; a++) {
                for (int b = 0; b < jumlahGen; b++) {
                    individu[a][b] = populasi[generasi][a][b];
                    populasi[generasi + 1][a][b] = individu[a][b];
                }
            }
        }

        pBest[generasi] = fitnessSorted[0];
        for (int a = 0; a < jumlahGen; a++) {
            pBestKromosom[generasi][a] = populasi[generasi][0][a];
        }
        cariGBest(generasi);
    }

    public void cariGBest(int generasi) {
        if (generasi == 0) {
            gBest = pBest[0];
        } else {
            if (pBest[generasi] > gBest) {
                gBest = pBest[generasi];
            }
// update kromosom
            for (int b = 0; b < jumlahGen; b++) {
                gBestKromosom[b] = pBestKromosom[generasi][b];
            }
        }
    }
}

```

Source Code 5 Seleksi

5.3 Implementasi UI

UI sistem terdiri dari 5 tab yaitu beranda yang berisi data *input* pasien data parameter algoritma genetika. Tab kedua adalah data makanan yang terdiri dari data keseluruhan bahan makanan yang digunakan dalam menemukan solusi. Tab ketiga terdiri dari aktifitas algoritma genetika yang berisi informasi setiap individu dalam populasi setiap generasi yang diinputkan. Informasi terdiri dari kromosom, penalti, total harga, dan fitnessnya. Tab keempat terdiri dari individu terbaik atau individu yang nantinya akan dijadikan solusi yang ditampilkan kromosom dan nilai fitness. Tab kelima berisi informasi data pasien yang diinputkan serta data total kalori, karbohidrat, lemak dan protein seimbang yang harus dipenuhi oleh pasien. Selain itu terdapat susunan menu makan pagi, siang malam yang harus dikonsumsi oleh pasien dalam waktu satu hari. Implementasi ui dapat dilihat pada Gambar 5.1, Gambar 5.2, Gambar 5.3, Gambar 5.4 dan Gambar 5.5

5.3.1 Implementasi UI Beranda

Pada UI beranda terdiri dari data *input* yang dibutuhkan sehingga data-data tersebut akan diproses. Data-data tersebut dibagi menjadi dua yaitu data pasien data parameter dari algoritma genetika. Data pasien terdiri dari nama pasien, umur, tinggi badan, berat badan, jenis kelamin, aktifitas, dan penyakit komplikasi yang diderita. Data parameter algoritma genetika terdiri dari jumlah generasi, *popsi*, *crossover rate* dan *mutation rate*. UI beranda dapat dilihat pada Gambar 5.1 berikut

Optimasi Komposisi Makanan pada Penderita Diabetes Melitus dan Komplikasinya
Menggunakan Algoritma Genetika

Data Input

Data Pasien

Nama: Abdullah

Umur: 52

Tinggi Badan: 165

Berat Badan: 65

Jenis Kelamin: Laki-laki

Aktifitas: Ringan

Komplikasi

☐ Mata ☐ Stroke

☐ Saraf ☐ Jantung

☒ Ginjal ☐ Tidak Komplikasi

☐ Lambung

Generasi: 30

Popsi: 10

Crossover rate: 0.1

Mutation rate: 0.9

Hitung

Maryamah 135150201111008 TIF-13

Gambar 5.1 UI Beranda

5.3.2 Implementasi UI Data Makanan

Pada UI data makanan digunakan untuk melihat data bahan makanan yang nantinya digunakan sebagai penentuan menu makanan harian dari pasien. Terdapat 5 bahan makanan yang tersedia yaitu sumber pokok, sumber nabati, sumber hewani, sumber sayuran dan sumber pelengkap. Untuk melihat data bahan makanan, terdapat menu yang dapat dipilih sehingga nantinya akan menampilkan data makanan sesuai pilihan. Adapun UI data makanan dapat dilihat pada Gambar 5.2

Optimasi Komposisi Makanan pada Penderita Diabetes Melitus dan Komplikasinya
Menggunakan Algoritma Genetika

Data Input Data Makanan Algoritma Genetika Individu Terbaik Hasil

Data Bahan Makanan

Pilih Data Sumber Pokok

No	Jenis Ba...	Bahan M...	Kalori	Karbohi...	Protein	Lemak	Harga
1	1	Bihun	525	12	120	0	750
2	1	Biskuit	656	15	150	0	900
3	1	Havermut	583	13	133	0	1200
4	1	Jagung	210	5	48	0	900
5	1	Kentang	125	3	29	0	1100
6	1	Kentang ...	525	5	48	0	1050
7	1	Makaroni	525	12	120	0	700
8	1	Mie Basah	131	3	30	0	800
9	1	Mie Kering	525	12	120	0	850
10	1	Beras Gi...	263	6	60	0	1800
11	1	Beras Gi...	263	6	60	0	2250
12	1	Beras Gi...	263	6	60	0	1500
13	1	Beras 1/...	263	6	60	0	1600
14	1	Nasi Ket...	263	6	60	0	1300
15	1	Roti Putih	375	9	86	0	1750
16	1	Roti War...	375	9	86	0	1500
17	1	Singkong	219	5	50	0	900

Maryamah 135150201111008 TIF-13

Gambar 5.2 UI Data Makanan

5.3.3 Implementasi UI Algoritma Genetika

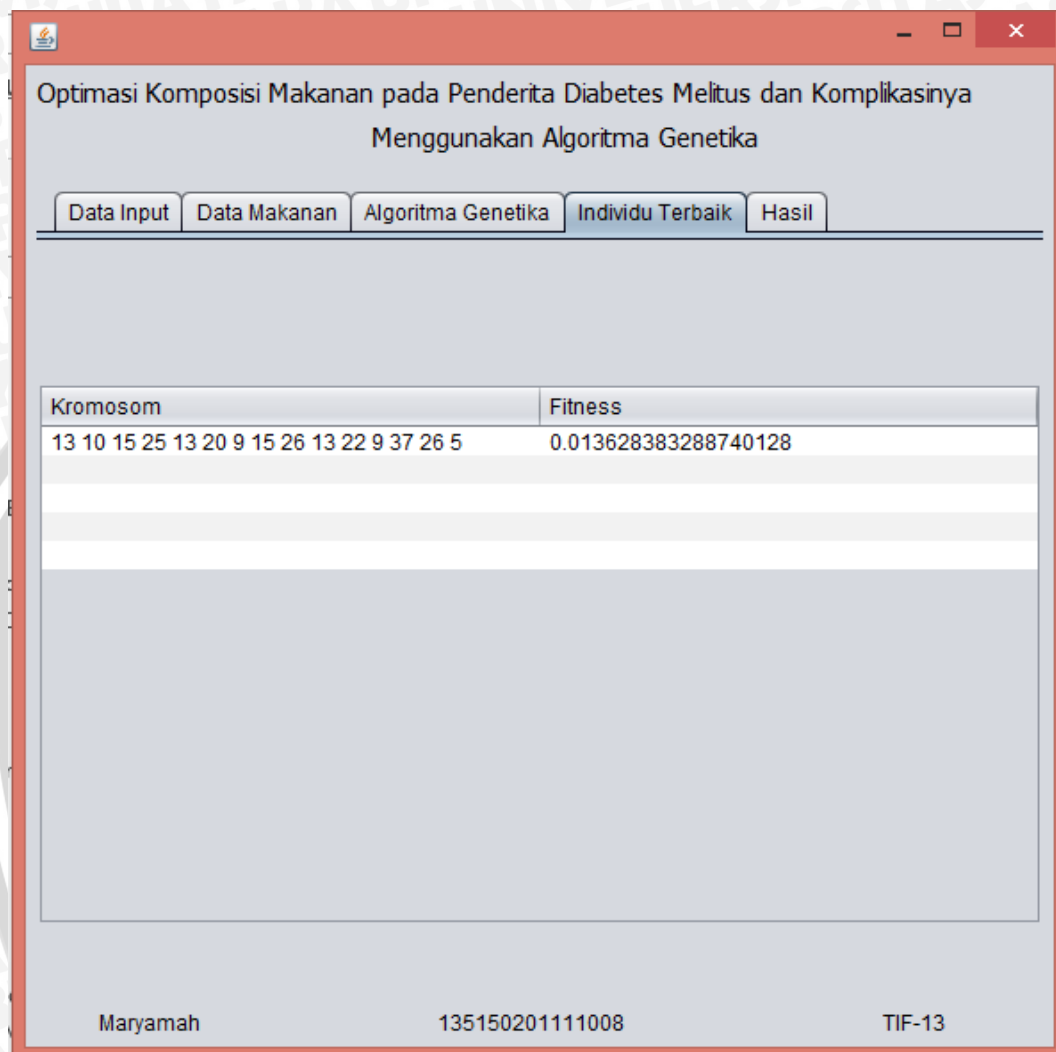
UI algoritma genetika berisi aktifitas dari individu dalam populasi. Diperlihatkan informasi semua individu dalam generasi yang diinputkan sebelumnya. Informasi- informasi tersebut adalah kromosom dari individu, total penalti, total harga dan fitness. Urutan dari individu tersebut sesuai dari besar fitness yang dimiliki. UI algoritma genetika dapat dilihat pada Gambar 5.3

Generasi	Individu	Kromosom	Penalti	Total Harga	Fitness
1	individu ke-1	13 9 38 1 4 2...	1354.21275...	19560.0	0.00645239...
1	individu ke-2	13 11 38 1 4 ...	1358.21275...	19560.0	0.00643578...
1	individu ke-3	16 11 26 35 ...	1317.71275...	32056.0	0.00610398...
1	individu ke-4	16 11 26 35 ...	1317.71275...	32056.0	0.00610398...
1	individu ke-5	16 11 26 4 1...	1317.71275...	32056.0	0.00610398...
1	individu ke-6	18 8 22 32 5 ...	1555.71275...	22260.0	0.00562330...
1	individu ke-7	20 11 20 34 ...	1545.71275...	25613.0	0.00554987...
1	individu ke-8	20 11 20 24 ...	1545.71275...	25613.0	0.00554987...
1	individu ke-9	14 11 15 11 ...	1708.71275...	26040.0	0.00507842...
1	individu ke-10	14 3 15 11 1...	1708.71275...	26040.0	0.00507842...
2	individu ke-1	13 9 15 1 4 2...	1351.21275...	19560.0	0.00646490...
2	individu ke-2	13 9 38 1 4 2...	1354.21275...	19560.0	0.00645239...
2	individu ke-3	13 9 38 1 4 2...	1354.21275...	19560.0	0.00645239...
2	individu ke-4	13 11 38 1 4 ...	1358.21275...	19560.0	0.00643578...
2	individu ke-5	22 11 38 1 4 ...	1358.21275...	19560.0	0.00643578...
2	individu ke-6	13 11 38 1 4 ...	1368.21275...	19560.0	0.00639462...
2	individu ke-7	16 5 26 4 18 ...	1303.71275...	32056.0	0.00615660...
2	individu ke-8	16 11 26 35 ...	1314.71275...	32056.0	0.00611518...
2	individu ke-9	16 11 26 35 ...	1317.71275...	32056.0	0.00610398...
2	individu ke-10	16 11 26 35 ...	1317.71275...	32056.0	0.00610398...

Gambar 5.3 UI Algoritma Genetika

5.3.4 Implementasi UI Individu Terbaik

UI individu terbaik menampilkan individu yang terbaik dari semua generasi. Individu tersebut ditampilkan kromosom yang dimiliki dan nilai fitnessnya. Adapun UI individu terbaik dapat dilihat pada Gambar 5.4



Gambar 5.4 UI Individu Terbaik

5.3.5 Implementasi UI Hasil

Pada UI hasil berisi data pasien yang telah diinputkan dan total kalori, karbohidrat, protein dan lemak seimbang yang dibutuhkan bagi pasien. Selain itu menampilkan menu dari solusi terbaik dari proses algoritma genetika yang terdiri dari makan pagi, siang dan malam dengan masing-masing memiliki 5 makanan dan harga makanan. Adapun UI hasil dapat dilihat pada Gambar 5.5

Optimasi Komposisi Makanan pada Penderita Diabetes Melitus dan Komplikasinya
Menggunakan Algoritma Genetika

Data Input | **Data Makanan** | **Algoritma Genetika** | **Individu Terbaik** | **Hasil**

Data Pasien

Nama	Abdullah
Umur	52
Tinggi Badan	165
Berat Badan	65
Jenis Kelamin	Laki-laki
Aktivitas	Ringan
Komplikasi	Ginjal

Kebutuhan Gizi

Energi	2193.75
Karbohidrat	318.09375
Protein	29.22075
Lemak	164.53125

Nama menu	Kalori	Karbohidrat	Lemak	Protein	Biaya
Jeruk garut	43	9	0	0	1000
Malam					
Ubi Jalar Ku...	194	4	44	0	750
Oncom	80	8	6	3	400
Telur bebek ...	120	0	7	5	4000
Mangkakan	100	20	6	0	1000
Belimbing	40	8	0	0	1000
Total	2160.0	138.0	230.0	23.0	19780.0

Maryamah 135150201111008 TIF-13

Gambar 5.5 UI Hasil

BAB 6 PENGUJIAN

Dalam pengujian dilakukan dengan menggunakan 4 metode pengujian yaitu pengujian populasi, pengujian *crossover rate* dan *mutation rate*, pengujian generasi dan pengujian kromosom. Tahap pengujian yang dilakukan yaitu pertama melakukan pengujian populasi untuk mengetahui jumlah populasi optimal, setelah mendapatkan populasi optimal melakukan pengujian kombinasi cr dan mr. Setelah mendapatkan kombinasi cr dan mr yang optimal, dilakukan pengujian generasi. Pengujian terakhir yang dilakukan adalah pengujian kromosom.

6.1 Pengujian Populasi

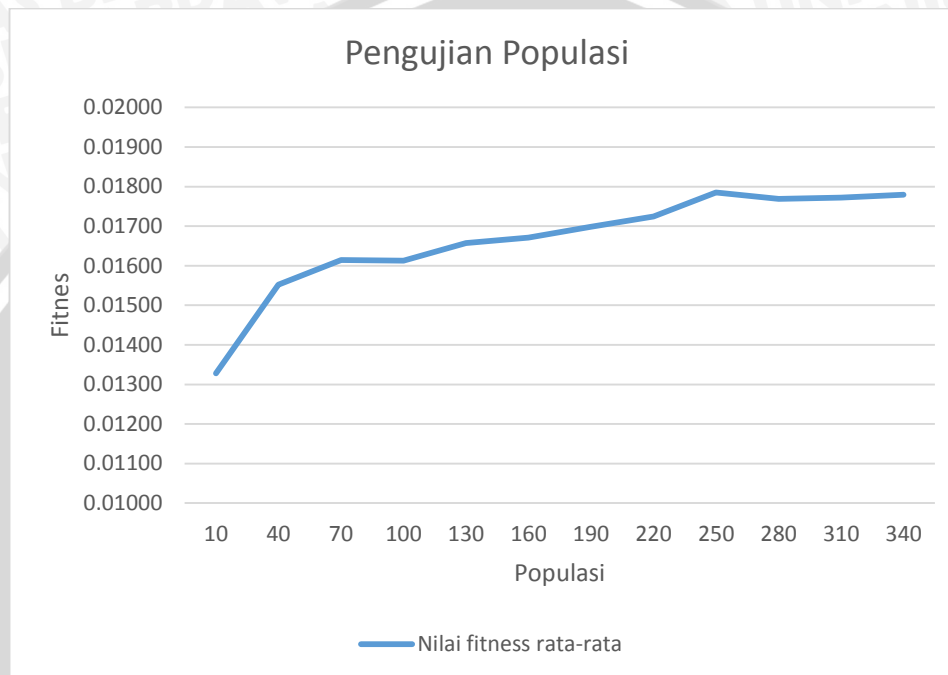
Pengujian populasi digunakan untuk mencari populasi yang optimal. Pengujian populasi dilakukan dengan lima kali percobaan dengan 5 ukuran populasi yang berbeda lalu dihitung rata-rata fitness yang didapat. Pada pengujian populasi dilakukan dengan mengimpuln generasi sebesar 30, *crossover rate* = 0.5 dan *mutation rate* = 0.5. Selain itu data pasien menggunakan nama samaran, tinggi badan 180 cm, berat badan 60 cm, umur 50 tahun, aktifitas ringan, dan komplikasi yang diderita pasien adalah ginjal. Hasil pengujian populasi dapat dilihat pada tabel 6.1 berikut.

Tabel 6.1 Pengujian Populasi

Popsize	Nilai Fitnes					Nilai fitnes rata-rata
	Percobaan ke					
	1	2	3	4	5	
10	0.01189	0.01260	0.01412	0.01276	0.01502	0.01328
40	0.01443	0.01564	0.01606	0.01622	0.01527	0.01552
70	0.01611	0.01737	0.01574	0.01598	0.01550	0.01614
100	0.01525	0.01562	0.01705	0.01571	0.01701	0.01613
130	0.01641	0.01625	0.01793	0.01613	0.01615	0.01658
160	0.01646	0.01760	0.01705	0.01640	0.01606	0.01671
190	0.01649	0.01598	0.01723	0.01761	0.01762	0.01699
220	0.01747	0.01783	0.01767	0.01723	0.01603	0.01724
250	0.01774	0.01799	0.01774	0.01791	0.01786	0.01785
280	0.01777	0.01762	0.01763	0.01742	0.01800	0.01769
310	0.01770	0.01781	0.01797	0.01758	0.01756	0.01772
340	0.01775	0.01777	0.01748	0.01808	0.01791	0.01780

Berdasarkan hasil pengujian populasi Gambar 6.1 menunjukkan bahwa bertambahnya jumlah populasi rata-rata nilai fitness cenderung meningkat. Pada pengujian pertama jumlah *popsiz*e 10 memiliki fitness yang paling rendah yaitu 0.01328 dan jumlah populasi yang optimal adalah 250 dengan nilai rata-rata fitness 0.01785. Dari jumlah populasi 100 rata-rata fitness mengalami kenaikan yang tidak signifikan bahkan terjadi penurunan. Dapat dilihat pada jumlah populasi 250 dan 280 terdapat penurunan rata-rata nilai fitness namun kembali meningkat pada

jumlah populasi selanjutnya yaitu 310. Walaupun jumlah populasi 310 dan 340 mengalami kenaikan namun rata-rata fitness tidak melebihi nilai rata-rata fitness jumlah populasi 250 dan peningkatan rata-rata fitness tidak terlalu signifikan sehingga kondisi dihentikan. Menurut Mahmudy (2015) kondisi algoritma genetika terdapat kemungkinan mengeksplorasi area yang tidak memiliki nilai optimum sehingga eksplorasi harus dihentikan. Dari hasil tersebut menunjukkan bahwa semakin banyak jumlah populasi dalam suatu generasi maka fitness yang diperoleh cenderung meningkat.



Gambar 6.1 Grafik Pengujian Populasi

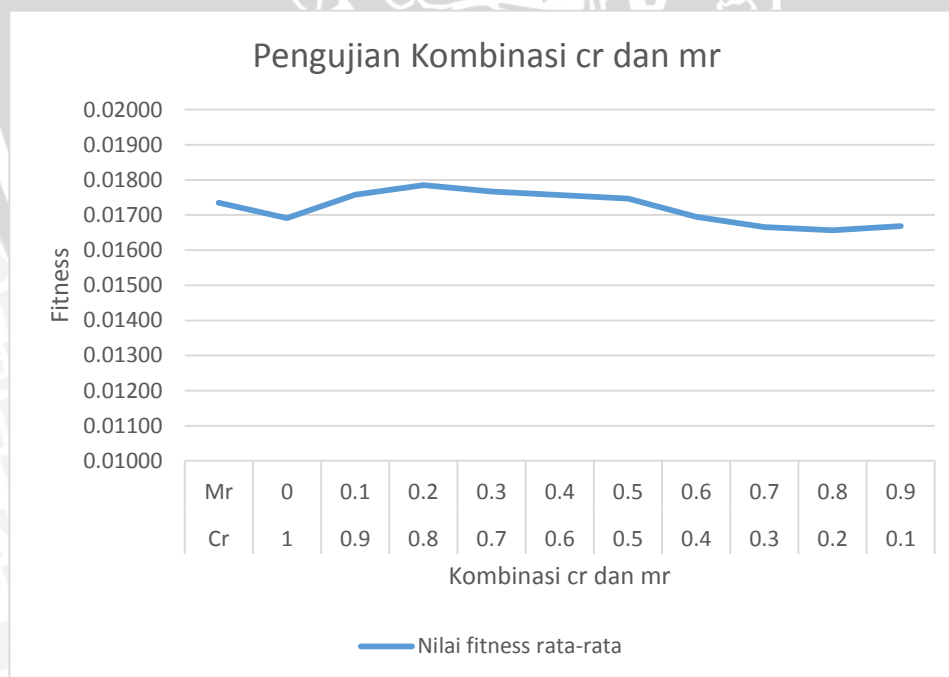
6.2 Pengujian Crossover Rate dan Mutation Rate

Setelah menemukan jumlah populasi (*popsiz*) yang paling optimal, perlu melakukan percobaan untuk mengetahui kombinasi *crossover rate* dan *mutation rate* yang optimal. Data pasien juga menggunakan data uji yang sama dengan pengujian populasi namun populasi yang digunakan adalah rata-rata fitness yang memiliki jumlah populasi tertinggi yaitu 250 individu dalam populasi. Skenario pengujian kombinasi *crossover rate* dan *mutation rate* menggunakan kombinasi nilai yaitu 1 dan 0, 0,9 dan 0,1, 0,8 dan 0,2, 0,7 dan 0,3, 0,6 dan 0,4, 0,5 dan 0,5 dan seterusnya seperti pada perancangan uji coba kombinasi *crossover rate* dan *mutation rate*. Hasil pengujian *crossover rate* dan *mutation rate* dapat dilihat pada Tabel 6.2 berikut.

Tabel 6.2 Pengujian Crossover rate dan Mutation rate

Kombinasi		Nilai Fitnes					Nilai fitness rata-rata
		Percobaan ke					
Cr	Mr	1	2	3	4	5	
1	0	0.01657	0.01715	0.01788	0.01759	0.01754	0.01735
0.9	0.1	0.01626	0.01629	0.01624	0.01772	0.01805	0.01691
0.8	0.2	0.01771	0.01761	0.01789	0.01756	0.01711	0.01758
0.7	0.3	0.01754	0.01766	0.01795	0.01797	0.01814	0.01785
0.6	0.4	0.01771	0.01732	0.01755	0.01759	0.01818	0.01767
0.5	0.5	0.01720	0.01756	0.01767	0.01782	0.01758	0.01757
0.4	0.6	0.01764	0.01751	0.01720	0.01768	0.01732	0.01747
0.3	0.7	0.01757	0.01645	0.01658	0.01690	0.01725	0.01695
0.2	0.8	0.01604	0.01699	0.01706	0.01625	0.01694	0.01666
0.1	0.9	0.01610	0.01704	0.01694	0.01623	0.01651	0.01656
0	1	0.01669	0.01730	0.01672	0.01632	0.01642	0.01669

Dari hasil pengujian kombinasi *crossover rate* dan *mutation rate* didapatkan hasil bahwa kombinasi *crossover rate* 0.7 dan *mutation rate* 0.3 adalah kombinasi *crossover rate* dan *mutation rate* yang paling optimal. Kombinasi tertinggi kedua adalah kombinasi cr=0.8 dan mr=0.2, ketiga cr=0.6 dan mr=0.4 dan yang terendah adalah kombinasi cr=0 dan mr=1. Sehingga kombinasi cr dan mr yang paling optimal adalah kombinasi cr=0.7 dan mr=0.3.



Gambar 6.2 Grafik Pengujian kombinasi cr dan mr

Hasil kombinasi cr dan mr pada Gambar 6.2 menunjukkan bahwa nilai cr yang lebih tinggi memiliki rata-rata fitness lebih besar. Penentuan *crossover rate* dan *mutation rate* yang tepat perlu dilakukan karena dapat menentukan kemampuan algoritma genetika dalam mengeksplorasi dan mengeksploitasi solusi seimbang (Lozano & Herrera, 2003). Cr yang lebih tinggi menunjukkan pada proses reproduksi sangat terganutng pada proses *crossover* sehingga *child* yang dihasilkan pada proses *crossover* menjadi lebih tinggi dibandingkan *child* yang dihasilkan proses mutasi. Dengan banyaknya *child* yang dihasilkan pada proses *crossover* menyebabkan *child* memiliki kemiripan dengan *parent* dari generasi sebelumnya dan menyebabkan konvergensi dini serta mutasi yang rendah dapat menyebabkan level eksplorasi dan konvergensi populasi menjadi rendah sehingga kemampuan mengeksplorasi area lain memiliki kemungkinan yang sangat tinggi (Mahmudy, 2015).

6.3 Pengujian Generasi

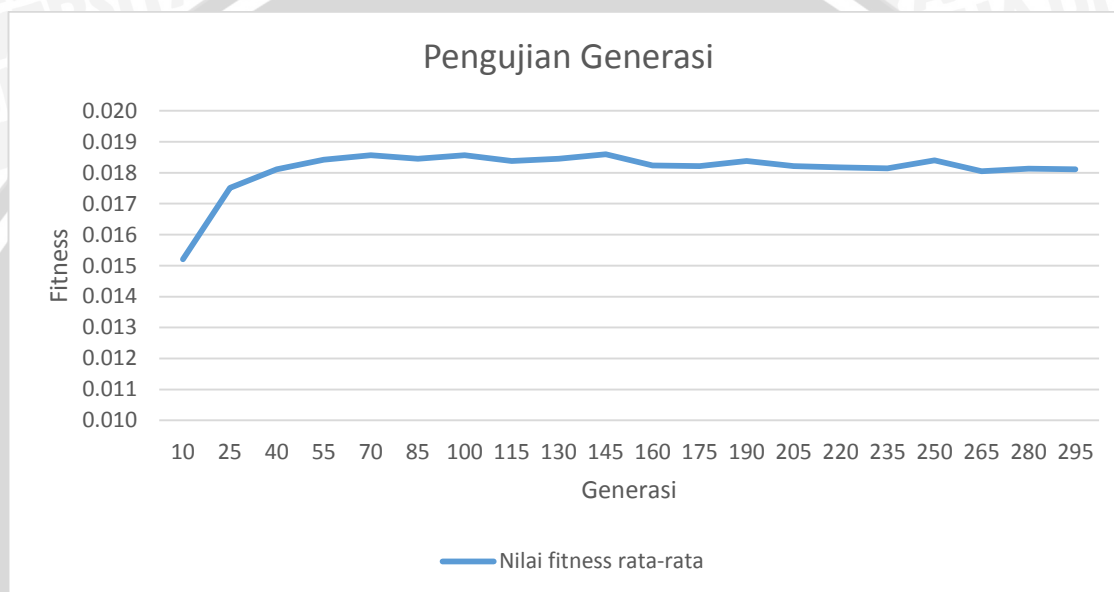
Pengujian Generasi digunakan untuk mengetahui generasi yang paling optimal sehingga diharapkan dengan optimalnya generasi solusi yang dihasilkan juga optimal. Pengujian yang dilakukan hampir sama dengan pengujian populasi dan kombinasi *crossover rate* dan *mutation rate* namun populasi, *crossover rate* dan *mutation rate* yang digunakan merupakan hasil yang optimal yang didapatkan pada pengujian sebelumnya yaitu populasi = 250, kombinasi *crossover rate* = 0.7 dan *mutation rate* = 0.3. Hasil dari pengujian generasi dapat dilihat pada tabel 6.3 berikut.

Tabel 6.3 Pengujian Generasi

Ukuran generasi	Nilai fitness					Nilai fitness rata-rata
	Percobaan ke					
	1	2	3	4	5	
10	0.015028	0.015182	0.015973	0.014593	0.015229	0.0152009
25	0.017227	0.017508	0.018124	0.017700	0.017024	0.0175165
40	0.018296	0.017778	0.018390	0.018265	0.017846	0.0181152
55	0.018558	0.018256	0.018346	0.018694	0.018239	0.0184185
70	0.018587	0.018763	0.018475	0.018506	0.018523	0.0185708
85	0.018506	0.018405	0.018461	0.018520	0.018397	0.0184577
100	0.018665	0.018472	0.018587	0.018589	0.018541	0.0185706
115	0.018246	0.018223	0.018441	0.018658	0.018324	0.0183784
130	0.018163	0.018602	0.018489	0.018568	0.018431	0.0184507
145	0.018763	0.018561	0.018516	0.018592	0.018568	0.0186001
160	0.017993	0.017792	0.018592	0.018558	0.018226	0.0182323
175	0.018534	0.018798	0.018249	0.018520	0.016973	0.0182149
190	0.017929	0.018173	0.018499	0.018755	0.018558	0.0183829
205	0.018084	0.018286	0.018373	0.018026	0.018326	0.0182192
220	0.018686	0.017290	0.017897	0.018506	0.018472	0.0181702
235	0.018738	0.018357	0.016920	0.018323	0.018377	0.0181431

Ukuran generasi	Nilai fitness					Nilai fitness rata-rata
	Percobaan ke					
	1	2	3	4	5	
250	0.018280	0.018363	0.018686	0.018203	0.018449	0.0183961
265	0.018492	0.018403	0.018516	0.017961	0.016899	0.0180543
280	0.018609	0.018313	0.018404	0.018489	0.016834	0.0181299
295	0.018479	0.016870	0.018401	0.018627	0.018163	0.0181080

Dari hasil pengujian generasi pada Tabel 6.3 didapatkan bahwa generasi terbaik adalah generasi 145 dengan rata-rata fitness 0.0186001. Grafik hasil pengujian generasi dapat dilihat pada Gambar 6.3



Gambar 6.3 Pengujian Generasi

Hasil pengujian generasi pada Gambar 6.3 menunjukkan bahwa rata-rata fitness semakin meningkat dengan bertambahnya jumlah generasi namun mulai generasi 55 sampai generasi 250 memiliki rata-rata fitness meningkat dan menurun serta puncak tertinggi generasi adalah 145 dimana nilai terendah adalah generasi 10 dengan rata-rata fitness adalah 0.0152009. Nilai rata-rata fitness terendah akan dimiliki oleh jumlah generasi terendah hal ini terjadi karena algoritma genetika belum menghasilkan solusi yang terbaik (Permatasari & Mahmudy, 2015). Setelah mendapatkan nilai tertinggi rata-rata fitness menurun dan agak meningkat pada generasi 190 namun kembali menurun lagi di generasi selanjutnya. Pada generasi 250 memiliki kenaikan fitness yang hampir sama dengan 190 namun kembali menurun. Sehingga generasi paling optimal dalam permasalahan ini adalah 145. Pada tahap seleksi terdapat mekanisme dalam menangani terjadinya konvergensi dini yaitu *random injection*. Untuk mempersingkat waktu komputasi *random injection* dilakukan sesuai interval g generasi dimana interval tersebut didapatkan dari percobaan yang dilakukan berkali-kali. Percobaan dilakukan dengan memasukkan nilai parameter inputan yang sama yaitu nilai generasi = 50, populasi = 50, $cr:mr=0.7:0.3$ dan dilakukan berkali-kali. Hasil perhitungan fitness didapatkan bahwa konvergensi nilai sering terjadi pada generasi 15. Sehingga

random injection akan dilakukan setiap interval 15 generasi. Pada pengujian generasi terbukti pada generasi 15 sampai seterusnya memiliki nilai yang bervariasi dan sehingga dapat disimpulkan permasalahan konvergensi ini dapat teratasi.

6.4 Pengujian Kromosom

Pengujian kromosom merupakan pengujian yang dilakukan untuk menguji kromosom yang sudah dirancang sesuai atau tidak dengan permasalahan yang diteliti. Kromosom dikatakan sesuai jika jumlah gen dalam kromosom berjumlah 15 dan masing-masing nilai gen tidak melebihi batas dari jumlah makanan. Pengujian kromosom dilakukan dengan memasukkan data pasien dan data parameter algoritma lalu diproses sesuai tahapan-tahapannya. Input dari pengujian kromosom merupakan generasi sebesar 235, popsize = 250, kombinasi *crossover rate* = 0.7 dan *mutation rate* = 0.3. Selain itu data pasien menggunakan nama Abdullah, tinggi badan = 165 cm, berat badan = 60 cm, umur = 50 tahun, aktifitas = ringan, dan komplikasi yang diderita pasien adalah ginjal. Input dari pengujian kromosom dapat dilihat pada Gambar 6.4 dan hasil pengujian kromosom dapat dilihat pada Gambar 6.5.

Optimasi Komposisi Makanan pada Penderita Diabetes Meltus dan Komplikasinya
Menggunakan Algoritma Genetika

Data Input

Data Pasien

Nama: Abdullah

Umur: 52

Tinggi Badan: 165

Berat Badan: 65

Jenis Kelamin: Laki-laki

Aktifitas: Ringan

Komplikasi: ☐ Mata ☐ Stroke ☐ Saraf ☐ Jantung ☒ Ginjal ☐ Tidak Komplikasi ☐ Lambung

Generasi: 235

Popsize: 250

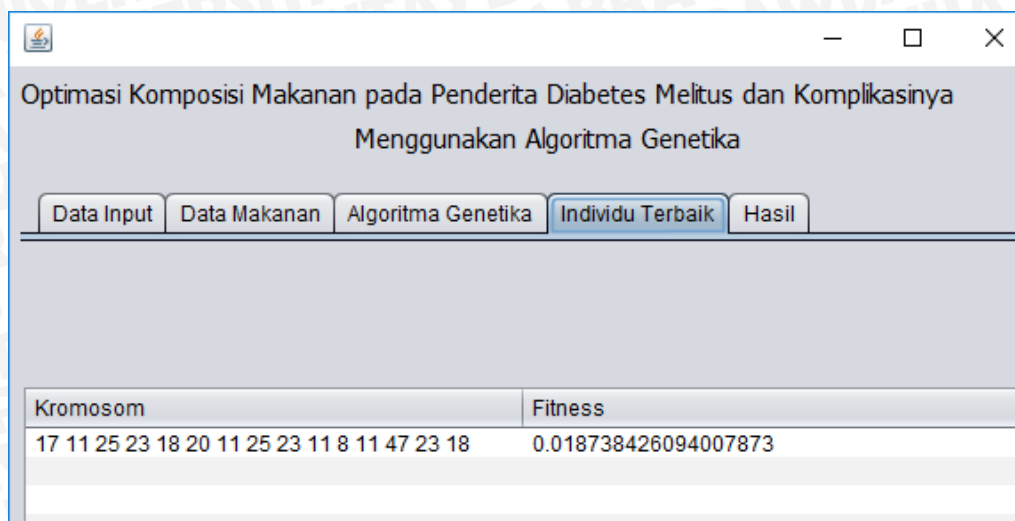
Crossover rate: 0.7

Mutation rate: 0.3

Hitung

Maryamah 135150201111008 TIF-13

Gambar 6.4 Input Pengujian Kromosom



Gambar 6.5 Individu Terbaik Pengujian Kromosom

Pada Gambar 6.5 menunjukkan bahwa hasil kromosom pada individu terbaik terdiri dari gen dan keterangan seperti pada Gambar 6.6

SP	SN	SH	SS	P	SP	SN	SH	SS	P	SP	SN	SH	SS	P
17	11	25	23	18	20	11	25	23	11	8	11	47	23	18

Gambar 6.6 Hasil Kromosom

Keterangan :

SP = Sumber Makanan Pokok

SH = Sumber Hewani

SN = Sumber Nabati

SS = Sumber Sayuran

P = Sumber Pelengkap (Buah-buahan)

Masing-masing bahan makanan memiliki jumlah bahan yang berbeda-beda yang telah dijelaskan sebelumnya pada Bab Perancangan yaitu 22 bahan makanan pokok, 12 bahan makanan nabati, 48 bahan makanan hewani, 36 bahan makanan sayuran dan 38 bahan makanan pelengkap (buah) . Setiap gen dalam kromosom wajib tidak melebihi jumlah atau menu yang dihasilkan akan tidak sesuai dengan kebutuhan atau menu yang akan dirancang. Pada Tabel 6.4 sumber pokok memiliki gen 17, 20 dan 8. Ketiga gen tersebut tidak melebihi batas jumlah makanan pokok yakni 22 sehingga sumber pokok sesuai. Pada sumber nabati terdiri dari gen bernomor 11, 11, 11 ketiganya tidak melebihi jumlah sumber nabati sehingga gen sumber nabati sesuai. Pada sumber hewani terdiri dari gen 25, 25, 47 yang ketiganya juga sesuai. Sumber sayuran memiliki gen 23, 23, 23 yang ketiganya tidak melebihi batas dan sumber pelengkap yang terdiri dari 18, 11, 18 dimana ketiganya tidak memiliki batas. Sehingga dapat disimpulkan bahwa perancangan kromosom yang dilakukan pada komposisi makanan sudah sesuai.

6.5 Analisis Pengujian

Pengujian yang dilakukan menggunakan data pasien:

Nama : Abdullah

Umur : 52 tahun

Tinggi badan : 165 cm

Berat badan : 65 cm

Jenis Kelamin : Laki-laki

Aktifitas : Ringan

Komplikasi : Ginjal

Sesuai data pasien yang dimasukkan kebutuhan kalori seimbang harian yang harus dipenuhi sebesar 2193,75 dengan kebutuhan seimbang karbohidrat, protein dan lemak sebesar 318,09375, 29,22075, 164,53125. Hasil yang didapatkan pada parameter algoritma genetika yaitu jumlah individu dalam populasi yang optimal berjumlah 250, kombinasi cr dan mr adalah 0.7 dan 0.3, jumlah generasi adalah 145.

Optimasi Komposisi Makanan pada Penderita Diabetes Melitus dan Komplikasinya
Menggunakan Algoritma Genetika

Data Input Data Makanan Algoritma Genetika Individu Terbaik Hasil

Data Pasien

Nama: Abdullah

Umur: 52

Tinggi Badan: 165

Berat Badan: 65

Jenis Kelamin: Laki-laki

Aktifitas: Ringan

Komplikasi: Ginjal

Kebutuhan Gizi

Energi: 2193.75

Karbohidrat: 318.09375

Protein: 29.22075

Lemak: 164.53125

Nama menu	Kalori	Karbohidrat	Lemak	Protein	Biaya
Jambu biji	50	10	0	0	700
Malam					
Kentang	125	3	29	0	1800
Tahu	40	4	3	2	300
Telur bebek	218	0	7	13	1600
Labu waluh	50	10	2	0	666
Kurma	333	67	0	0	1200
Total	2140.0	188.0	211.0	31.0	11906.0

Maryamah 135150201111008 TIF-13

Gambar 6.7 Hasil Pengujian

Setelah melakukan tahap-tahapan algoritma genetika solusi yang dihasilkan seperti pada Gambar 6.7 memiliki total kalori sebesar 2140 dengan total harga 11905 rupiah. Total karbohidrat, lemak, dan protein sebesar 188, 31, 211 dengan komposisi makanan seperti pada Tabel 6.5. Sesuai dengan penyakit komplikasi pasien yaitu ginjal maka kebutuhan makanan yang wajib terpenuhi adalah total kalori dan total protein harian yang dihasilkan. Dari hasil yang didapatkan total kalori tidak dapat melebihi 100 kkal dan hasil yang didapatkan memenuhi total kalori dan tidak lebih atau kurang. Total protein yang dihasilkan tidak melebihi 50 kkal atau 5.5 gram dari kebutuhan seimbang dan hasil yang didapatkan memenuhi total protein juga tidak kurang atau lebih. Namun pada total karbohidrat yang dihasilkan melebihi total karbohidrat seimbang begitu pula dengan total lemak. Walaupun total karbohidrat dan lemak tidak diutamakan namun akan lebih baik jika solusi tidak melebihi atau kurang dari total kebutuhan seimbang harian. Sehingga dari hasil pengujian dapat disimpulkan bahwa solusi yang dihasilkan cukup optimal namun akan mendapatkan hasil yang lebih baik jika seluruh kebutuhan seimbang terpenuhi.

Tabel 6.4 Hasil Menu Makanan

Individu	Bahan makanan	Energi (Kkal)	Karbohidrat (gram)	Protein (gram)	Lemak (gram)	Harga (Rp)
Makan pagi	Tape Beras Ketan	263	6	60	0	480
	Kacang Kedelai	160	16	12	6	600
	Ikan lemur	114	0	7	2	640
	Daun katuk	100	20	6	0	750
	Jambu biji	50	10	0	0	700
Makan siang	Tape Beras Ketan	263	6	60	0	480
	Kacang Kedelai	160	16	12	6	600
	Ikan lemur	114	0	7	2	640
	Daun katuk	100	20	6	0	750
	Jambu biji	50	10	0	0	700
Makan malam	Kentang	125	3	29	0	1800
	Tahu	40	4	3	2	300
	Telur bebek	218	0	7	13	1600
	Labu waluh	50	10	2	0	666
	Kurma	333	67	0	0	1200

BAB 7 PENUTUP

7.1 Kesimpulan

Pada permasalahan optimasi komposisi makanan pada penderita diabetes melitus dan komplikasinya menggunakan algoritma genetika mendapatkan kesimpulan bahwa

1. Representasi kromosom yang dirancang dalam permasalahan sudah sesuai dengan permasalahan optimasi komposisi makanan pada penderita diabetes melitus dan komplikasinya. Representasi kromosom yang digunakan adalah representasi integer yang masing-masing dari susunan gen memiliki jumlah 15 dalam kromosom dibatasi angka *random* yang digunakan sesuai dengan jumlah data makanan yang didapatkan.
2. Komposisi makanan yang dihasilkan sudah optimal dikarenakan pada perhitungan fitness terdapat aturan-aturan yang harus dipenuhi. Aturan tersebut mewakili informasi pakar dalam menentukan komposisi makanan yang optimal. Hal ini dibuktikan pada penjelasan analisis pengujian komposisi makanan yang dihasilkan tidak melanggar aturan utama diet diabetes melitus dan komplikasinya sehingga dapat disimpulkan bahwa nilai fitness pada individu terbaik menghasilkan komposisi makanan yang paling optimal.
3. Penerapan metode algoritma genetika dalam permasalahan dilakukan dengan melakukan inisialisasi awal yang berisi representasi kromosom. Dilanjutkan dengan proses *crossover one cut point* dan mutasi *reciprocal exchange mutation*, evaluasi atau perhitungan fitness sesuai rumus yang ditentukan, dan seleksi yang digunakan metode *elitism*. Hasil penerapan metode algoritma genetika mendapatkan nilai generasi paling optimal adalah 145, populasi = 250, kombinasi *cr* dan *mr* = 0.7 dan 0.3 serta fitness terbaik 0.01857.

7.2 Saran

Saran yang dapat digunakan pada penelitian selanjutnya adalah

1. Diharapkan pada penelitian selanjutnya dilakukan hybrid dengan metode yang metode local search seperti *Variable Neighborhood Search*, *Simulated Annealing*, atau metode yang lain sehingga dapat menghasilkan hasil yang lebih optimal.
2. Proses *crossover*, mutasi, dan seleksi dilakukan dengan metode lain sehingga didapatkan hasil yang berbeda.

DAFTAR PUSTAKA

- Beck, M. E., 2011. *IMU GIZI DAN DIET Hubungannya dengan Penyakit-penyakitnya untuk Perawat dan Dokter*. Yogyakarta: Andi.
- Ernawati, 2013. *Penatalaksanaan Keperawatan Diabetes Melitus Terpadu dengan Penerapan Teori Keperawatan Self Care Orem*. Jakarta: Mitra Wacana Media.
- Ganguly, S., Datta, S. & Chakraborti, N., 2009. Genetic algorithm based search on the role of variables in the work hardening process of multiphase steels. *Computational Materials Science*, Issue 45, p. 158–166.
- Kingham, K., 2009. *Makan Oke Hidup Oke dengan Diabetes*. Jakarta: Erlangga.
- Lozano, M. & Herrera, F., 2003. Fuzzy adaptive genetic algorithms: design, taxonomy, and future directions. *Soft Computing*, Volume 7, p. 545–562.
- Mahmudy, W. F., 2015. *Algoritma Genetika*. Malang: Fakultas Ilmu Komputer.
- Mahmudy, W. F., 2015. Optimization of part type selection and machine loading problems in flexible manufacturing system using variable neighborhood search. *IAENG International Journal of Computer Science*, Volume 42, pp. 254-264.
- Paszkowicz, W., 2009. Properties of a genetic algorithm equipped with a dynamic penalty function. *Computational Materials Science*, Issue 45, p. 77–83.
- Permatasari, A. I. & Mahmudy, W. F., 2015. Pemodelan regresi linear dalam konsumsi Kwh listrik di Kota Batu menggunakan algoritma genetika. *Doro: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya*, 5(14).
- Prameswari, d. A., 2016. *Penyakit Diabetes Mellitus dan Komplikasinya* [Interview] (11 November 2016).
- Ramuna, M. & Mahmudy, W. F., 2015. Optimasi persediaan barang dalam produksi jilbab menggunakan algoritma genetika. *DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya*, Volume 5, p. 14.
- Republik Indonesia. 2014. "Peraturan Menteri Kesehatan Nomor 41 Tahun 2014 tentang Pedoman Gizi Seimbang"
- Rianawati, A. & Mahmudy, W. F., 2015. Impelementasi Algoritma Genetika untuk Optimasi Komposisi Makanan bagi Penderita Diabetes Mellitus. Volume V, p. 14.
- Romahorbo, D. H., 2014. *Mencegah Diabetes Melitus dengan Perubahan Gaya Hidup*. 1st ed. Bogor: IN MEDIA.
- Saputro, H. & Mahmudy, W. F., 2015. Implementasi Algoritma Genetika untuk optimasi penggunaan lahan pertanian. *DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya*, Volume 5, p. 12.
- Sari, D. & Mahmudy, W. F., 2015. Optimasi penjadwalan mata pelajaran menggunakan algoritma genetika (studi kasus : SMPN 1 Gondang Mojokerto). *DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya*, Volume 5, p. 13.

Suci, W. W. & Mahmudy, W. F., 2015. Optimasi Biaya Pemenuhan Gizi dan Nutrisi pada Manusia Lanjut Usia Menggunakan Algoritma Evolusi. Volume 5, p. 17.

Susilo, d. Y. & Wulandari, A., 2011. *Cara Jitu Mengatasi Diabetes Melitus (Kencing Manis)*. Yogyakarta: Andi.

Sutanto, 2010. *CEKAL (Cegah & Tangkal) Penyakit Modern*. Yogyakarta: Andi.

Tandra, P. D. H., 2013. *Life Healty with Diabetes Diabetes Mengapa & Bagaimana?*. Yogyakarta: Andi.

Tjokroprawiro, A., 2012. *Garis Besar Pola Makan dan Pola Hidup sebagai Pendukung Diabetes Melitus*. Surabaya: Fakultas Kedokteran Universitas Airlangga.

Wahid, N. & Mahmudy, W. F., 2015. Optimasi komposisi makanan untuk penderita kolesterol menggunakan algoritma genetika. *DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya*, Volume 5, p. 15.

Wahyuningsih, R., 2013. *Penatalaksanaan Diet pada Pasien*. Edisi Pertama ed. Yogyakarta: Graha Ilmu.

Zukhri, Z., 2014. *Algoritma Genetika*. Yogyakarta: Andi Yogyakarta.



LAMPIRAN

Lampiran A Data Bahan Makanan

No	Nama Pangan	Berat Gram	Kalori kkal	Karbohidrat gram	Protein gram	lemak gram	harga
1	Bihun	50	525	120	12	0	2700
2	Biskuit	40	656.25	150	15	0	2400
3	Havermut	45	583.3333	133.3333	13.33333	0	3600
4	Jagung Segar	125	210	48	4.8	0	900
5	Kentang	210	125	28.57143	2.857143	0	1800
6	Kentang Hitam	125	210	48	4.8	0	1800
7	Makaroni	50	525	120	12	0	2400
8	Mie Basah	200	131.25	30	3	0	1800
9	Mie Kering	50	525	120	12	0	1800
10	Beras Giling putih	100	262.5	60	6	0	1500
11	Beras Giling Merah	100	262.5	60	6	0	1800
12	Beras Giling Hitam	100	262.5	60	6	0	3300
13	Beras ½ Giling	100	262.5	60	6	0	1500
14	Nasi Ketan Putih	100	262.5	60	6	0	3000
15	Roti Putih	70	375	85.71429	8.571429	0	1500
16	Roti Warna Coklat	70	375	85.71429	8.571429	0	1750
17	Singkong	120	218.75	50	5	0	450
18	Sukun	150	175	40	4	0	1050
19	Talas	125	210	48	4.8	0	1050
20	Tape Beras Ketan	100	262.5	60	6	0	480
21	Tape Singkong	100	262.5	60	6	0	900
22	Ubi Jalar Kuning	135	194.4444	44.44444	4.444444	0	750
23	Kacang Hijau	25	160	16	12	6	1000
24	Kacang Kedelai	25	160	16	12	6	600
25	Kacang Mete	15	266.6667	26.66667	20	10	5000
26	Kacang Merah	25	160	16	12	6	1200
27	Kacang Tanah	20	200	20	15	7.5	1250
28	Kacang Toto	20	200	20	15	7.5	1500
29	Keju Kacang Tanah	15	266.6667	26.66667	20	10	2500
30	Kembang Tahu	20	200	20	15	7.5	2000
31	Oncom	50	80	8	6	3	400
32	Petai Segar	20	200	20	15	7.5	1250
33	Tahu	100	40	4	3	1.5	300
34	Sari Kedelai	185	21.62162	2.162162	1.621622	0.810811	500
35	Daging sapi	35	114.2857	0	7	2	8800
36	Daging ayam	40	100	0	7	2	2400
37	Hati Sapi	50	80	0	7	2	6400

No	Nama Pangan	Berat Gram	Kalori kkal	Karbohidrat gram	Protein gram	lemak gram	harga
38	Ikan Asin	15	266.6667	0	7	2	2400
39	Ikan Teri Kering	20	200	0	7	2	4000
40	Telur Ayam	55	72.72727	0	7	2	1360
41	Udang Basah	35	114.2857	0	7	2	5600
42	Babat	40	100	0	7	2	3600
43	Cumi-cumi	45	88.88889	0	7	2	4000
44	Daging asap	20	200	0	7	2	8800
45	Daging kerbau	35	114.2857	0	7	2	8000
46	Dendeng sapi	15	266.6667	0	7	2	8800
47	Gabus kering	10	400	0	7	2	2400
48	Otak	65	92.30769	0	7	5	2800
49	Ikan asin kering	15	266.6667	0	7	2	2400
50	Ikan kakap	35	114.2857	0	7	2	4000
51	Ikan kembung	30	133.3333	0	7	2	2400
52	Ikan lele	40	100	0	7	2	1600
53	Ikan mas	45	88.88889	0	7	2	1920
54	Ikan mujair	30	133.3333	0	7	2	2240
55	Ikan peda	35	114.2857	0	7	2	3200
56	Ikan pindang	25	160	0	7	2	2400
57	Ikan cakalang asin	20	200	0	7	2	2800
58	Kerang	90	44.44444	0	7	2	3200
59	Ikan lemuru	35	114.2857	0	7	2	640
60	Putih telur ayam	65	61.53846	0	7	2	1360
61	Rebon kering	10	400	0	7	2	4480
62	Rebon basah	45	88.88889	0	7	2	4480
63	Selar kering	20	200	0	7	2	2400
64	Sepat kering	20	200	0	7	2	2400
65	Teri nasi	20	200	0	7	2	3200
66	Udang segar	35	114.2857	0	7	2	5600
67	Bakso	170	35.29412	0	7	5	2133.333
68	Daging kambing	40	150	0	7	5	6400
69	Ginjal sapi	45	133.3333	0	7	5	6400
70	Hati ayam	30	200	0	7	5	1600
71	Telur bebek asin	50	120	0	7	5	4000
72	Telur puyuh	55	109.0909	0	7	5	2080
73	Usus sapi	50	120	0	7	5	4800
74	Daging Bebek	45	266.6667	0	7	13	2880
75	Belut	45	266.6667	0	7	13	1920
76	Kornet daging sapi	45	266.6667	0	7	13	3000
77	Ayam dengan kulit	40	300	0	7	13	2400
78	Ham	40	300	0	7	13	3000

No	Nama Pangan	Berat	Kalori	Karbohidrat	Protein	lemak	harga
		Gram	kkal	gram	gram	gram	
79	Sardencis	35	342.8571	0	7	13	2400
80	Sosis	50	240	0	7	13	1600
81	Kuning telur ayam	45	266.6667	0	7	13	1360
82	Telur bebek	55	218.1818	0	7	13	1600
83	Bayam	100	50	10	2	0	1000
84	Kapri muda	100	50	10	2	0	4800
85	Brokoli	100	50	10	2	0	2400
86	Kembang kol	100	50	10	2	0	3000
87	Kemangi	100	50	10	2	0	3333.333
88	Taoge	100	50	10	2	0	1600
89	Pare	100	50	10	2	0	1400
90	Kangkung	100	50	10	2	0	1000
91	Terong	100	50	10	2	0	1400
92	Wortel	100	50	10	2	0	1600
93	Bit	100	50	10	2	0	1400
94	Kol	100	50	10	2	0	1200
95	Labu waluh	100	50	10	2	0	666
96	Daun talas	100	50	10	2	0	2000
97	Daun kecipir	100	50	10	2	0	1066.667
98	Buncis	100	50	10	2	0	1400
99	Daun kacang panjang	100	50	10	2	0	1800
100	Genjer	100	50	10	2	0	1200
101	Jagung muda	100	50	10	2	0	4000
102	Kacang panjang	100	50	10	2	0	1200
103	Sawi	100	50	10	2	0	1500
104	Rebung	100	50	10	2	0	1600
105	Pepaya muda	100	50	10	2	0	600
106	Labu Siam	100	50	10	2	0	1400
107	Bayam merah	100	100	20	6	0	1200
108	Mangkakan	100	100	20	6	0	1000
109	Nangka muda	100	100	20	6	0	1400
110	Daun pepaya	100	100	20	6	0	1000
111	Daun katuk	100	100	20	6	0	750
112	kacang kapri	100	100	20	6	0	8000
113	Mlinjo	100	100	20	6	0	2500
114	Taoge kedelai	100	100	20	6	0	1600
115	Daun Melinjo	100	100	20	6	0	1800
116	Daun talas	100	100	20	6	0	1400
117	Kluwih	100	100	20	6	0	5600
118	Daun singkong	100	100	20	6	0	1000

No	Nama Pangan	Berat	Kalori	Karbohidrat	Protein	lemak	harga
		Gram	kkal	gram	gram	gram	
119	Alpokot	50	100	20	0	0	1000
120	Anggur	165	30.30303	6.060606	0	0	2000
121	Apel merah	85	58.82353	11.76471	0	0	2200
122	Apel malang	75	66.66667	13.33333	0	0	1200
123	Belimbing	125	40	8	0	0	1000
124	Blewah	70	71.42857	14.28571	0	0	800
125	Duku	80	62.5	12.5	0	0	2000
126	Durian	35	142.8571	28.57143	0	0	3000
127	Jambu air	100	50	10	0	0	1500
128	Jambu biji	100	50	10	0	0	700
129	Jambu bol	90	55.55556	11.11111	0	0	700
130	Jeruk bali	105	47.61905	9.52381	0	0	1200
131	Jeruk garut	115	43.47826	8.695652	0	0	1000
132	Jeruk manis	100	50	10	0	0	1000
133	Jeruk nipis	135	37.03704	7.407407	0	0	700
134	Kedondong	100	50	10	0	0	800
135	Kesemek	65	76.92308	15.38462	0	0	1000
136	Kurma	15	333.3333	66.66667	0	0	1200
137	Leci	75	66.66667	13.33333	0	0	2500
138	Mangga	90	55.55556	11.11111	0	0	2000
139	Manggis	80	62.5	12.5	0	0	1500
140	Markisa	35	142.8571	28.57143	0	0	2000
141	Melon	90	55.55556	11.11111	0	0	1000
142	Nangka masak	50	100	20	0	0	1000
143	Nanas	85	58.82353	11.76471	0	0	700
144	Pear	85	58.82353	11.76471	0	0	1500
145	Pepaya	100	50	10	0	0	700
146	Pisang ambon	50	100	20	0	0	2000
147	Pisang kepok	45	111.1111	22.22222	0	0	1500
148	Pisang mas	40	125	25	0	0	1500
149	Pisang raja	40	125	25	0	0	2000
150	Rambutan	75	66.66667	13.33333	0	0	1000
151	Sawo	50	100	20	0	0	1200
152	Salak	65	76.92308	15.38462	0	0	700
153	Semangka	180	27.77778	5.555556	0	0	700
154	Sirsak	60	83.33333	16.66667	0	0	800
155	Srikaya	50	100	20	0	0	1000
156	Strawberry	215	23.25581	4.651163	0	0	2500

Lampiran B Data Uji

No	No RM	Jenis kelamin	Tanggal Lahir	Usia	Berat badan	Tinggi badan	Aktifitas	Komplikasi
1	16008862	Laki-laki	4/8/1936	81	55	160	Istirahat	Hypertensi
2	17000525	Laki-laki	24/03/1980	37	70	180	Sedang	Tidak Komplikasi
3	17003028	Perempuan	5/12/1953	64	50	155	Ringan	Hypertensi
4	17003740	Perempuan	2/2/1955	62	60	160	Ringan	Tidak Komplikasi
5	15005900	Perempuan	7/3/1951	66	45	145	Istirahat	Tidak Komplikasi
6	16013787	Perempuan	15/8/1954	63	65	153	Ringan	Jantung
7	17001752	Perempuan	29/5/1940	77	74	152	Ringan	Saraf
8	15004716	Laki-laki	1/1/1949	68	81	171	Berat	Jantung
9	17002971	Laki-laki	31/12/1963	54	65	165	Ringan	Ginjal
10	15004715	Perempuan	27/9/1947	70	49	150	Ringan	Low Back Pain (Saraf)

