



TESIS - KI092361

**OPERATOR MOMENTUM PADA ALGORITMA  
HEURISTIK UNTUK *SCREENING TEST* PENYAKIT  
DIABETES MELLITUS BERDASARKAN DATA  
URINALYSIS**

Abidatul Izzah  
5112201020

DOSEN PEMBIMBING  
Dr. Ir. R.V. Hari Ginardi, M.Sc.

PROGRAM MAGISTER  
BIDANG KEAHLIAN KOMPUTASI CERDAS & VISUALISASI  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2014



THESIS - KI092361

# MOMENTUM OPERATOR FOR HEURISTIC ALGORITHM FOR SCREENING TEST OF DIABETES MELLITUS BASED ON URINALYSIS

Abidatul Izzah  
5112201020

SUPERVISOR  
Dr. Ir. R.V. Hari Ginardi, M.Sc.

MASTER PROGRAM  
INFORMATICS ENGINEERING  
FACULTY OF INFORMATION TECHNOLOGY  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2014

# OPERATOR MOMENTUM PADA ALGORITMA HEURISTIK UNTUK *SCREENING TEST* PENYAKIT DIABETES MELLITUS BERDASARKAN DATA URINALYSIS

Nama mahasiswa : Abidatul Izzah  
NRP : 5112201020  
Pembimbing : Dr. Ir. R.V. Hari Ginardi, M.Sc.

## ABSTRAK

Pemeriksaan urin menggunakan *dipstick* merupakan uji urinalysis yang mudah dilakukan. Akan tetapi, interpretasi dari hasil pemeriksaan *dipstick* urin masih sulit dipahami oleh para penggunanya. Di bidang pengolahan data, diagnosis awal suatu penyakit sering dikaitkan dengan permasalahan klasifikasi. *Artificial Neural Network* (ANN) merupakan algoritma *supervised learning* yang memerlukan proses pembelajaran. Proses ini digunakan untuk menemukan bobot dan bias pada struktur ANN. Oleh karena itulah dibutuhkan sebuah metode pembelajaran yang dapat menentukan bobot dan bias yang optimal. Di sisi lain, *Gravitational Search Algorithm* (GSA) adalah algoritma heuristik baru yang diinspirasi dari fenomena alam. Sebagai algoritma heuristik, GSA memiliki kemampuan yang bagus dalam pencarian global. Namun jika konvergensi dini terjadi, algoritma ini akan berhenti dan tidak mampu lagi melakukan pencarian. Untuk memperbaiki kemampuan GSA maka diusulkan penambahan sebuah operator baru yang diturunkan dari hukum kekekalan momentum sehingga dapat mengeksplorasi solusi yang didapatkan oleh GSA.

Dengan demikian, pada tesis ini diusulkan sebuah algoritma heuristik baru yang disebut GSA-Momentum (GSAM). GSAM merupakan algoritma GSA dengan penambahan operator momentum. Dalam tesis ini, GSAM digunakan sebagai metode pembelajaran pada ANN. Metode ini kemudian digunakan sebagai pendekatan untuk *screening test* hiperglikemia, dimana hiperglikemia merupakan indikator dari penyakit Diabetes Mellitus. *Screening test* dilakukan berdasarkan hasil pemeriksaan *dipstick* urin menggunakan konsep klasifikasi data. Metode ini disimulasikan menggunakan MATLAB R2009a. Pada kasus klasifikasi dataset urin, hasil terbaik yang diperoleh dengan akurasi sebesar 78.2%, sensitivitas sebesar 66.5%, dan spesifisitas sebesar 89.9% dengan arsitektur 10-15-1. Hasil yang diperoleh menunjukkan bahwa adanya operator momentum berpengaruh pada kasus dataset berdimensi tinggi.

**Kata kunci:** *Artificial Neural Network*, Heuristik, Klasifikasi, *Screening Test*

# MOMENTUM OPERATOR FOR HEURISTIC ALGORITHM FOR SCREENING TEST OF DIABETES MELLITUS BASED ON URINALYSIS

Name : Abidatul Izzah  
Student Identity Number : 5112201020  
Supervisor : Dr. Ir. R.V. Hari Ginardi, M.Sc.

## ABSTRACT

Urinalysis using a dipstick is one of the simple methods for urine examination. However, interpretation of test results is still difficult to be understood by the user. In data mining, early diagnostic is always associated with classification. Artificial Neural Network (ANN) is a supervised learning algorithm which needs learning process. This process is used to find weight and bias for ANN. Therefore, there is need for a learning method to determine an optimal weight and bias. Moreover, Gravitational Search Algorithm (GSA) is a new heuristic algorithm which derives its inspiration from natural phenomenon. As a heuristic algorithm, GSA has a good performance in global searching. However, if premature convergence happens, GSA will stop and lose its ability to search. Therefore, a new operator that inspired by conservation of linear momentum is added to improve GSA's performance by exploring the solution.

In this thesis, a new heuristic algorithm named GSA–Momentum (GSAM) is proposed. It is a GSA with a momentum operator. It is used in this thesis as learning method in ANN for hyperglykemia screening test, which is an indicator of Diabetes Mellitus. Screening test is applied based on urinalysis result classification. The method will be simulated in MATLAB R2009a. In urine classification, we obtained accuracy 78.2%, sensitivity 66.5%, dan spesifisity 89.9% as the best result. The results show that the new operator takes effect in high dimension.

**Keywords:** Artificial Neural Network, Heuristic, Classification, Screening Test


**Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar  
Magister Komputer (M.Kom)  
di  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember**

**Oleh:  
ABIDATUL IZZAH  
NRP: 5112201020**

**Tanggal Ujian: 3 Juli 2014  
Periode Wisuda: September 2014**

**Disetujui oleh:**

**Dr. Ir. R.V. Hari Ginardi, M.Sc.  
NIP. 196505181992031003**

  
.....  
(Pembimbing I)

**Prof.Dr.Ir.Joko Lianto Buliali, M.Sc  
NIP. 196707271992031002**

  
.....  
(Penguji)

**Diana Purwitasari, S.Kom., M.Sc  
NIP. 197804102003122001**

.....  
(Penguji)

**Wijayanti Nurul Khotimah, S.Kom., M.Sc.  
NIP. 198603122012122004**

  
.....  
(Penguji)

**Direktur Program Pascasarjana,**

**Prof.Dr. Ir. Adi Soeprijanto, M.T.  
NIP. 19640405 199002 1001**



## KATA PENGANTAR

Alhamdulillah segala puji bagi Allah SWT. Penulis bersyukur atas rahmat dan pertolongan-Nya sehingga penulis dapat menyelesaikan Tesis ini. Tesis ini disusun untuk memenuhi salah satu persyaratan menyelesaikan studi di Program Magister Jurusan Teknik Informatika, Fakultas Teknologi Informasi (FTIF), Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Pada kesempatan ini penulis ingin mengucapkan banyak terima kasih kepada:

1. Ibu, kakak, dan adik serta Alm Bapak yang telah memberikan doa, motivasi, dukungan moril dan materiil serta pengertian selama penulis menyelesaikan studi ini.
2. Bapak Dr. Ir. R V Hari Ginardi, M.Sc selaku dosen pembimbing yang telah memberikan ilmu, meluangkan waktu, memberikan saran dan kritik yang membangun, serta memberikan banyak pengalaman yang berharga kepada penulis selama ini.
3. Bapak Prof. Riyanarto Sarno selaku ketua tim peneliti labaratorium sistem cerdas topik urinalysis.
4. Segenap staff Laboratorium Klinik Utama “Populer” Bendul Merisi, atas bantuan dalam pengambilan sampel *urine dipstick* sebagai data penunjang Tesis ini.
5. Tim Penguji Tesis, Bapak Prof. Joko Lianto, Ibu Diana Purwitasari, S.Jom, M.Sc, dan Ibu Wijayanti Nurul Khotimah, S.Kom, M.Sc atas saran dan kritik yang membangun untuk penyusunan Tesis ini. Tak lupa kepada Ibu Anny Yuniarti, S.Kom., M.Comp.Sc dan Bapak Arya Yudhi Wijaya., S.Kom, M.Kom. atas saran dan kritik yang membangun pada saat proposal Tesis.
6. Bapak Waskitho Wibisono, S.Kom, M.Eng, Ph.D selaku Ketua Program Studi Pascasarjana Informatika ITS, yang telah memberikan nasehat selama penulis menempuh studi S2.
7. Bapak Prof. Royyana Mulim selaku dosen wali saat penulis menempuh studi S2.

8. Teman-teman tim peneliti topik urinalysis yakni Mbak Ratna, Pak Tri, Pak Ali, dan Pak Taufik serta tim peneliti topik keragaan tebu yakni mbak Sari, Mbak Ratih, Mbak Mutrofin, Mbak Evi, dan Mbak Tari. Terima kasih atas kerjasama, semangat, dan motivasi selama penulis mengerjakan penelitian ini dalam menjalankan penelitian di Laboratorium Sistem Cerdas, Teknik Informatika ITS. Terima kasih semuanya.
9. Mbak Rini atas bantuan informasi dan adminstrasi yang berkaitan dengan ujian proposal, dan ujian Tesis.
10. Mas Kunto atas kesediaan menjaga Laboratorium Pascasarjana Informatika ITS, sebagai tempat untuk penulis mengerjakan Tesis.
11. Teman-teman seperjuangan di S2 Teknik Informatika ITS Angkatan 2012 dan teman-teman lain yang tidak dapat disebutkan satu persatu. Terima kasih atas doa, semangat, bahkan sekedar kesediaan mendengarkan keluh kesah penulis.

Dengan segala keterbatasan, penulis menyadari bahwa penelitian tesis ini masih jauh dari sempurna dan membutuhkan pengembangan lebih lanjut. Untuk itu penulis mengharapkan saran positif dari para pembaca. Akhir kata, penulis berharap semoga Penelitian ini dapat bermanfaat bagi banyak pihak terutama untuk pengembangan ilmu pengetahuan dan teknologi di bidang Komputasi Cerdas dan Visualisasi.

Surabaya, Juli 2014

Abidatul Izzah

## DAFTAR ISI

ABSTRAK.....	iii
ABSTRACT.....	v
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan dan Manfaat Penelitian.....	4
1.5 Kontribusi Penelitian.....	4
BAB 2 KAJIAN PUSTAKA.....	5
2.1 Pemeriksaan Urinalysis.....	5
2.2 <i>Screening test</i> pada Diabetes Mellitus.....	7
2.3 <i>Gravitational Search Algorithm</i> dan Hukum Kekekalan Momentum.....	7
2.4 <i>Artificial Neural Network</i> .....	14
2.5 GSA sebagai Pembelajaran pada ANN.....	16
BAB 3 METODE PENELITIAN.....	19
3.1 Studi literatur.....	19
3.2 Pengumpulan data.....	19
3.3 Desain sistem.....	21
3.4 Pengujian sistem.....	27

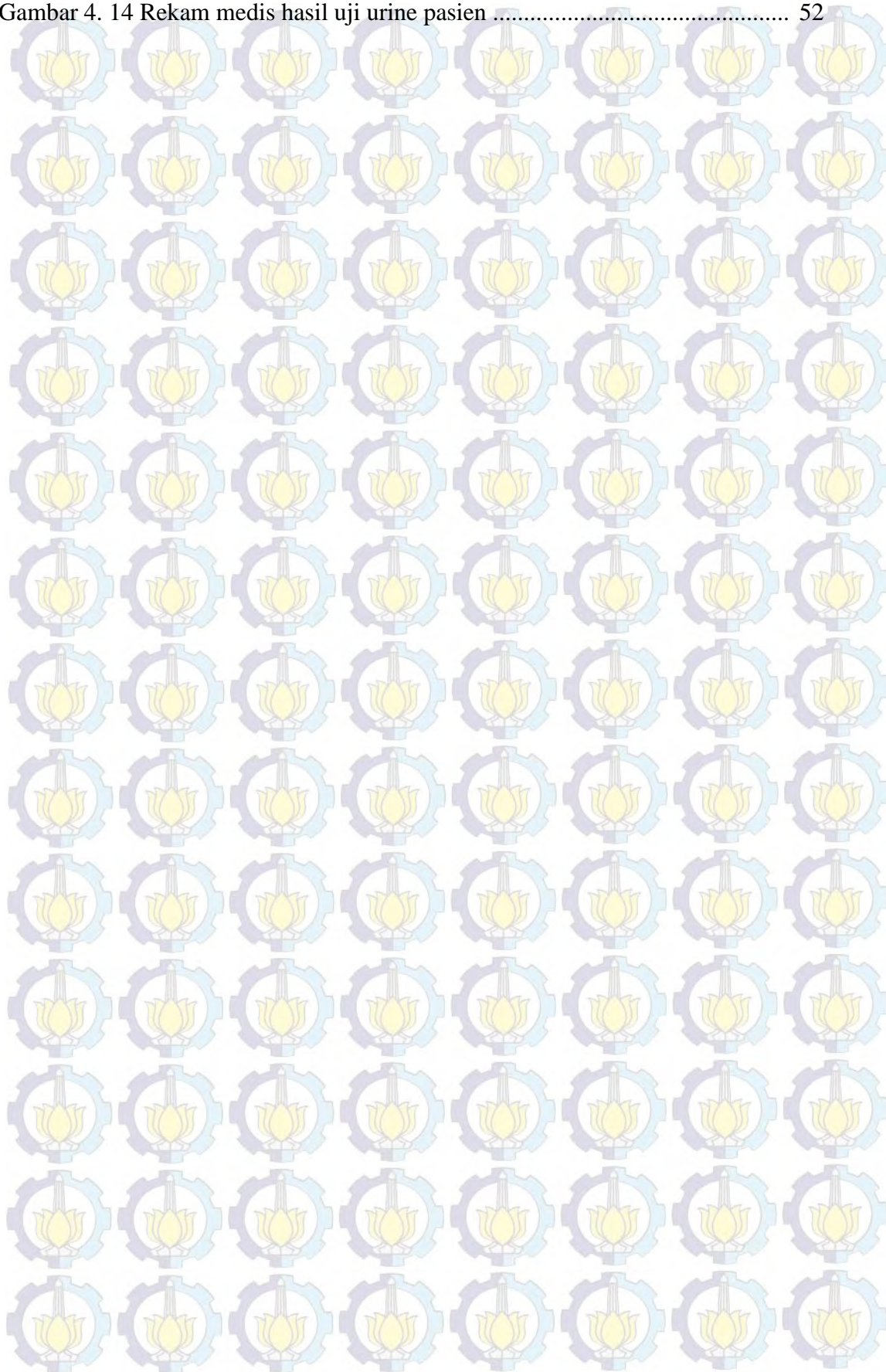


3.5 Analisis hasil.....	28
3.6 Penulisan laporan penelitian .....	29
<b>BAB 4 HASIL DAN PEMBAHASAN .....</b>	<b>31</b>
4.1 Pengumpulan Data.....	31
4.2 Spesifikasi Sistem.....	32
4.3 GSA untuk Klasifikasi Data .....	32
4.4 Perumusan Komputasi Operator Momentum .....	37
4.5 GSA dan Operator Momentum untuk Klasifikasi Data.....	39
4.6 Analisis Hasil Pengujian menggunakan GSA dan GSAM.....	43
4.7 Hasil Rekam Medis Pasien Hiperglikemia .....	51
<b>BAB 5 KESIMPULAN DAN SARAN.....</b>	<b>53</b>
5.1 Kesimpulan .....	53
5.2 Saran .....	53
LAMPIRAN 1 .....	59
LAMPIRAN 2 .....	61
LAMPIRAN 3 .....	63
LAMPIRAN 4 .....	65
LAMPIRAN 5 .....	67
LAMPIRAN 6 .....	69
LAMPIRAN 7 .....	71
LAMPIRAN 8 .....	73
LAMPIRAN 9 .....	75
LAMPIRAN 10 .....	77
LAMPIRAN 11 .....	79
<b>BIOGRAFI PENULIS.....</b>	<b>81</b>

## DAFTAR GAMBAR

Gambar 2. 1 Bagan Warna Acuan Dipstick Urinalysis produk Urit 11G.....	6
Gambar 2. 2 Konsep gaya tarik menarik antar partikel (Rashedi, 2009) .....	8
Gambar 2. 3 Diagram alir GSA (Rashedi, 2009) .....	10
Gambar 2. 4 Hukum Kekekalan Momentum.....	13
Gambar 2. 5 Struktur Neuron (Rojas (1996)) .....	14
Gambar 2. 6 Arsitektur Neural Network (a) ANN Satu Layer (b) ANN Multi Layer (Rojas (1996)) .....	15
Gambar 2. 7 Fungsi Aktivasi (a) Linear (b) Threshold (c) Sigmoid Biner (d) Sigmoid Bipolar (Rojas (1996)).....	15
Gambar 3. 1 Diagram Alir Prosedur Screening Test .....	22
Gambar 3. 2 Arsitektur Jaringan .....	24
Gambar 3. 3 Struktur Agen .....	24
Gambar 3. 4 Bagan alur kerja sistem untuk screening test .....	26
Gambar 3. 5 Contoh rekam medis glukosa dalam urin selama 2 bulan.....	27
Gambar 4. 1 Diagram alir GSAM.....	41
Gambar 4. 2 Perbandingan Kecepatan Konvergensi pada Dataset Wine .....	44
Gambar 4. 3 Perbandingan Kecepatan Konvergensi pada Dataset XOR 3 bit .....	44
Gambar 4. 4 Perbandingan Kecepatan Konvergensi pada Kasus Urinalysis 2 Kelas .....	45
Gambar 4. 5 Perbandingan Kecepatan Konvergensi pada Kasus Urinalysis 3 Kelas .....	45
Gambar 4. 6 Perbandingan hasil klasifikasi pada Dataset Umum .....	46
Gambar 4. 7 Perbandingan hasil klasifikasi pada Kasus Urinalysis 2 Kelas .....	46
Gambar 4. 8 Perbandingan hasil klasifikasi pada Kasus Urinalysis 3 Kelas .....	47
Gambar 4. 9 Perbandingan Hasil Klasifikasi Urin 2 Kelas pada Kelompok Usia .....	49
Gambar 4. 10 Perbandingan Hasil Klasifikasi Urin 3 Kelas pada Kelompok Usia .....	49
Gambar 4. 11 Perbandingan hasil klasifikasi pada Jenis Kelamin 2 kelas .....	49
Gambar 4. 12 Perbandingan hasil klasifikasi pada Jenis Kelamin 3 kelas .....	50
Gambar 4. 13 Perbandingan Waktu Komputasi.....	50

Gambar 4. 14 Rekam medis hasil uji urine pasien ..... 52



## DAFTAR TABEL

Tabel 2. 1 Kadar normal kandungan urin.....	6
Tabel 2. 2 Kriteria diagnosis berdasarkan glukosa darah puasa (NIDDK, 2008)....	7
Tabel 3. 1 Sepuluh kandungan yang terdapat pada urin .....	20
Tabel 3. 2 Nilai hasil konversi .....	20
Tabel 4. 1 Sebaran Kelas Sampel Data .....	31
Tabel 4. 2 Sebaran Usia dan Jenis Kelamin Sampel Data .....	31
Tabel 4. 3 Kompleksitas operator momentum.....	42
Tabel 4. 4 Korelasi fitur dan label.....	48
Tabel 4. 5 Data pasien yang melakukan pemeriksaan lebih dari sekali.....	51
Tabel 4. 6 Rekam medis urine pasien .....	51
Tabel 4. 7 Hasil screen test menggunakan sistem.....	52

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Metode pemeriksaan urinalysis menggunakan *dipstick* untuk mengetahui kandungan kimia yang terdapat pada urin merupakan uji yang mudah dilakukan. Pemeriksaan dilakukan dengan mencelupkan *dipstick* urin ke dalam cairan urin pasien. Dalam hitungan detik, terjadi perubahan warna pada *dipstick* urin yang menginterpretasikan kandungan kimia pada urin tersebut. Pemeriksaan menggunakan *dipstick* urin adalah tahap diagnosis awal untuk kemudian dilakukan pemeriksaan selanjutnya. Pemeriksaan ini umum dilakukan karena memberikan hasil yang cepat dan relatif murah (Whiting, 2006). Interpretasi hasil uji *dipstick* untuk menentukan level kandungan kimia pada urin secara visual dilakukan dengan mencocokkan perubahan warna pada *dipstick* dengan warna yang sesuai pada bagan warna acuan. Dengan menggunakan metode sederhana ini, beberapa kandungan kimia dalam urin yang dapat dianalisis antara lain adalah kandungan darah, protein, glukosa, leukosit, nitrit, dan  $\beta$ -HCG. Sedangkan beberapa kandungan lain juga dapat dianalisis namun jarang dilakukan adalah kandungan keton, urobilinogen, bilirubin, berat jenis, dan pH. Salah satu manfaat pemeriksaan urinalysis menggunakan *dipstick* adalah sebagai alat *screening test* untuk mengetahui seorang pasien menderita suatu penyakit tertentu, misalnya Infeksi Saluran Kemih (ISK), gangguan ginjal, atau Diabetes Mellitus (Barrat, 2007).

Beberapa penelitian terkait *dipstick* urinalysis untuk mengindikasikan suatu penyakit yang diderita pasien telah banyak dilakukan. Pemeriksaan awal menggunakan *dipstick* urin sangat membantu untuk mengetahui apakah pasien perlu melakukan pemeriksaan tambahan. Pugia (2004) telah melakukan diagnosis trypsin inhibitor (urinstatin) dan plasma pasien yang terinfeksi saluran kemih menggunakan *dipstick* urin. Sultana (2001) juga telah meneliti bahwa dengan melakukan pemeriksaan *dipstick* urin, dapat menambah akurasi dari gejala klinis dalam mendiagnosis penyakit pasien. Kombinasi antara uji *dipstick* dan gejala klinis adalah metode yg baik untuk mendiagnosis dibandingkan dengan hanya menggunakan gejala klinis. Selain itu, pemeriksaan *dipstick* dapat pula digunakan untuk *screening*

*test* kesehatan pada bayi (Falakaflaki, 2011). Pemeriksaan menggunakan *dipstick* urin tergolong mudah karena pemeriksaan ini dapat dilakukan secara individu oleh pasien. Akan tetapi kesulitan yang dihadapi adalah interpretasi dari hasil pemeriksaan *dipstick* urin yang masih sulit dipahami oleh para penggunanya.

Di sisi lain, berkembangnya ilmu pengetahuan dan teknologi di bidang analisis data juga telah dimanfaatkan dalam penelitian urinalysis. Calva (2004) telah melakukan klasifikasi sedimen urin menggunakan *Neural Network* dan *fractal geometry*. Klasifikasi dilakukan dengan menggunakan parameter spesifikasi fraktal sehingga dapat menambah performa metode yang digunakan. Studi tentang klasifikasi sedimen urin dan parasit pada urin juga telah dilakukan juga oleh Calva (2009) menggunakan *generalized entropy* dan *Neural Network*. Selain itu, berkembang pula penelitian tentang penerjemahan hasil pemeriksaan *dipstick* urin untuk *screen test* penyakit infeksi saluran kemih yang memanfaatkan metode sistem cerdas oleh Izzah (2013).

Di bidang pengolahan data, diagnosis awal suatu penyakit sering dikaitkan dengan permasalahan klasifikasi. Klasifikasi merupakan proses pelabelan pada suatu dataset berdasarkan pembelajaran pada dataset sebelumnya (Mastrogiannis, 2009). Metode yang digunakan untuk klasifikasi salah satunya adalah *Artificial Neural Network* (ANN). ANN adalah algoritma yang diinspirasi dari sistem syaraf manusia, dimana sistem syaraf manusia terdiri dari beberapa neuron yang menerima rangsangan dari luar tubuh untuk diteruskan menuju otak melalui dendrit. ANN adalah salah satu metode *supervised learning* sehingga membutuhkan algoritma pembelajaran. Algoritma yang digunakan antara lain, *feed forward*, *backpropagation*, *Genetic Algorithm* (GA), *Particle Swarm Optimization* (PSO), dan *Gravitational Search Algorithm* (GSA).

Di sisi lain, berkembangnya algoritma heuristik untuk masalah pencarian solusi ditandai dengan ditemukannya banyak algoritma baru seperti GSA. GSA adalah algoritma yang diinspirasi dari hukum gravitasi dan hukum perpindahan benda menuju posisi seimbang. Sebagai algoritma heuristik, GSA memiliki kemampuan yang bagus dalam pencarian global. Namun jika konvergensi dini terjadi, algoritma ini akan berhenti dan tidak mampu lagi melakukan pencarian. Untuk mengatasi kelemahan ini, Safarazi (2011) pernah menambahkan operator

baru yang disebut “disruption” berdasarkan konsep astrofisika untuk mengatasi kelemahan tersebut dan menunjukkan hasil yang lebih baik. Izzah (2013) juga telah melakukan uji coba pada GSA dengan operator tambahan tersebut pada ANN untuk mengklasifikasi data iris, glass, dan wine juga untuk screen test pada penyakit infeksi saluran kemih. Penelitian ini menunjukkan bahwa dengan adanya operator baru, GSA menunjukkan hasil yang lebih baik daripada GSA sebelumnya dan GA.

Dengan demikian, pada tesis ini akan dirumuskan sebuah operator baru guna mengeksplorasi solusi untuk memperbaiki kemampuan GSA. Operator baru ini diinspirasi dari konsep tumbukan dan hukum kekekalan momentum yang dialami dua benda bergerak. Dengan adanya tumbukan yang terjadi antar solusi, diharapkan diversitas setiap solusi akan terjaga. Operator ini diharapkan mampu mengatasi konvergensi dini karena konsep tumbukan dan hukum kekekalan momentum mengakibatkan dua solusi yang konvergen akan saling menjauh. Selain itu, adanya operator ini diharapkan mampu menemukan solusi global pada setiap iterasi. Selanjutnya, GSA dengan operator momentum ini digunakan sebagai metode pembelajaran untuk ANN. Metode ini digunakan untuk *screening test* hiperglikemia yang merupakan indikator penyakit Diabetes Mellitus. *Screening test* dilakukan berdasarkan hasil pemeriksaan *dipstick* urin berdasarkan konsep klasifikasi data.

## 1.2 Perumusan Masalah

Perumusan masalah dalam tesis ini adalah sebagai berikut:

1. Bagaimana algoritma heuristik yakni GSA dengan operator momentum digunakan sebagai metode pembelajaran untuk ANN?
2. Bagaimana pengaruh operator momentum untuk GSA sebagai metode pembelajaran untuk ANN?
3. Bagaimana membuat model menggunakan pendekatan algoritma heuristik dan ANN untuk melakukan *screening test* pasien yang berpotensi menderita hiperglikemia berdasarkan hasil uji *dipstick* urin?

### 1.3 Batasan Masalah

Dalam tesis ini, batasan masalah yang dibahas diuraikan sebagai berikut:

1. Jaringan *Artificial Neural Network* yang digunakan adalah jaringan dengan dua layer.
2. Sistem disimulasikan pada perangkat lunak MATLAB R2009a.
3. Data yang digunakan dalam penelitian ini adalah data hasil uji *dipstick* urin yang telah dikuantisasi oleh laboratorium.
4. Tumbukan yang terjadi adalah tumbukan lenting sempurna

### 1.4 Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini adalah mengatasi konvergensi dini dengan menambahkan operator momentum pada GSA. Algoritma ini kemudian digunakan sebagai algoritma pembelajaran pada ANN untuk melakukan *screening test* hiperglikemia berdasarkan hasil uji dipstick urin sehingga dapat diketahui potensi pasien menderita Diabetes Mellitus. Manfaat dari penelitian ini adalah sistem dapat digunakan untuk memonitoring kesehatan pasien dengan memanfaatkan dipstick urin.

### 1.5 Kontribusi Penelitian

Kontribusi penelitian ini adalah perumusan operator baru pada GSA yang disebut dengan operator momentum berdasarkan hukum kekekalan momentum dan hukum kekekalan energi kinetik. Operator ini digunakan untuk mengatasi konvergensi dini pada GSA sehingga dapat lebih mengeksplorasi solusi.



## BAB 2

### KAJIAN PUSTAKA

Pada subbab ini, dijelaskan teori yang menjadi dasar pengerjaan tesis. Dasar teori yang diuraikan meliputi pemeriksaan urinalysis, *screening test* pada hiperglikemia, *Gravitational Search Algorithm* dan operator momentum, dan *Artificial Neural Network*.

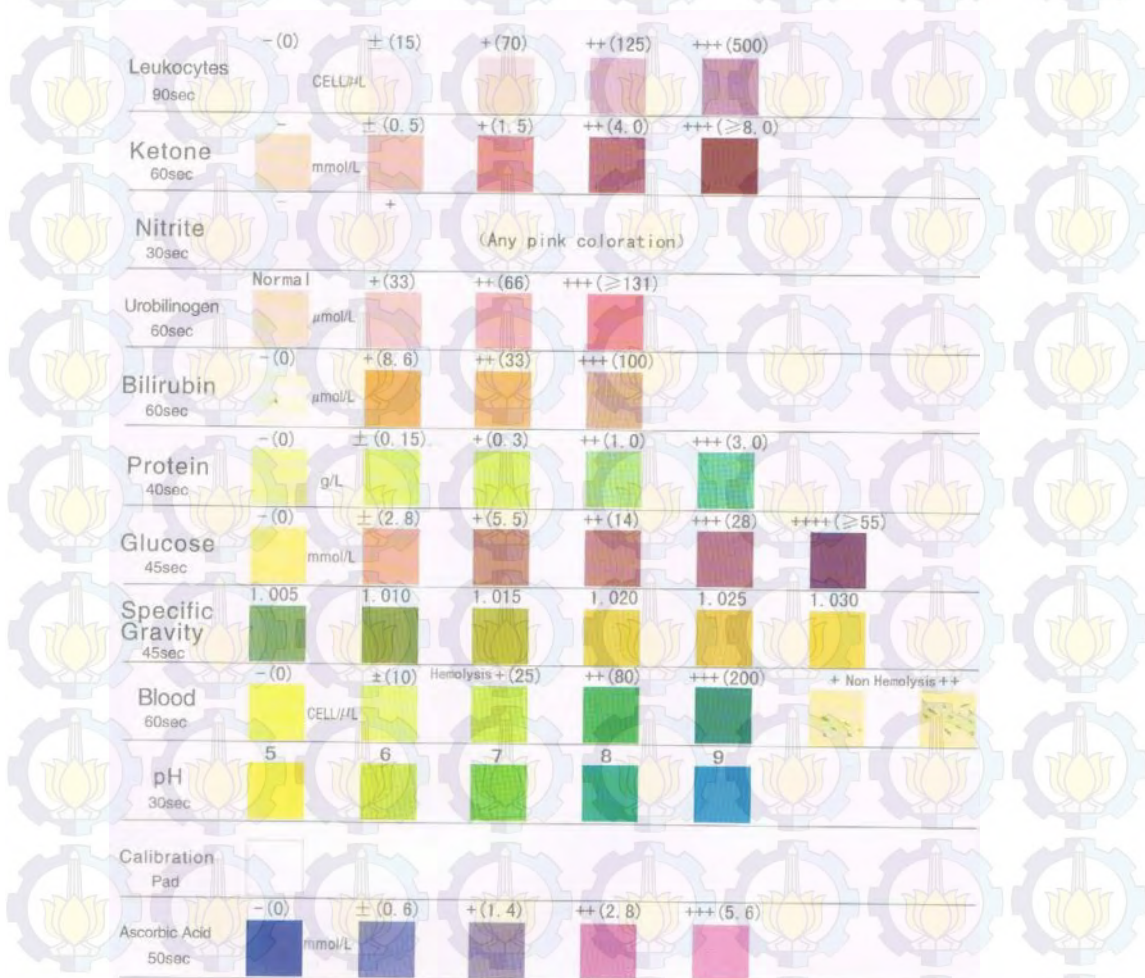
#### 2.1. Pemeriksaan Urinalysis

Urin terbentuk melalui proses filtrasi, reabsorpsi, dan sekresi pada glomerulus, tubulus, dan saluran dalam ginjal. Sebagai zat hasil sekresi, urin mengandung banyak informasi mengenai kandungan biokimia pada tubuh seseorang. Dengan memeriksa 15 sampai 30 mL sampel urin, pemeriksaan urinalysis dapat dilakukan secara rutin untuk *screening test* guna mengetahui kondisi kesehatan pasien. Beberapa penyakit yang diderita pasien menyebabkan adanya abnormalitas pada kandungan urin. Dengan melakukan *screening test*, identifikasi adanya potensi jenis penyakit tertentu dapat dilakukan berdasarkan abnormalitas tersebut (Patel, 2006). Kandungan urin dalam kadar normal dapat dilihat pada Tabel 2.1. Beberapa penyakit yang dapat diketahui melalui *screening test* berdasarkan urin adalah penyakit ginjal, penyakit saluran kemih, hepatitis, atau diabetes mellitus. Pemeriksaan urinalysis meliputi pemeriksaan secara fisik, kimia, analisis mikroskopis, dan analisis makroskopis. Analisis makroskopis dilakukan untuk mengetahui keberadaan beberapa kandungan kimia seperti glukosa, keton, protein, dan darah pada urin. Analisis ini dapat dilakukan dengan menggunakan reagen strip atau dipstick. (Wu, 2010).

Pada pemeriksaan menggunakan dipstick urin, keberadaan kandungan kimia direpresentasikan sebagai perubahan level warna pada dipstick. Perubahan level warna pada dipstick urin dapat dilihat pada Gambar 2.1.

Tabel 2. 1 Kadar normal kandungan urin

Kandungan	Kadar Normal
pH Urine	4,6 - 8,0
Berat Jenis	1.010 - 1.030
Protein	Negatif (-)
Glukosa / reduksi	Negatif (-)
Urobilin	Normal
Bilirubin	Negatif (-)
Keton	Negatif (-)
Nitrit	Negatif (-)
Eritrosit	Negatif (-)
Lekosit	Negatif (-)



Gambar 2. 1 Bagan Warna Acuan Dipstick Urinalysis produk Urit 11G

### **Screening test pada Diabetes Mellitus**

Diabetes adalah suatu penyakit dimana kadar glukosa dalam darah di atas normal. Umumnya setelah makan, tubuh akan memecah makanan menjadi glukosa yang kemudian diedarkan ke sel-sel melalui darah di seluruh tubuh. Namun penderita diabetes memiliki masalah dalam proses perubahan makanan menjadi energi. Tes glukosa darah puasa atau *Fasting Plasma Glucose* (FPG) merupakan tes alternatif untuk mendiagnosis diabetes dengan biaya yang terjangkau. (NIDDK,2008).

Diagnosis penderita hiperglikemia didasarkan pada FPG ketika bernilai 126 mg/dl (7,0 mmol/liter) atau glukosa plasma 2 jam ketika bernilai 200 mg/dl (11,1 mmol /liter) (ADA, 2012). Kondisi diabetes mellitus terjadi jika secara rutin pemeriksaan darah menunjukkan kondisi hipergikemia. Sedangkan pasien dengan tingkat glukosa puasa 100 s.d. 125 mg/dL memiliki kecenderungan pradiabetes yakni gangguan glukosa puasa atau *Impaired Fasting Glucose* (IFG). (NIDDK, 2008). Kriteria diagnosis berdasarkan glukosa darah puasa dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Kriteria diagnosis berdasarkan glukosa darah puasa (NIDDK, 2008)

<b>Glukosa Darah Puasa</b>	<b>Diagnosis</b>
<b>99</b>	Normal
<b>100 s.d. 125</b>	Prediabetes (IFG)
<b>126</b>	Diabetes

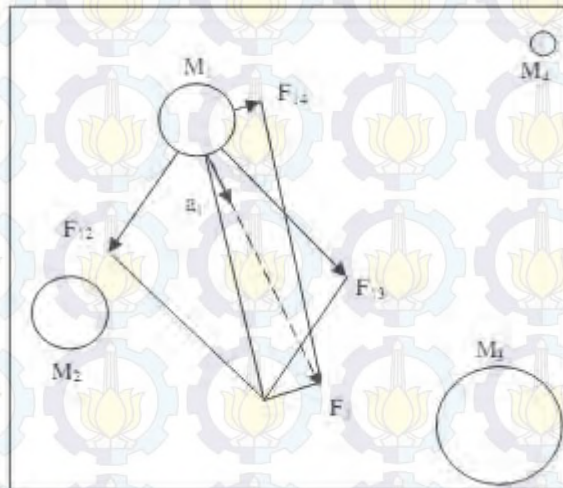
### **Gravitational Search Algorithm dan Hukum Kekekalan Momentum**

Pada bab ini akan dibahas mengenai konsep dasar *Gravitational Search Algorithm* (GSA), prosedur GSA, kelemahan GSA, dan hukum kekekalan momentum.

#### **2.3.1 Konsep Dasar GSA**

Gravitasi adalah kecenderungan sebuah benda untuk melakukan tarik-menarik dengan benda lain yang memiliki massa. Gaya ini adalah salah satu interaksi dasar di alam selain gaya elektromagnetik, gaya nuklir lemah, dan gaya nuklir kuat. Berdasarkan hukum gravitasi Newton, setiap partikel menarik partikel

lain dengan gaya gravitasi sedangkan percepatan partikel ditentukan oleh gaya dan massa partikel tersebut. Konsep gaya tarik menarik antar benda dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Konsep gaya tarik menarik antar partikel (Rashedi, 2009)

Dari Gambar 2.2 terlihat bahwa semakin besar partikel yang ditarik maka semakin besar gaya yang dibutuhkan. Tarikan oleh partikel yang lebih besar ( $M_3$ ) mampu mendominasi resultan gaya yang dialami sebuah partikel sehingga  $F_1$  sebagai gaya resultan yang dialami  $M_1$  akan menuju  $M_3$ . Di sisi lain, jika kedua partikel dipisahkan semakin jauh, maka gaya yang ditimbulkan semakin kecil. Gaya gravitasi inilah yang menyebabkan setiap partikel yang memiliki massa akan menarik satu sama lain sehingga partikel berpindah menuju massa yang lebih besar (Rashedi, 2009).

Hukum Newton menyatakan bahwa gaya gravitasi atau gaya tarik-menarik dapat berlaku secara universal dan sebanding dengan massa masing-masing partikel dan berbanding terbalik dengan kuadrat jarak kedua benda. Secara matematis hal ini dapat ditulis pada persamaan (2.1)

$$F = G \frac{M_1 M_2}{R^2} \quad (2.1)$$

Newton juga mendefinisikan hubungan antara gaya dan percepatan. Hukum II Newton tentang gerak menyatakan gaya sebanding dengan perkalian massa benda

dengan percepatan yang dialami benda tersebut. Secara sederhana dapat dinyatakan pada persamaan (2.2)

$$a = \frac{F}{M} \quad (2.2)$$

dimana

$F$  = besar gaya gravitasi

$G$  = gravitasi konstan

$M_1$  &  $M_2$  = masa partikel 1 dan 2

$R$  = jarak antar dua partikel

$a$  = percepatan

(Tipler, 1998)

### 2.2.2 Prosedur GSA

GSA adalah algoritma heuristik yang diinspirasi dari fenomena gravitasi dari sekumpulan (*swarm*) partikel di alam semesta. Algoritma heuristik adalah teknik pencarian solusi baru disekitar solusi sekarang yang mempertimbangkan *computational cost* walaupun tidak ada jaminan bahwa solusi baru yang diperoleh adalah yang paling optimum (Russell dan Norvig, 1995). Dalam hukum gravitasi, setiap massa memiliki 4 spesifikasi, yaitu posisi, massa inersia, massa gravitasi aktif, dan massa gravitasi pasif. Hukum gravitasi menyatakan bahwa setiap partikel yang memiliki massa menarik satu sama lain dengan gaya gravitasi sehingga menyebabkan perpindahan posisi menuju massa yang lebih besar. Dalam GSA, posisi masa yang disebut dengan agen merepresentasikan solusi permasalahan.

Gambar 2.3 menggambarkan diagram alir GSA.

Secara rinci, Rashedi (2009) menjelaskan langkah langkah GSA sebagai berikut:

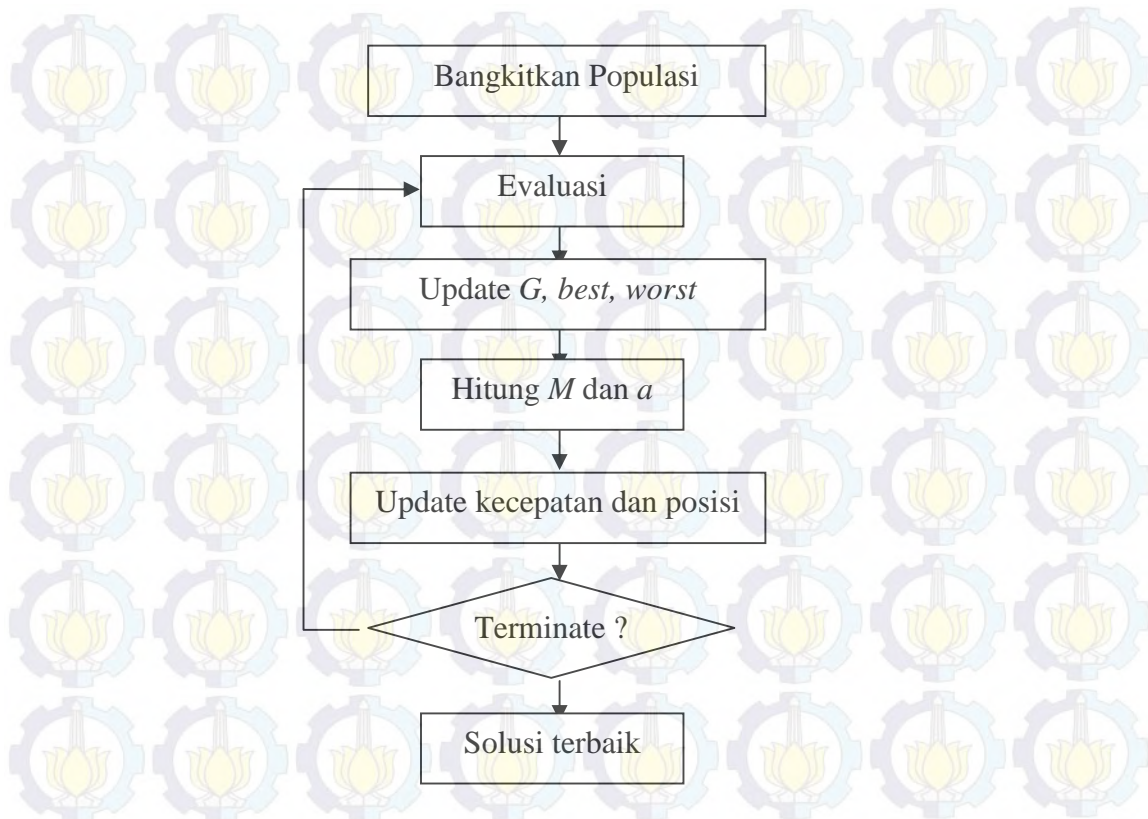
a. Inisialisasi solusi (agen) awal secara random

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^2)$$
 untuk  $i = 1, 2, \dots, N$

$x_i^d$  = posisi agen ke- $i$  pada dimensi ke- $d$

b. Engevaluasi fitness setiap agen

c. Update  $G(t)$ ,  $best(t)$ ,  $worst(t)$  and  $M_i(t)$  for  $i = 1, 2, \dots, N$  menggunakan persamaan (2.3).



Gambar 2. 3 Diagram alir GSA (Rashedi, 2009)

$$G(t) = G(G_0, t) = G(t_0)e^{-\alpha \frac{t}{T}} \quad (2.3)$$

dimana

$G(t)$  = konstanta gravitasi pada saat  $t$

$G(t_0)$  = konstanta gravitasi pada interval kuantum kosmik pada saat  $t_0$

$T$  = maksimum iterasi

Jika fungsi minimasi, maka *best* dan *worst* ditentukan menggunakan persamaan (2.4).

$$best(t) = \min_{j \in \{1, \dots, N\}} fit_j(t); worst(t) = \max_{j \in \{1, \dots, N\}} fit_j(t) \quad (2.4)$$

Jika fungsi maksimasi ditentukan dengan persamaan (2.5).

$$best(t) = \max_{j \in \{1, \dots, N\}} fit_j(t); worst(t) = \min_{j \in \{1, \dots, N\}} fit_j(t) \quad (2.5)$$

dengan

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (2.6)$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)} \quad (2.7)$$

Karena permasalahan dalam tesis ini adalah melakukan klasifikasi yang bertujuan untuk memperkecil nilai *Mean Square Error* (MSE), maka digunakan fungsi minimasi persamaan (2.4).

- d. Menghitung total gaya yang dialami agen berdasarkan persamaan (2.8)

$$F_i^d(t) = \sum_{\substack{j \in KBest \\ j \neq i}}^N rand_j G(t) \frac{M_i(t) M_j(t)}{R_{ij}(t)} (x_i^d(t) - x_j^d(t)) \quad (2.8)$$

$$\text{dengan } R_{ij}(t) = \|X_i(t), X_j(t)\|_2 \quad (2.9)$$

- e. Menghitung kecepatan dan percepatan agen menggunakan persamaan (2.10) dan (2.11)

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} \quad (2.10)$$

$$v_i^d(t+1) = rand_i v_i^d(t) + a_i^d(t) \quad (2.11)$$

- f. Update posisi agen berdasarkan persamaan (2.12)

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (2.12)$$

- g. Ulangi langkah c sampai g sehingga kriteria terpenuhi.

Penelitian sebelumnya menyebutkan bahwa kemampuan pencarian GSA dalam pemecahan fungsi nonlinear memiliki performa yang bagus. Namun ditemukan beberapa kelemahan dalam algoritma ini, yakni ketika banyak agen yang berada dalam batas ruang pencarian akan menyebabkan algoritma ini kehilangan diversitas populasi dan akan tetap berada pada lokal optimum (Yin, 2011). Pada penelitian lain, Sarafrazi (2011) menyatakan bahwa jika konvergensi dini terjadi, algoritma ini tidak mampu lagi melakukan pencarian. Dengan kata lain jika terlalu cepat terjadi konvergensi, algoritma ini akan kehilangan kemampuannya untuk eksplorasi dan menjadi tidak efektif. Yang dimaksud konvergensi dini menurut Nicoara (2009) adalah keadaan dimana ketika iterasi algoritma naik tetapi beberapa solusi secara cepat mendominasi populasi. Dalam keadaan ini, algoritma tidak dapat menghasilkan solusi yang lebih baik lagi. Secara substansial kemampuan algoritma untuk melanjutkan mencari solusi yang lebih baik akan berkurang.

### 2.2.3 Hukum Kekekalan Momentum

Momentum berkaitan dengan kecepatan suatu benda yang bergerak. Momentum didefinisikan sebagai hasil kali antara massa dan kecepatan benda. Secara matematis momentum dapat dihitung dengan persamaan (2.13)

$$p = mv \quad (2.13)$$

dimana

$P$  = momentum

$m$  = massa benda

$v$  = kecepatan benda

Jika sebuah gaya bekerja pada sebuah benda dalam suatu selang waktu sehingga terjadi perubahan kecepatan, maka momentum benda tersebut akan berubah. Perubahan momentum yang dialami suatu benda disebut dengan impuls. Berdasarkan Hukum II Newton dan definisi percepatan, maka impuls dapat dihitung menggunakan persamaan (2.14).

$$I = F\Delta t = mv_t - mv_0 \quad (2.14)$$

dimana

$I$  = Impuls suatu benda

$F$  = Gaya yang dialami benda

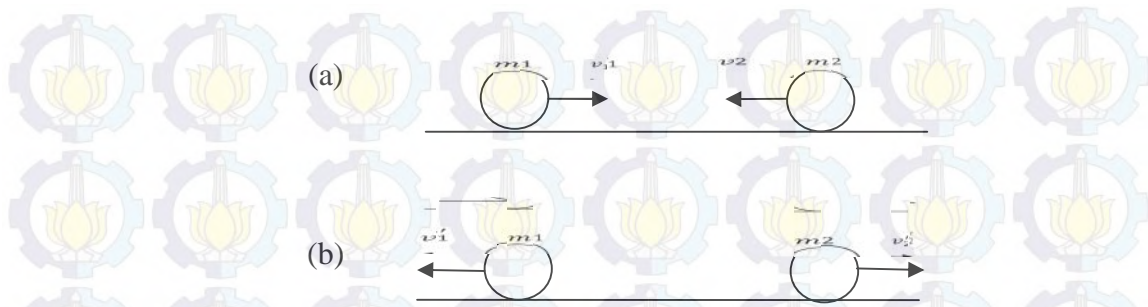
$\Delta t$  = Perubahan waktu

$v_0$  = Kecepatan benda pada posisi awal

$v_t$  = Kecepatan benda pada saat  $t$

Proses tumbukan dua benda dapat dilihat pada Gambar 2.4. Pada Gambar 2.4(a) dapat dilihat bahwa dua benda bergerak yang saling mendekat dengan kecepatan  $v_1$  dan  $v_2$ . Kemudian kedua benda akan bertumbukan sehingga benda  $m_1$  dan  $m_2$  akan berbalik dengan kecepatan masing-masing  $v'_1$  dan  $v'_2$  seperti terlihat pada Gambar 2.4(b).





Gambar 2. 4 Hukum Kekekalan Momentum

Pada peristiwa tumbukan akan berlaku hukum kekekalan momentum. Hukum kekekalan momentum menyatakan bahwa momentum kedua benda sebelum tumbukan sama dengan momentum kedua benda setelah tumbukan. Secara matematis hukum kekekalan momentum dapat ditulis menggunakan persamaan (2.15)

$$m_1 v_1 + m_2 v_2 = m_1 v_1' + m_2 v_2' \quad (2.15)$$

Secara umum terdapat beberapa jenis tumbukan, yakni tumbukan lenting sempurna, tumbukan lenting sebagian, dan tumbukan tidak lenting sama sekali.

1) Tumbukan Lenting Sempurna

Pada tumbukan lenting sempurna, momentum dan energi kinetik kedua benda sebelum dan setelah tumbukan adalah sama. Dengan kata lain, pada tumbukan lenting sempurna berlaku hukum kekekalan momentum dan hukum kekekalan energi kinetik. Hukum kekekalan energi kinetik dirumuskan pada persamaan (2.16)

$$\frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 = \frac{1}{2} m_1 v_1'^2 + \frac{1}{2} m_2 v_2'^2 \quad (2.16)$$

2) Tumbukan Lenting Sebagian

Pada tumbukan lenting sebagian, hukum kekekalan energi kinetik tidak berlaku karena terjadi perubahan energi kinetik sebelum dan sesudah tumbukan. Pada tumbukan lenting sebagian hanya berlaku hukum kekekalan momentum.

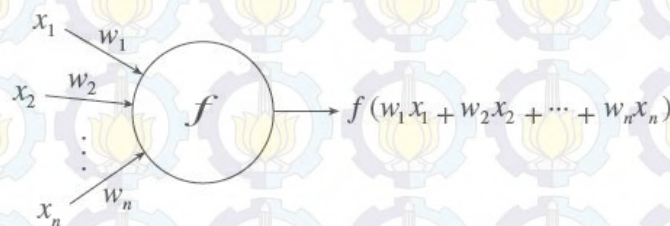
3) Tumbukan Tidak Lenting Sama Sekali

Tumbukan antara dua buah benda dikatakan tidak lenting sama sekali sesudah tumbukan kedua benda menjadi satu (bergabung), sehingga kedua benda memiliki kecepatan sama.

(Tipler, 1998)

## Artificial Neural Network

*Artificial Neural Network* (ANN) merupakan teknik pemodelan yang diinspirasi dari kemampuan sistem saraf bertukar informasi. Pemodelan informasi ini dirancang dengan menggunakan jaringan syaraf tiruan seperti neuron dan sinapsis. Dalam arsitektur ANN, sebuah node adalah suatu elemen komputasi yang digambarkan sebagai suatu neuron. Informasi yang dikandung dalam node akan dilanjutkan ke node yang lain layaknya sinapsis pada jaringan syaraf manusia. Gambar 2.5 menunjukkan struktur dari sebuah neuron dengan  $n$  input dimana masing-masing input  $i$  mengirimkan sebuah nilai  $x_i$  dengan bobot  $w_i$  (Rojas, 1996).

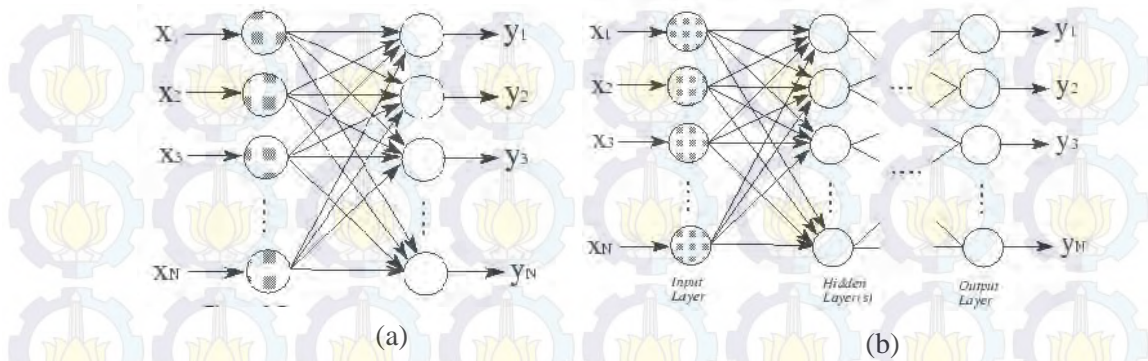


Gambar 2. 5 Struktur Neuron (Rojas (1996))

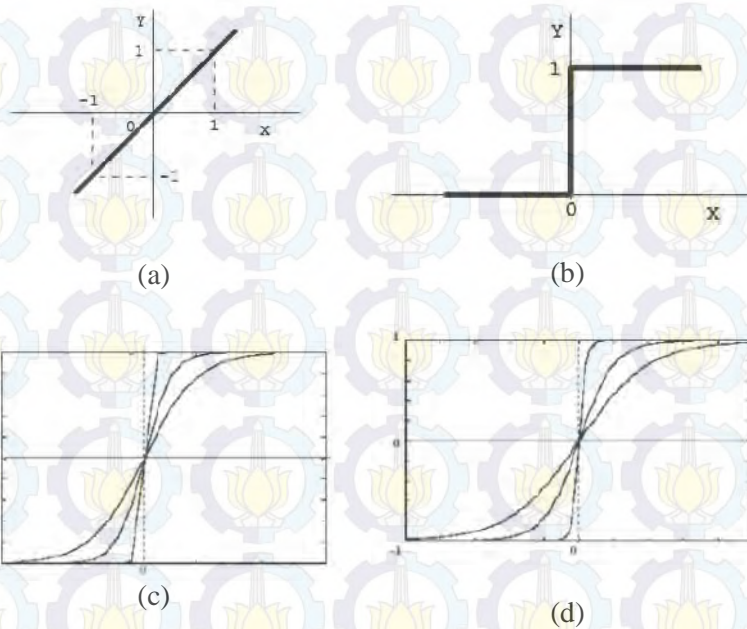
Arsitektur lapisan pada ANN terdiri dari lapisan input, lapisan tersembunyi, dan lapisan output. Arsitektur ANN tanpa lapisan tersembunyi disebut dengan ANN satu layer, sedangkan arsitektur ANN yang memiliki lapisan tersembunyi disebut dengan ANN multi layer. Masing-masing node antar lapisan dihubungkan dengan bobot  $w_i$  dan dipengaruhi oleh bias  $\theta_i$ . Kemampuan pemodelan ANN ditentukan oleh 3 hal (Wijaya, 2013), yakni:

1. Pola hubungan antar neuron (disebut arsitektur jaringan),
2. Metode penentuan bobot (disebut metode *training/learning*),
3. Fungsi aktivasi (fungsi transfer).

Contoh arsitektur jaringan ANN dapat dilihat pada Gambar 2.6. Sedangkan macam-macam fungsi aktivasi dapat dilihat pada Gambar 2.7.



Gambar 2. 6 Arsitektur Neural Network (a) ANN Satu Layer (b) ANN Multi Layer (Rojas (1996))



Gambar 2. 7 Fungsi Aktivasi (a) Linear (b) Threshold (c) Sigmoid Biner (d) Sigmoid Bipolar (Rojas (1996))

Metode pembelajaran pada ANN bertujuan untuk menentukan bobot dan bias yang optimal. Bobot dan bias ini dilakukan secara berulang sebanyak iterasi. Salah satu metode inisialisasi bobot dan bias adalah metode Nguyen Widrow sebagai berikut:

1. Menentukan besarnya skala berdasarkan persamaan (2.17)

$$\beta = 0.7p^n \quad (2.17)$$

dimana

$p$  = jumlah node hidden

$n$  = jumlah node input.

2. Inisialisasi bobot  $w_{ij}$  secara random dengan nilai inisialisasi  $w_{ij}$  adalah  $-0.5$

$$w_{ij} \leq 0.5.$$

3. Menghitung besarnya magnitude bobot  $w_{ij}$  berdasarkan persamaan (2.18)

$$\|w_j\| = \sqrt{\sum_{i=1}^p w_{ij}^2} \quad (2.18)$$

4. Mengupdate bobot  $w_{ij}$  berdasarkan persamaan (2.19)

$$w_{ij} = \frac{\beta w_{ij}}{w_j} \quad (2.18)$$

5. Membangkitkan nilai bias  $\theta_j$  sebesar  $-\beta \leq \theta_j \leq \beta$

(Pavelka, 2004)

### GSA sebagai Pembelajaran pada ANN

Algoritma pembelajaran adalah metode adaptif yang digunakan sebagai alat komputasi antara input-output pada jaringan ANN. Pengoreksian yang dilakukan secara iteratif bertujuan agar jaringan ANN mampu menghasilkan output yang diinginkan (Rojas, 1996). Algoritma evolusi dan optimasi seperti GA, PSO, dan GSA dapat digunakan sebagai algoritma pembelajaran untuk menemukan solusi bobot dan bias yang optimum.

#### 2.5.1 Pengkodean Solusi

Dalam penentuan bobot dan bias yang optimum, bobot dan bias dikodekan dalam suatu untaian solusi. Marjalili (2012) mengkodekan solusi ANN sebagai sebuah matriks (array 2D) dengan elemen vektor kolom berupa bobot dan bias pada layer satu sampai dengan  $n$ . Dengan demikian dalam suatu populasi, sekumpulan solusi tersebut disimpan dalam array 3D. Misalkan untuk jaringan dengan 2 input dengan 3 node hidden, matriks solusi dapat direpresentasikan seperti pada persamaan (2.19) dan (2.20).

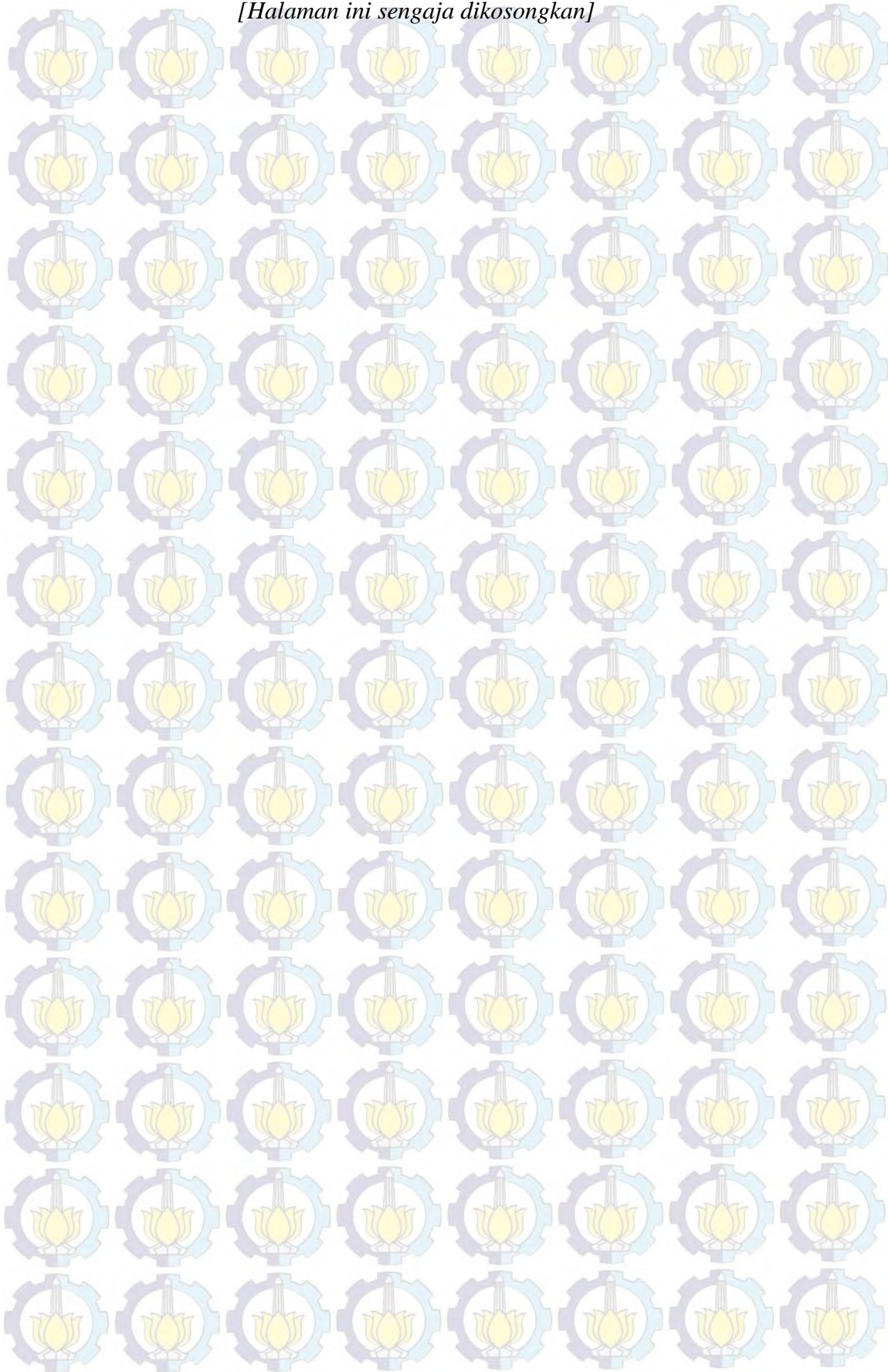
$$\text{solusi}(:, :, i) = [\text{Bobot}_1, \text{Bias}_1, \dots, \text{Bobot}_n, \text{Bias}_n] \quad (2.19)$$

$$\text{Bobot}_1 = \begin{bmatrix} b_{11} & b_{21} \\ b_{12} & b_{22} \\ b_{13} & b_{23} \end{bmatrix}; \text{Bias}_1 = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \quad (2.20)$$

#### 2.4.2 Fungsi Tujuan dan Fungsi Fitness

Fungsi tujuan yang digunakan merupakan fungsi aktivasi dari arsitektur ANN. Fungsi aktivasi yang dapat digunakan antara lain fungsi linear dan fungsi sigmoid. Sedangkan fungsi fitness yang digunakan adalah fungsi yang digunakan untuk menghitung kesalahan target. Fungsi fitness yang dapat digunakan antara lain *Mean Square Error (MSE)*, *Root Mean Square Error (RMSE)*, atau *Sum Square Error (SSE)* (Marjalili, 2012).

*[Halaman ini sengaja dikosongkan]*



## BAB 3

### METODE PENELITIAN

Secara umum, penelitian ini diawali dengan studi literatur, pengumpulan data, desain sistem, pengujian sistem, analisis hasil, dan penyusunan laporan. Secara lebih detail, penelitian ini dirancang dengan urutan sebagai berikut:

#### 3.1. Studi literatur

Dalam studi literatur, dikaji berbagai referensi yang berkaitan dengan uji urinalysis, proses *screening test* penyakit hiperglikemia, klasifikasi pada data mining, *Gravitational Search Algorithm* (GSA), konsep hukum kekekalan momentum, dan *Artificial Neural Network* (ANN). Kemudian dicari isu yang sedang dihadapi sekaligus solusi pada kasus yang akan diteliti. Dalam tesis ini isu yang diangkat adalah masih banyaknya pengguna yang belum mampu mengintepretasikan hasil uji urinalysis. Selanjutnya ditentukan metode untuk pemecahan solusi tersebut. Dari metode yang akan digunakan ditemukan kekurangan dan kelebihan dari masing-masing metode. Dalam tesis ini, ditemukan kelemahan dari algoritma heuristik GSA yang kemudian akan diatasi oleh operator momentum. Selanjutnya, pendekatan dua metode ini digunakan sebagai algoritma pembelajaran pada ANN.

#### 3.2. Pengumpulan data

Pengumpulan data hasil uji urinalysis diperoleh dari Laboratorium Klinik Popular, Jln. Bendul Merisi no. 12, Surabaya. Data yang diambil adalah data pasien dimulai pada bulan Januari 2012 s.d. Maret 2013. Data tersebut terdiri dari hasil uji glukosa darah puasa dan uji urinalysis menggunakan dipstick urin yang digunakan sebagai fitur yang diteliti. Hasil uji glukosa darah puasa digunakan untuk menentukan label hiperglikemia dimana kadar normal glukosa darah puasa seseorang adalah  $<126$ . Sepuluh kandungan hasil uji urinalysis menggunakan dipstick urin yang telah dikuantisasi oleh laboratorium dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Sepuluh kandungan yang terdapat pada urin

Kandungan	Level					
	1	2	3	4	5	6
<b>pH</b>	5	6	7	8	9	
<b>Berat Jenis</b>	1.005	1.010	1.015	1.020	1.025	1.030
<b>Protein</b>	Negatif	Positif	Positif 1+	Positif 2+	Positif 3+	
<b>Glukosa</b>	Negatif	Positif	Positif 1+	Positif 2+	Positif 3+	Positif 4+
<b>Urobilin</b>	Normal	Positif	Positif 1+	Positif 2+		
<b>Bilirubin</b>	Negatif	Positif	Positif 1+	Positif 2+		
<b>Keton</b>	Negatif	Positif	Positif 1+	Positif 2+	Positif 3+	
<b>Nitrit</b>	Negatif	Positif				
<b>Eritrosit</b>	Negatif	Positif	Positif 1+	Positif 2+	Positif 3+	
<b>Lekosit</b>	Negatif	Positif	Positif 1+	Positif 2+	Positif 3+	

Untuk melakukan klasifikasi, dilakukan konversi pada fitur yang masih berupa level zat kandungan menjadi bilangan numerik. Level 1 untuk fitur non numerik akan dikonversikan menjadi nilai 0. Selanjutnya, naiknya level kandungan akan diikuti bertambahnya nilai konversi. Hasil konversi tersebut dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Nilai hasil konversi

Kandungan	Level					
	1	2	3	4	5	6
<b>pH</b>	5	6	7	8	9	
<b>Berat Jenis</b>	1.005	1.010	1.015	1.020	1.025	1.030
<b>Protein</b>	0	1	2	3	4	
<b>Glukosa</b>	0	1	2	3	4	5
<b>Urobilin</b>	0	1	2	3		
<b>Bilirubin</b>	0	1	2	3		
<b>Keton</b>	0	1	2	3	4	
<b>Nitrit</b>	0	1				
<b>Eritrosit</b>	0	1	2	3	4	
<b>Lekosit</b>	0	1	2	3	4	



### 3.3.Desain sistem

Tahap awal desain sistem adalah merumuskan kontribusi utama penelitian ini. Dari studi literatur yang telah dilakukan, diketahui bahwa GSA merupakan algoritma pencarian yang memiliki performa yang bagus. Namun, ketika konvergensi dini terjadi, GSA menjadi lambat dan terhenti. Oleh karena itu dilakukan penambahan operator momentum sebagai operator yang mengatasi kelemahan ini (untuk selanjutnya GSA dengan adanya penambahan operator momentum disebut dengan GSA dan Momentum atau GSAM). Kemudian algoritma ini digunakan sebagai metode pembelajaran pada ANN guna menentukan kombinasi bobot dan bias. Desain sistem yang diusulkan pada tesis ini (dapat dilihat pada Gambar 3.1) terdiri dari beberapa tahap yakni sebagai berikut:

1) Pelabelan data hasil uji urinalysis pasien

Pada proses klasifikasi, dibutuhkan data hasil uji urinalysis berlabel yang digunakan sebagai data latih. Pelabelan dilakukan berdasarkan hasil uji glukosa darah puasa atau *Fasting Plasma Glukose* (FPG).

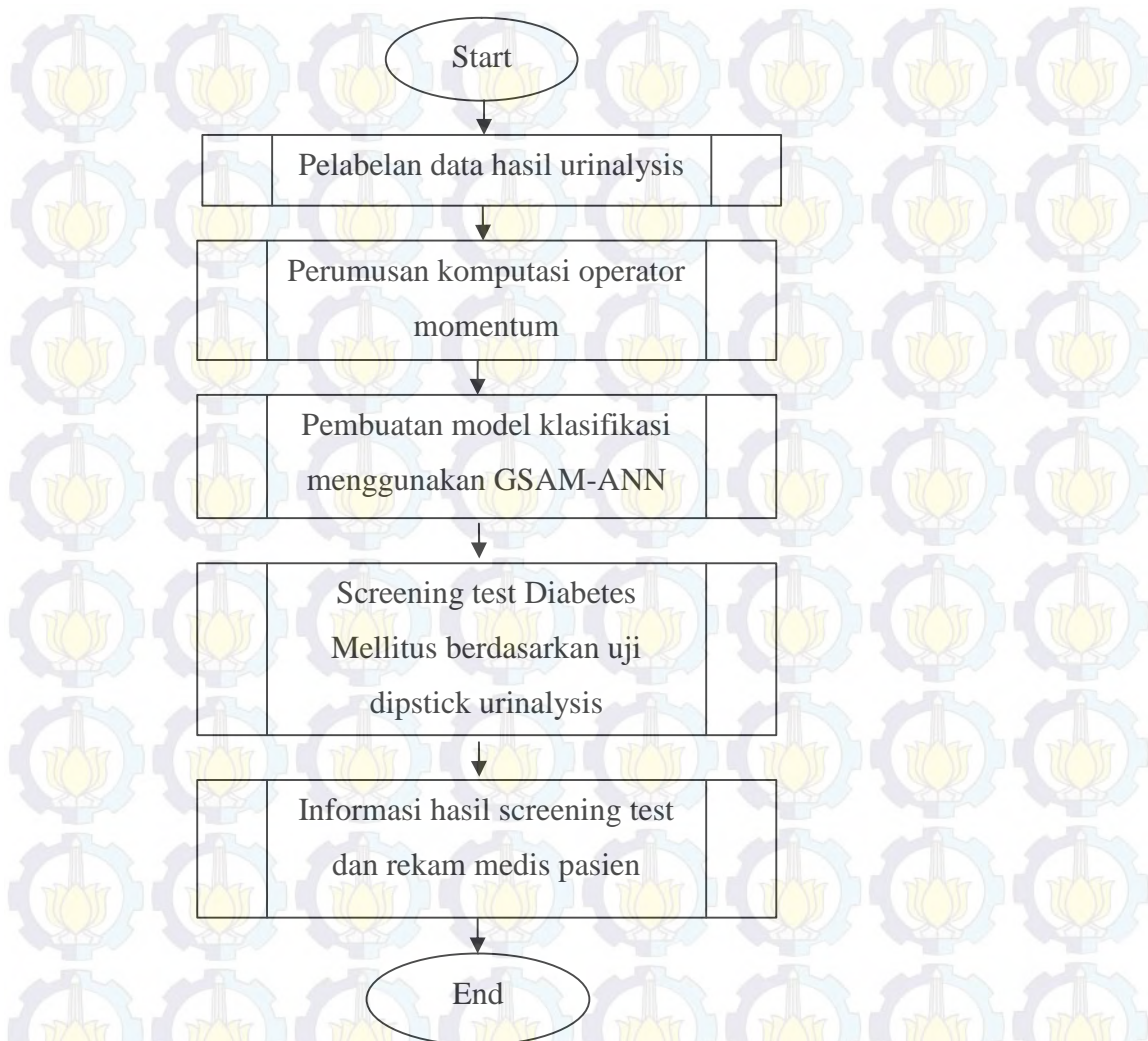
Oleh karena itu dibutuhkan suatu rule untuk menentukan label dari suatu *record* data latih. Berdasarkan NIDDK yang telah dijelaskan pada Tabel 2.2, rule yang dapat dibangkitkan adalah sebagai berikut:

(i) Rule 1

**IF FPG  $\geq$  125 THEN LABEL = “NEGATIF HIPERGLIKEMIA”**  
**ELSE LABEL = “POSITIF HIPERGLIKEMIA”**

(ii) Rule 2

**IF FPG  $\geq$  99 THEN LABEL = “NEGATIF HIPERGLIKEMIA”**  
**ELSE IF FPG  $\geq$  125 THEN LABEL = “PRE HIPERGLIKEMIA”**  
**ELSE LABEL = “HIPERGLIKEMIA”**



Gambar 3. 1 Diagram Alir Prosedur Screening Test

## 2) Perumusan operator momentum

Pada tesis ini, konsep hukum kekekalan momentum dan hukum kekekalan energi kinetik pada persamaan (2.15) dan (2.16) akan digunakan secara komputasi untuk mengatasi kelemahan konvergensi dini pada GSA. Asumsi yang digunakan dalam pemodelan komputasi ini adalah sebagai berikut:

- (i) Dua agen akan mengalami tumbukan jika berada pada suatu threshold berupa jarak tertentu.
- (ii) Tumbukan yang terjadi adalah tumbukan lenting sempurna sehingga berlaku hukum kekekalan momentum dan hukum kekekalan energi kinetik.
- (iii) Massa agen sebelum dan sesudah tumbukan adalah sama.

Selain itu, variabel-variabel pada hukum kekekalan momentum dan hukum kekekalan energi kinetik akan dikodekan sebagai berikut:

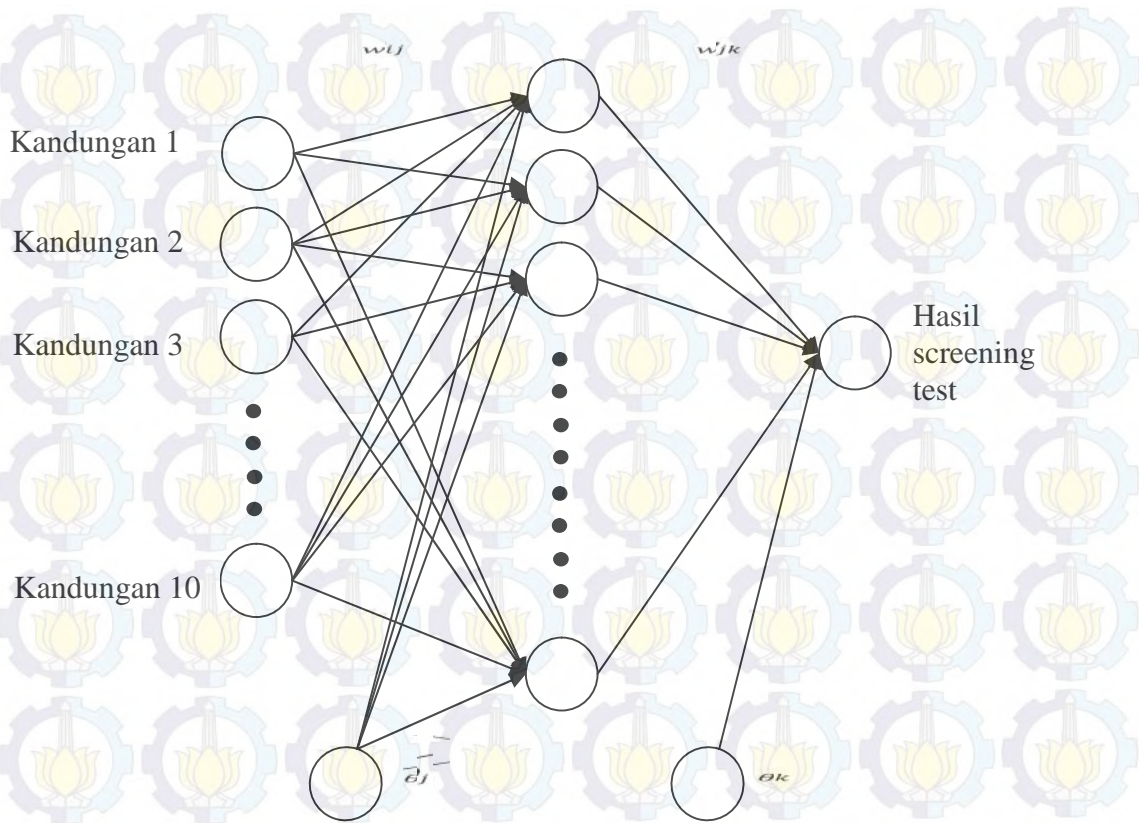
- (i)  $m_i$  akan merepresentasikan massa suatu agen ke- $i$
- (ii)  $v_i$  akan merepresentasikan kecepatan suatu agen ke- $i$
- (iii)  $v'_i$  akan merepresentasikan kecepatan suatu agen ke- $i$  setelah terjadi tumbukan

Dengan konsep ini, diharapkan jika dua agen yang saling mendekat dan terletak pada suatu ambang batas berupa jarak tertentu, maka dua agen tersebut akan bertumbukan sehingga mengalami perpindahan posisi dan menyebar. Dengan demikian, kemungkinan terjadinya konvergensi dini akan dapat diatasi.

### 3) Membangun model klasifikasi menggunakan GSAM-ANN

Untuk membentuk suatu model klasifikasi, digunakan struktur jaringan ANN dua lapis. Model klasifikasi yang dihasilkan dari ANN berupa kombinasi bobot dan bias. Kombinasi bobot dan bias tersebut yang akan mempengaruhi input sehingga diperoleh output berupa label suatu *record*. Struktur jaringan yang digunakan adalah jaringan dua layer dengan sepuluh node input,  $n$  node tersembunyi, dan satu node output. Antara node input ke- $i$  dengan node tersembunyi ke- $j$  dihubungkan oleh bobot  $w_{ij}$  dan dipengaruhi oleh bias  $\theta_j$ . Sedangkan node tersembunyi ke- $j$  dengan node output ke- $k$  dihubungkan oleh bobot  $w_{jk}$  dan dipengaruhi oleh bias  $\theta_k$ . Struktur jaringan yang digunakan dapat dilihat pada Gambar 3.2.

Untuk inialisasi agen, bobot dan bias dibangun menggunakan metode Nguyen Windrow. Kemudian kombinasi bobot dan bias tersebut dilatih secara berulang menggunakan GSAM untuk ditentukan bobot dan bias yang paling optimal. Untuk menentukan kombinasi tersebut perlu dilakukan pengkodean agen. Pengkodean agen yang merepresentasikan suatu solusi dapat dilihat pada Gambar 3.3.



Gambar 3. 2 Arsitektur Jaringan



Gambar 3. 3 Struktur Agen

Sejumlah agen yang dibangkitkan kemudian akan dilatih untuk memperoleh agen yang optimum. Pelatihan tersebut dilakukan menggunakan GSAM dengan prosedur sebagai berikut:

- Membangkitkan sejumlah agen  $X_i = [W_1 B_1 W_2 B_2]$  berupa kombinasi bobot dan bias dengan struktur seperti pada Gambar 3.3. Selanjutnya tetapkan nilai massa  $M_i(t)$ , percepatan  $a_i(t)$ , dan kecepatan  $v_i(t)$  adalah 0.
- Mengevaluasi agen dengan menghitung fungsi aktivasi pada lapisan tersembunyi menggunakan fungsi sigmoid sesuai dengan persamaan (3.1).

$$f(s_j) = \frac{1}{1 + e^{-\sum(w_{ij}x_{ij} - \theta_j)}}; \quad (3.1)$$

dimana  $s_j = \sum(w_{ij}x_{ij} - \theta_j)$ ,  $j = 1, 2, \dots, h$ ,  $n$  adalah jumlah node input,  $w_{ij}$  adalah bobot dari node  $i$  ke  $j$ ,  $x_{ij}$  adalah data instance ke  $i$  fitur ke  $j$ ,  $\theta_j$  adalah bias node ke- $j$ . Kemudian menghitung *Minimum Square Error* (MSE) sebagai fungsi fitness berdasarkan persamaan (3.2).

$$MSE = \sum_{i=1}^q \frac{(o_i - d_i)^2}{q} \quad (3.2)$$

dimana  $q$  adalah jumlah data training,  $o_i$  adalah output label data training ke  $i$ , dan  $d_i$  adalah target label data training ke  $i$ .

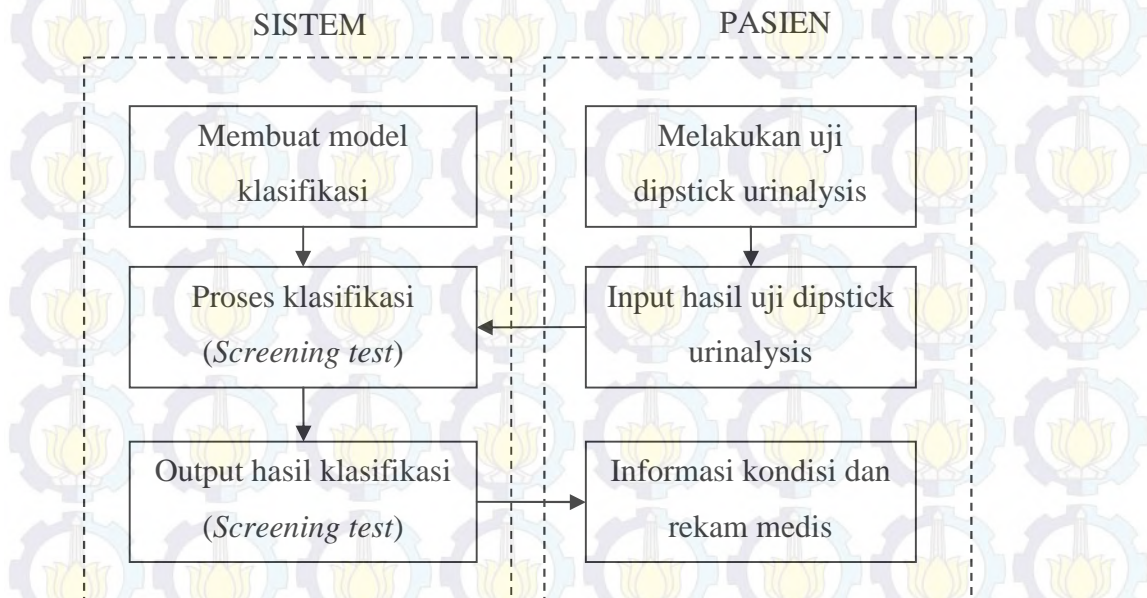
- c. Menghitung jarak  $R_{ij}(t)$  untuk setiap agen  $X_i$  dan  $X_j$  dan gaya total  $F_i^d(t)$  yang dialami oleh setiap agen  $X_i$  berdasarkan persamaan (2.8) dan (2.9). Kemudian update nilai  $G(t)$  berdasarkan persamaan (2.3).
- d. Menentukan agen terbaik  $best(t)$  dan agen terburuk  $worst(t)$ . Karena kasus ini adalah kasus minimasi maka  $best(t)$  dipilih dari agen yang memiliki fungsi fitness terendah, sedangkan  $worst(t)$  dipilih dari agen yang memiliki fungsi fitness tertinggi berdasarkan persamaan (2.4). Kemudian update massa agen  $M_i(t)$  berdasarkan persamaan (2.7).
- e. Update percepatan agen  $a_i^d(t)$  dan kecepatan agen  $v_i^d(t + 1)$  berdasarkan persamaan (2.10) dan (2.11).
- f. Menghitung kecepatan baru agen  $v_i^d(t + 1)$  jika kemungkinan terjadi tumbukan.
- g. Menentukan posisi baru agen  $x_i^d(t + 1)$  berdasarkan persamaan (2.12).

Prosedur ini dilakukan sampai kondisi konvergen terpenuhi. Konvergensi terjadi dengan kriteria jika perulangan dilakukan sebanyak  $N$  iterasi atau nilai dari  $MSE < 0.05$ . Kombinasi bobot dan bias optimum ini kemudian dijadikan sebagai solusi akhir guna pembobotan fitur.

#### 4) *Screening test* hiperglikemia berdasarkan uji dipstick urinalysis

Pada saat melakukan *screening test*, pasien memasukkan hasil uji urinalysis berdasarkan dipstick urin ke dalam sistem. Dari hasil tersebut pasien akan

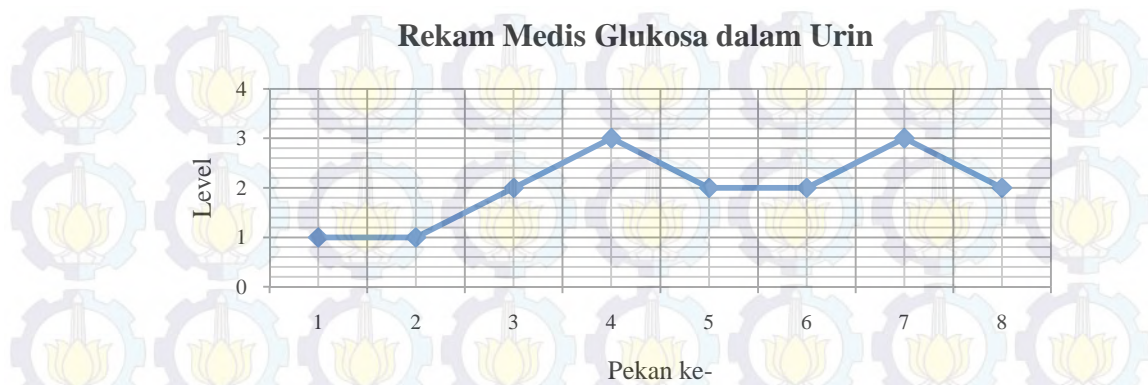
mendapatkan informasi berupa keputusan kondisi kesehatan dan rekam medis uji urinalysis (jika pasien telah menggunakan sistem ini sebelumnya). Bagan alur kerja sistem untuk *screening test* dapat dilihat pada Gambar 3.4.



Gambar 3. 4 Bagan alur kerja sistem untuk screening test

5) Informasi hasil *screening test* dan rekam medis pasien

Informasi yang akan diperoleh dari sistem ini adalah hasil *screening test* berupa kondisi “NEGATIF HIPERGLIKEMIA” dan “POSITIF HIPERGLIKEMIA” atau “NORMAL”, “PREHIPERGLIKEMIA”, dan “HIPERGLIKEMIA”. Sedangkan rekam medis yang dimaksud adalah jika pasien telah melakukan *screening test* secara berulang, akan ditampilkan rekam medis kondisi urin pada beberapa waktu sebelumnya. Contoh hasil rekam medis yang diperoleh jika pasien telah melakukan *screening test* selama 2 bulan dapat dilihat pada Gambar 3.5.



Gambar 3. 5 Contoh rekam medis glukosa dalam urin selama 2 bulan

### 3.4. Pengujian sistem

Untuk menguji apakah metode yang diajukan bisa berjalan dengan baik, perlu dilakukan beberapa kali uji coba. Uji coba sistem diimplementasikan pada MATLAB R2009a. Ada beberapa kombinasi skenario pengujian yang akan dilakukan. Pengujian dilakukan pada masing-masing rule pelabelan dengan arsitektur jaringan  $10 - n - 1$ , dengan  $n = 7, 11, \text{ dan } 15$ . Skenario pengujian tersebut adalah sebagai berikut:

#### 1) Skenario 1

Pengujian pada skenario ini bertujuan untuk menguji metode yang diusulkan pada kasus klasifikasi umum. Pengujian terdiri dari pengambilan data sampel menggunakan metode 5 fold cross validation. Pada skenario ini dihitung pula waktu komputasi dari metode yang diusulkan.

#### 2) Skenario 2

Pengujian pada skenario ini bertujuan untuk menguji metode yang diusulkan pada kasus klasifikasi urinalysis. Pengujian terdiri dari pengambilan data sampel menggunakan metode  $k$  fold cross validation, dengan  $k = 3, 5, \text{ dan } 10$ .

### 3) Skenario 3

Pengujian pada skenario ini bertujuan untuk mengetahui kemampuan metode untuk melakukan *screening* pada kasus-kasus khusus. Pengujian ini terdiri dari :

Pengujian 1 : Pengujian pada dataset dengan pengelompokan faktor usia dengan kriteria:

- 1) Kelompok usia < 20 tahun
- 2) Kelompok usia 20 tahun s.d 50 tahun
- 3) Kelompok usia > 50 tahun

Pengujian 2 : Pengujian pada dataset dengan pengelompokan faktor jenis kelamin.

Pengujian 3 : Pengujian pada dataset yang berlabel positif hiperglikemia dengan tujuan memonitori kondisi pasien

### 3.5. Analisis hasil

Hasil uji coba akan dievaluasi sehingga dapat dilihat kinerja metode yang diajukan. Ukuran evaluasi yang digunakan adalah akurasi, sensitivitas, dan spesifisitas sistem. Akurasi adalah kemampuan sistem untuk mengklasifikasi dengan benar. Akurasi dapat dihitung menggunakan persamaan (3.3). Sensitivitas adalah kemampuan sistem untuk memprediksi pasien positif sakit dari seluruh pasien yang sakit. Sensitivitas dapat dihitung menggunakan persamaan (3.4). Spesifisitas adalah kemampuan sistem untuk memprediksi pasien sehat dari seluruh pasien yang sehat. Spesifisitas dapat dihitung menggunakan persamaan (3.5).

$$\text{akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.3)$$

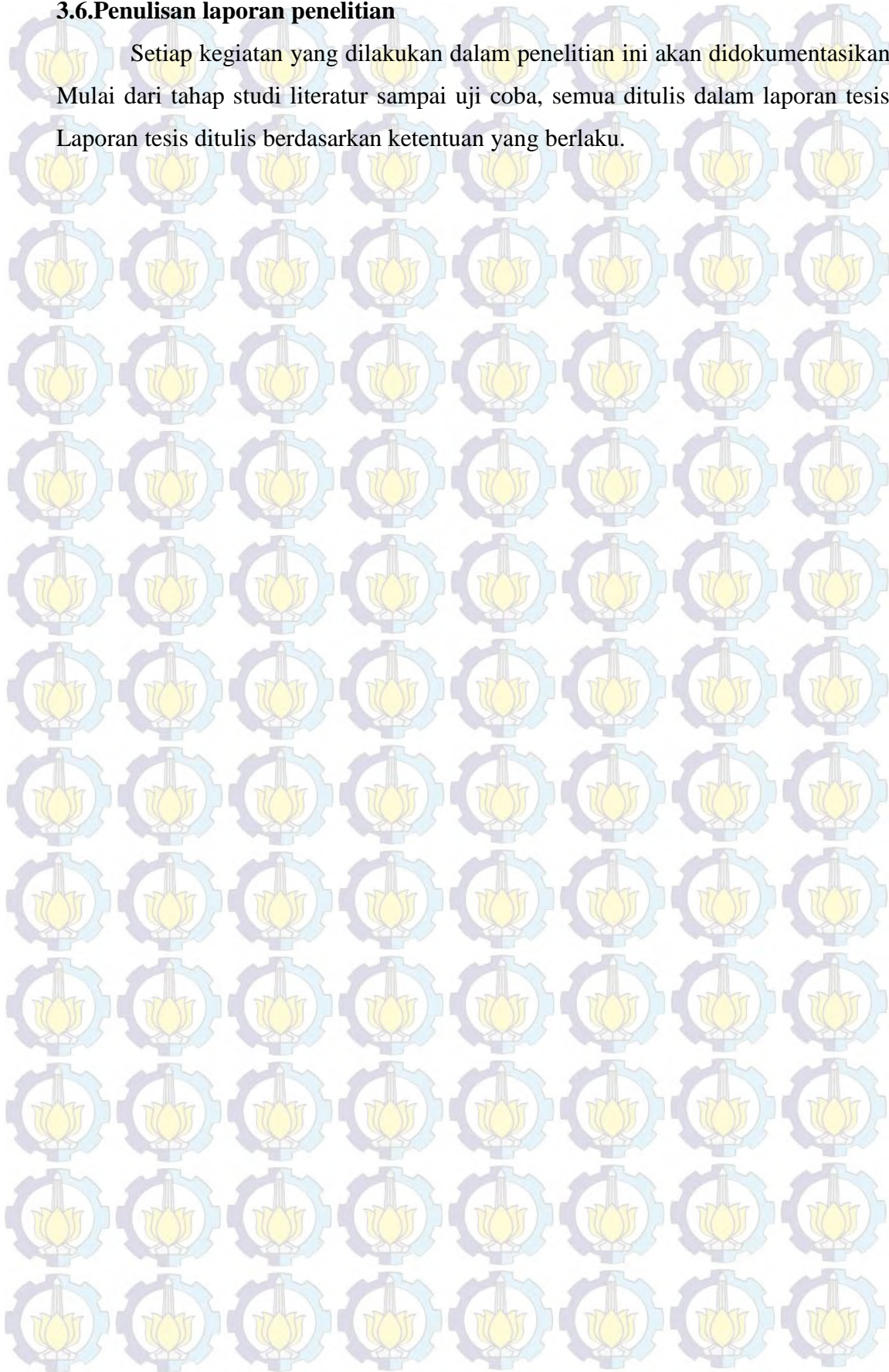
$$\text{sensitivitas} = \frac{TP}{TP + FN} \quad (3.4)$$

$$\text{spesifisitas} = \frac{TN}{TN + FP} \quad (3.5)$$

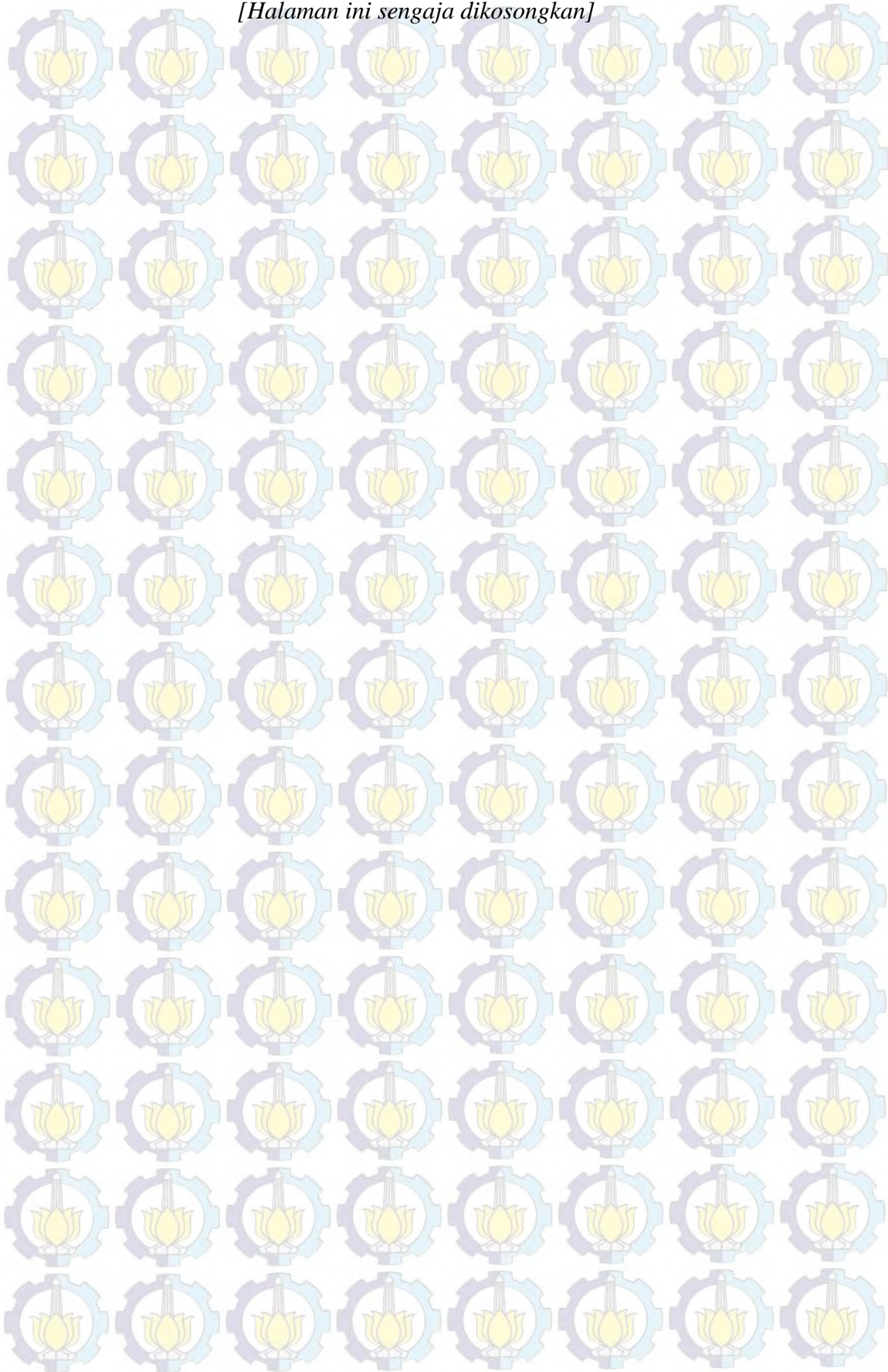


### 3.6. Penulisan laporan penelitian

Setiap kegiatan yang dilakukan dalam penelitian ini akan didokumentasikan. Mulai dari tahap studi literatur sampai uji coba, semua ditulis dalam laporan tesis. Laporan tesis ditulis berdasarkan ketentuan yang berlaku.



*[Halaman ini sengaja dikosongkan]*



## BAB 4 HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan tahapan pengumpulan dataset, perumusan komputasi operator momentum, spesifikasi sistem, implementasi sistem, pengujian dan analisis hasil penelitian.

### 4.1. Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data primer yang diambil dari hasil pemeriksaan urinalysis pasien di Laboratorium Klinik Utama Popular, Surabaya sejak Januari 2012 sampai dengan Maret 2013. Fitur yang diambil adalah sepuluh kandungan urin yang diperoleh dari hasil uji dipstick urin dan beberapa data personal seperti usia dan jenis kelamin. Sebaran sampel kelas serta fitur usia dan jenis kelamin data pasien dapat dilihat pada Tabel 4.1 dan Tabel 4.2.

Tabel 4. 1 Sebaran Kelas Sampel Data

Kelas	Rule Pelabelan	
	1	2
Positif Hiperglikemia	198	198
Pre Hiperglikemia		189
Negatif Hiperglikemia	198	198
Jumlah	396	585

Tabel 4. 2 Sebaran Usia dan Jenis Kelamin Sampel Data

Kelompok		Rule Pelabelan	
		1	2
Usia	< 20	18	21
	20 - 50	165	288
	> 50	213	276
Jenis Kelamin	Pria	252	382
	Wanita	144	203

## 4.2. Spesifikasi Sistem

Pada penelitian ini, digunakan dua perangkat utama yakni komputer sebagai simulator dan *smartphone*. Penelitian ini disimulasikan pada komputer dengan spesifikasi sebagai berikut.

1. CPU : Intel Core i3 1.33 GHz
2. RAM : 2GB
3. OS : Windows 7 Home Basic 32 Bit.

Sedangkan aplikasi pendukung yang digunakan untuk mensimulasikan penelitian ini adalah *toolbox* MATLAB R2009a.

## 4.3. GSA untuk Klasifikasi Data

### 4.3.1 Prosedur GSA untuk Klasifikasi Data

Dalam sub bab ini akan dijelaskan implementasi GSA pada sistem dengan mengacu pada desain yang sudah dijelaskan pada bab sebelumnya.

#### 1) Inisialisasi Parameter

Proses ini bertujuan untuk menentukan parameter-parameter yang digunakan. Parameter dari GSA antara lain jumlah agen  $n$ , konstanta  $G_0$ , konstanta  $\alpha$ , dan konstanta  $\theta$ . Sedangkan parameter untuk struktur jaringan ANN adalah jumlah node input, jumlah node hidden, dan maksimum iterasi.

#### 2) Normalisasi Data

Proses ini bertujuan menyeragamkan interval masing-masing fitur data dengan cara mengubah data bernilai 0 s.d. 1. Berikut ini adalah pseudocode untuk normalisasi data.

#### Algoritma 1. Pseudocode Normalisasi Data

**Input:** data

**Output:** datanormal

**Proses:**

**for each** fitur <sub>$i$</sub>

    maks <sub>$i$</sub>  = max(fitur <sub>$i$</sub> )

    mins <sub>$i$</sub>  = min(fitur <sub>$i$</sub> )

    datanormal = (data - mins <sub>$i$</sub> ) / (maks <sub>$i$</sub>  - mins <sub>$i$</sub> )

**end for**

### 3) Pembangkitan agen

Proses ini bertujuan untuk membangkitkan agen awal sebanyak  $n$ . Pembangkitan agen ini menggunakan metode Nguyen Widrow. Susunan agen yang dibangkitkan seperti pada Gambar 3.3. Ukuran agen berdasarkan jumlah node input, node hidden, dan node output. Sekumpulan agen disimpan pada array tiga dimensi. Berikut ini adalah pseudocode untuk pembangkitan  $n$  agen awal.

#### Algoritma 2. Pseudocode Pembangkitan Agen

```
Input: NodeInput, NodeHidden  
Output: agen  
Proses:  
= 0.7*(NodeHidden^(1/NodeInput))  
for each NodeHidden  
  for each NodeInput  
    inisialisasi  $w_{ij}$   
    end for  
     $w_j = \text{sqrt}(w_{ij})$   
  end for  
  for each NodeHidden  
    for each NodeInput  
       $w_{1ij} = \text{ * } w_{1ij} / w_{1i};$   
    end for  
  end for  
  1 = random (NodeHidden,1)*  
  w2 = random (NodeHidden,NodeOutput)  
  2 = random (NodeOutput,1)  
  agen = [w1 1 w2 2]
```

Sebagai contoh akan dibangkitkan agen dengan 10 input, 4 node hidden, dan 1 output sebanyak 5. Representasi sejumlah agen tersebut dapat dilihat pada Lampiran 1.

### 4) Evaluasi agen

Proses ini bertujuan untuk mengetahui kebaikan suatu agen sehingga dapat tetap dipertahankan. Tahapan pertama adalah mengalikan fitur yang bersesuaian dengan bobot dan bias pada agen sehingga diperoleh output berupa kelas. Kemudian evaluasi dilakukan dengan menghitung nilai MSE antara target dan output. Berikut ini adalah pseudocode untuk mengevaluasi agen.

**Algoritma 3.** Pseudocode Evaluasi Agen**Input:** data, target, populasi agen**Output:** evaluasi**Proses:****for each** record data    **for each** agen,

jalankan feed forward dengan pembobotan data di layer 1

target = klasifikasi data di layer 2

**end for****end for**

evaluasi = mse (target, hasil )

Sebagai contoh, sejumlah agen yang dibangkitkan sebelumnya akan dievaluasi. Hasil evaluasi agen tersebut dapat dilihat pada Lampiran 1.

5) Update nilai konstanta gravitasi  $G(t)$ ,  $Best$ , dan  $Worst$ 

Proses ini bertujuan untuk menghitung nilai  $G(t)$ , menentukan agen terbaik, dan terburuk. Nilai  $G(t)$  dihitung berdasarkan persamaan (2.3). Kemudian  $best(t)$  dipilih dari agen yang memiliki fungsi fitness terendah, sedangkan  $worst(t)$  dipilih dari agen yang memiliki fungsi fitness tertinggi. Berikut ini adalah pseudocode untuk update nilai konstanta gravitasi  $G(t)$ , agen  $best$ , dan agen  $worst$ .

**Algoritma 4.** Pseudocode Menghitung Konstanta Gravitasi, Best, dan Worst**Input:** populasi agen, hasil evaluasi,  $G_0$ ,  $\alpha$ ,  $t$ , maksiterasi**Output:**  $G(t)$ , best agen, worst agen**Proses:** $G(t) = G_0 \cdot \exp(-\alpha \cdot t / \text{maksiterasi})$ 

best agen = agen dengan MSE minimum

worst agen = agen dengan MSE maksimum

Sebagai contoh, perhitungan nilai  $G(t)$  pada iterasi pertama jika  $G_0 = 100$ ,  $\alpha = 20$ , dan maksimum iterasi = 3 serta penentuan agen  $best$  dan  $worst$  dapat dilihat pada Lampiran 1.

6) Hitung massa  $M$ , gaya resultan  $F$ , dan percepatan  $a$ 

Proses ini bertujuan untuk menghitung resultan gaya, massa dan percepatan agen. Massa  $M_i(t)$  dihitung berdasarkan persamaan (2.7). Kemudian menghitung gaya total  $F_i^d(t)$  yang dialami berdasarkan persamaan (2.8). Sedangkan untuk menghitung percepatan  $a_i^d(t)$  menggunakan persamaan (2.10). Berikut ini adalah pseudocode untuk

**Algoritma 5.** Pseudocode Menghitung Massa, Gaya Resultan, Percepatan**Input:** populasi agen**Output:** M, Fr, a**Proses:****for each** agen<sub>i</sub>

$$m_i = (\text{MSE}_i - \text{MSE worst agen}) / (\text{MSE best agen} - \text{MSE worst agen})$$

**end for**

$$M = m_i / \text{sum}(m)$$

**for each** agen<sub>i</sub>**for each** agen<sub>j</sub>

$$r_{ij} = \text{euclid}(\text{agen}_i, \text{agen}_j)$$

$$F_{ij} = G(t) * M_i * M_j / r_{ij} * (\text{agen}_i - \text{agen}_j)$$

**end for**

$$Fr_i = \text{rand} * F_{ij}$$

**end for****for each** agen<sub>i</sub>

$$a_i(t) = Fr_i / M_i$$

**end for**

Sebagai contoh, dilakukan perhitungan massa, gaya, dan percepatan. Hasil perhitungan tersebut dapat dilihat pada Lampiran 2.

7) Update kecepatan  $v$  dan posisi  $x$ 

Proses ini bertujuan untuk update kecepatan  $v_i^d(t+1)$  dan menentukan posisi baru agen  $x_i^d(t+1)$ . Update kecepatan agen dihitung berdasarkan persamaan (2.11) sedangkan untuk menentukan posisi baru agen dihitung berdasarkan persamaan (2.12). Berikut ini adalah pseudocode untuk update kecepatan agen  $v$  dan posisi agen  $x$ .

**Algoritma 6.** Pseudocode kecepatan dan posisi**Input:** populasi agen, a**Output:** v, x**Proses:****for each** agen<sub>i</sub>

$$v_i(t) = \text{rand} * v_i(t-1) + a_i(t)$$

$$x_i = x_i + v_i(t)$$

**end for****for each** agen<sub>i</sub>hitung MSE  $x_i$ **if** MSE  $x <$  MSE agenUpdate agen<sub>i</sub>**end for****end for**

Update best agen

Sebagai contoh, dilakukan perhitungan kecepatan dan update posisi baru pada agen yang telah dibangkitkan sebelumnya. Hasil perhitungan tersebut dapat dilihat pada Lampiran 3.

#### 4.3.2 Pengujian GSA

Parameter dari GSA untuk menentukan nilai  $G(t)$  dan  $C$  ditentukan sebagai berikut:

Jumlah agen = 20

$G_0$  = 100

$\alpha$  = 20

$\theta$  = 100

Sedangkan parameter untuk arsitektur jaringan ANN antara lain:

Jumlah node input = 10

Jumlah node hidden = 7, 11, dan 15

Maksimum iterasi = 50, 100, dan 200.

Pengujian menggunakan GSA dilakukan dalam 3 skenario. Ukuran evaluasi yang digunakan adalah akurasi, sensitifitas, spesifisitas, dan waktu komputasi.

##### 1) Skenario 1

Pengujian pertama pada skenario ini menggunakan dataset Wine dengan 3 kelas. Pengujian kedua pada skenario ini menggunakan data XOR 3 bit parity dengan 2 kelas. Pengujian dilakukan dengan metode 5 fold cross validation. Hasil rata-rata pengujian dapat dilihat pada Lampiran 5.

##### 2) Skenario 2

Pengujian pertama dilakukan pada data urine dengan rule pelabelan 1 yakni klasifikasi dalam 2 kelas (positif hiperglikemia dan negatif hiperglikemia). Sedangkan pengujian kedua dilakukan pada data urine dengan rule pelabelan 2 yakni klasifikasi dalam 3 kelas (positif hiperglikemia, pre hiperglikemia, dan negatif hiperglikemia). Hasil rata-rata pengujian dapat dilihat pada Lampiran 6. Pengujian pada skenario ini dilakukan dengan metode 3, 5, dan 10 fold cross validation.

##### 3) Skenario 3



Pengujian pertama pada skenario ini dilakukan pada data urine dengan pengelompokan faktor usia. Hasil rata-rata pengujian dapat dilihat di Lampiran 7. Pengujian kedua dilakukan pada data urine pada dataset dengan pengelompokan faktor jenis kelamin. Hasil rata-rata pengujian dapat dilihat di Lampiran 8.

#### 4.4 Perumusan Komputasi Operator Momentum

Dalam perumusan komputasi operator momentum, asumsi- asumsi yang digunakan adalah sebagai berikut:

- (i) Dua agen akan mengalami tumbukan jika berada pada suatu threshold berupa jarak tertentu.

Sarafrazi (2011) menyatakan bahwa agen  $i$  dikatakan konvergen dengan agen  $j$  jika jarak agen  $i$  dan agen  $j$  dibandingkan dengan jarak agen  $i$  dan agen terbaik kurang dari suatu nilai tertentu. Dengan demikian agen  $i$  dan agen  $j$  dikatakan mengalami tumbukan jika berada pada kondisi konvergen atau secara matematis dapat ditulis dalam persamaan (4.1)

$$\frac{R_{i,j}}{R_{i,best}} < C \quad (4.1)$$

dimana

$R_{i,j}$  = jarak euclid antara agen  $i$  dengan agen  $j$

$R_{i,best}$  = jarak euclid antara agen  $i$  dengan  $best$ .

Nilai  $C$  yang digunakan sebagai threshold adalah suatu variabel yang menentukan dua agen saling bertumbukan atau tidak. Karena diharapkan terjadi tumbukan, maka ketika agen tidak berada pada kondisi konvergen, nilai  $C$  merupakan nilai yang besar. Sebaliknya jika dalam keadaan konvergen, nilai  $C$  merupakan nilai yang kecil. Nilai  $C$  menurun sepanjang waktu dan dihitung menggunakan persamaan (4.2)

$$C = \theta \left(1 - \frac{t}{t_{max}}\right) \quad (4.2)$$

dimana

$\theta$  = konstanta

$t$  = iterasi saat ini

$t_{max}$  = maksimum iterasi

(ii) Tumbukan yang terjadi adalah tumbukan lenting sempurna.

Jika terjadi tumbukan lenting sempurna, maka berlaku hukum kekekalan momentum seperti pada persamaan (2.15). Selain itu juga berlaku hukum kekekalan energi kinetik seperti pada persamaan (2.16).

(iii) Massa agen sebelum dan sesudah tumbukan adalah sama.

Jika massa agen sebelum dan sesudah tumbukan adalah sama maka  $m_1 = m_1'$  dan  $m_2 = m_2'$ . Dari persamaan (2.15) dengan saling memindahkan ruas maka diperoleh persamaan (4.3).

$$\begin{aligned} m_1 v_1 + m_2 v_2 &= m_1 v_1' + m_2 v_2' \\ m_1 v_1' + m_2 v_2' &= m_1 v_1 + m_2 v_2 \end{aligned} \quad (4.3)$$

Dengan mengumpulkan variabel  $m_1$  dan  $m_2$  masing-masing pada ruas yang sama dan berdasarkan sifat distribusi matematika maka persamaan (4.3) dapat disederhanakan menjadi persamaan (4.4)

$$m_1(v_1 - v_1') = m_2(v_2' - v_2) \quad (4.4)$$

Dari persamaan (2.16) dengan menghilangkan nilai konstanta maka diperoleh persamaan (4.5).

$$\begin{aligned} \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 &= \frac{1}{2} m_1 v_1'^2 + \frac{1}{2} m_2 v_2'^2 \\ m_1 v_1^2 + m_2 v_2^2 &= m_1 v_1'^2 + m_2 v_2'^2 \end{aligned} \quad (4.5)$$

Dengan mengumpulkan variabel  $m_1$  dan  $m_2$  masing-masing pada ruas yang sama serta menggunakan sifat distribusi matematika dan selisih kuadrat, maka persamaan (4.5) dapat ditulis menjadi persamaan (4.6).

$$\begin{aligned} m_1 v_1^2 - m_1 v_1'^2 &= m_2 v_2'^2 - m_2 v_2^2 \\ m_1(v_1^2 - v_1'^2) &= m_2(v_2'^2 - v_2^2) \\ m_1(v_1 + v_1')(v_1 - v_1') &= m_2(v_2' + v_2)(v_2' - v_2) \end{aligned} \quad (4.6)$$

Dengan membagi persamaan (4.6) dengan persamaan (4.4) maka dapat diperoleh persamaan (4.7).

$$\frac{m_1(v_1 + v_1')(v_1 - v_1')}{m_1(v_1 - v_1')} = \frac{m_2(v_2' + v_2)(v_2' - v_2)}{m_2(v_2' - v_2)}$$

$$\begin{aligned} v_1 + v_1' &= v_2' + v_2 \\ v_1' - v_2' &= v_2 - v_1 \end{aligned} \quad (4.7)$$

Untuk menentukan nilai  $v_1'$  dan  $v_2'$  maka persamaan linear (4.3) dan (4.7) diselesaikan menggunakan metode Cremer dan dibentuk menjadi sistem persamaan linear sebagai berikut:

$$\begin{pmatrix} 1 & -1 \\ m_1 & m_2 \end{pmatrix} \begin{pmatrix} v_1' \\ v_2' \end{pmatrix} = \begin{pmatrix} v_2 - v_1 \\ m_1 v_1 + m_2 v_2 \end{pmatrix}$$

Dari sistem persamaan linear tersebut maka dihitung nilai determinan  $D$ , determinan  $D_{v_1'}$ , dan determinan  $D_{v_2'}$ .

$$D = \begin{vmatrix} 1 & -1 \\ m_1 & m_2 \end{vmatrix} = m_1 + m_2$$

$$\begin{aligned} D_{v_1'} &= \begin{vmatrix} v_2 - v_1 & -1 \\ m_1 v_1 + m_2 v_2 & m_2 \end{vmatrix} \\ &= (m_2 v_2 - m_2 v_1) + (m_1 v_1 + m_2 v_2) = (m_1 - m_2)v_1 + 2m_2 v_2 \end{aligned}$$

$$\begin{aligned} D_{v_2'} &= \begin{vmatrix} 1 & v_2 - v_1 \\ m_1 & m_1 v_1 + m_2 v_2 \end{vmatrix} \\ &= (m_1 v_1 + m_2 v_2) - (m_1 v_2 - m_1 v_1) = 2m_1 v_1 + (m_2 - m_1)v_2 \end{aligned}$$

Dengan demikian nilai  $v_1'$  dapat diperoleh menggunakan persamaan (4.8) dan  $v_2'$  dapat diperoleh menggunakan persamaan (4.9).

$$v_1' = \frac{D_{v_1'}}{D} = \frac{(m_1 - m_2)}{m_1 + m_2} v_1 + \frac{2m_2}{m_1 + m_2} v_2 \quad (4.8)$$

$$v_2' = \frac{D_{v_2'}}{D} = \frac{2m_1}{m_1 + m_2} v_1 + \frac{m_2 - m_1}{m_1 + m_2} v_2 \quad (4.9)$$

## 4.5 GSA dan Operator Momentum untuk Klasifikasi Data

### 4.5.1. Prosedur GSAM untuk Klasifikasi Data

Dari tiga asumsi tersebut, prosedur operator momentum dapat dilihat pada pseudocode Algoritma 7. Dengan operator ini dihitung kembali kecepatan baru agen  $v_i^d(t+1)$  jika kemungkinan terjadi tumbukan. Kecepatan baru dua agen yang bertumbukan dihitung menggunakan persamaan (4.8) dan (4.9). Kemudian menentukan posisi baru agen setelah tumbukan berdasarkan persamaan (2.12).

Proses ini bertujuan untuk menghindari konvergensi dini pada GSA.

Untuk lebih lanjut GSA dengan penambahan operator momentum disebut dengan GSAM. Diagram alir GSAM dapat dilihat pada Gambar 4.1. GSAM diimplementasikan sebagai metode pembelajaran pada ANN. Selanjutnya metode ini digunakan untuk mengklasifikasikan data tes urine. Output dari klasifikasi tersebut merupakan hasil screening test berupa positif hipergikemia, pre hiperglikemia, atau negatif hiperglikemia.

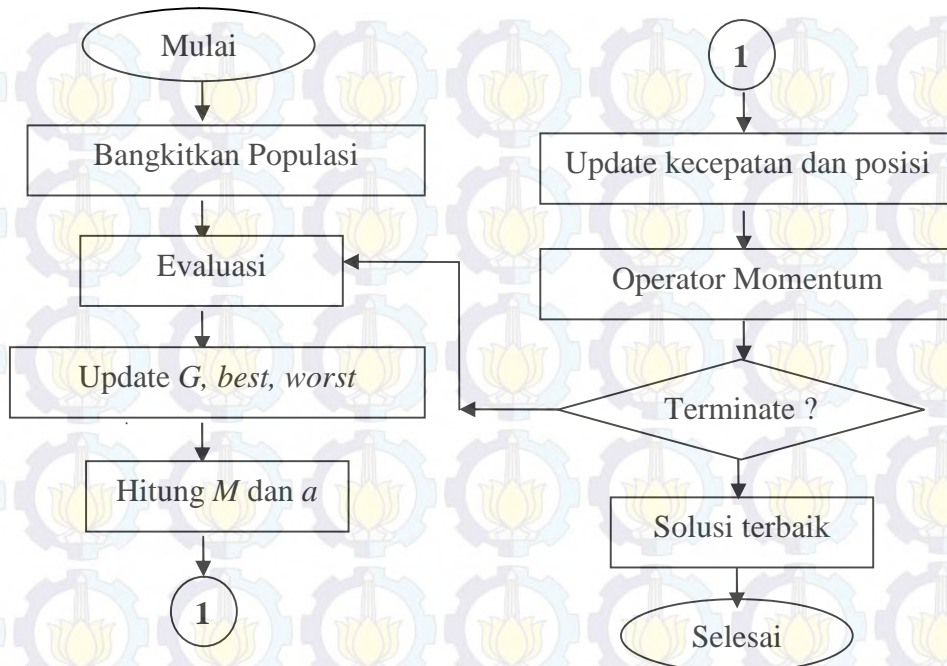
**Algoritma 7.** Pseudocode Operator Momentum

```

Input: agen, v, M, t
Output: agen
Proses:
best agen = agen dengan MSE minimum
for each ageni
  for each agenj
    rij = euclid(ageni, agenj)
    ri,best = euclid(ageni, best)
    C = * (1-t/maksiterasi)
    if rij / rij < C
      vi' = (Mi - Mj)/(Mi + Mj)*vi + (2*Mj)/(Mi + Mj)*vj
      new ageni = ageni + vi (t)
      vj' = (2*Mi)/(Mi + Mj)*vi + (Mj - Mi)/(Mi + Mj)*vj
      new agenj = agenj + vj (t)
    end for
  end for
for each ageni
  hitung MSE new ageni
  if MSE new ageni < MSE agen
    Update ageni
  end for
Update best agen

```

Sebagai contoh, akan dijalankan operator momentum pada hasil perhitungan GSA sebelumnya. Hasil dari dijalkannya operator momentum tersebut dapat dilihat pada Lampiran 4. Dari pengujian ini dapat dilihat bahwa 2 dari 3 agen terupdate sehingga memperoleh MSE yang lebih kecil.



Gambar 4. 1 Diagram alir GSAM

#### 4.5.2. Perhitungan Kompleksitas Operator Momentum

Untuk mengetahui keefektifannya, perlu dilakukan perhitungan kompleksitas algoritma. Kompleksitas algoritma dihitung berdasarkan perulangan yang dilakukan oleh algoritma tersebut. Tabel 4.3 menunjukkan proses perulangan dalam operator momentum. Jika  $n$  adalah jumlah agen yang dibangkitkan, berdasarkan perulangan tersebut kompleksitas operator momentum adalah:

$$\begin{aligned}
 O(n) &= n \times n + n \\
 &= n^2 + n \\
 &= n(n + 1)
 \end{aligned}$$

Kompleksitas ini merupakan jenis polynomial. Hal ini menunjukkan bahwa algoritma yang diusulkan kurang efisien. Oleh karena itu perlu strategi untuk mengurangi kompleksitas tersebut.

Tabel 4. 3 Kompleksitas operator momentum

Pseudocode	Perulangan	Jumlah Eksekusi
<pre> for each agen<sub>i</sub>   for each agen<sub>j</sub>     r<sub>ij</sub> = euclid(agen<sub>i</sub> , agen<sub>j</sub>)     r<sub>i,best</sub> = euclid(agen<sub>i</sub> , best)     C = * (1-t/maksiterasi)     if r<sub>ij</sub> / r<sub>ij</sub> &lt; C       v<sub>i</sub>' = (M<sub>i</sub> - M<sub>j</sub>)/(M<sub>i</sub> + M<sub>j</sub>)*v<sub>i</sub> + (2*M<sub>j</sub>)/(M<sub>i</sub> + M<sub>j</sub>)*v<sub>j</sub>       new agen<sub>i</sub> = agen<sub>i</sub> + v<sub>i</sub> (t)       v<sub>j</sub>' = (2*M<sub>i</sub>)/(M<sub>i</sub> + M<sub>j</sub>)*v<sub>i</sub> + (M<sub>j</sub> - M<sub>i</sub>)/(M<sub>i</sub> + M<sub>j</sub>)*v<sub>j</sub>       new agen<sub>j</sub> = agen<sub>j</sub> + v<sub>j</sub> (t)     end if   end for end for for each agen<sub>i</sub>   hitung MSE new agen<sub>i</sub>   if MSE new agen<sub>j</sub> &lt; MSE agen     Update agen<sub>i</sub>   end if end for Update best agen </pre>	<pre> for i = 1 to n   for j = 1 to n     .     .     if ( )     .     .   end if end for end for for i = 1 to n   if ( )   end if end for </pre>	<p>Diagram showing execution counts: <math>n^2</math> for the inner loop, <math>n</math> for the middle loop, and <math>n</math> for the final loop.</p>

### 4.5.3. Pengujian GSAM

Selanjutnya dilakukan pengujian GSAM untuk klasifikasi data. Parameter dari GSAM untuk menentukan nilai  $G(t)$  dan  $C$  ditentukan sebagai berikut:

- Jumlah agen = 20
- $G_0$  = 100
- $\alpha$  = 20
- $\theta$  = 100

Sedangkan parameter untuk arsitektur jaringan ANN antara lain:

- Jumlah node input = 10
- Jumlah node hidden = 7, 11, dan 15
- Maksimum iterasi = 50, 100, dan 200.

Pengujian menggunakan GSAM akan dibandingkan dengan screening test menggunakan GSA. Parameter yang digunakan pada GSAM sama dengan parameter pada GSA. Ukuran evaluasi yang digunakan adalah akurasi, sensitifitas, spesifisitas, dan waktu komputasi. Pengujian ini juga dilakukan dalam 3 skenario.

1) Skenario 1

Pengujian pertama pada skenario ini menggunakan dataset Wine dengan 3 kelas. Pengujian kedua pada skenario ini menggunakan data XOR 3 bit parity dengan 2 kelas. Pengujian dilakukan dengan metode 5 fold cross validation. Hasil rata-rata pengujian dapat dilihat pada Lampiran 5.

2) Skenario 2

Pengujian pertama dilakukan pada data urine dengan rule pelabelan 1 yakni klasifikasi dalam 2 kelas (positif hiperglikemia dan negatif hiperglikemia). Sedangkan pengujian kedua dilakukan pada data urine dengan rule pelabelan 2 yakni klasifikasi dalam 3 kelas (positif hiperglikemia, pre hiperglikemia, dan negatif hiperglikemia). Hasil rata-rata pengujian dapat dilihat pada Lampiran 9. Pengujian pada skenario ini dilakukan dengan metode 3, 5, dan 10 fold cross validation.

3) Skenario 3

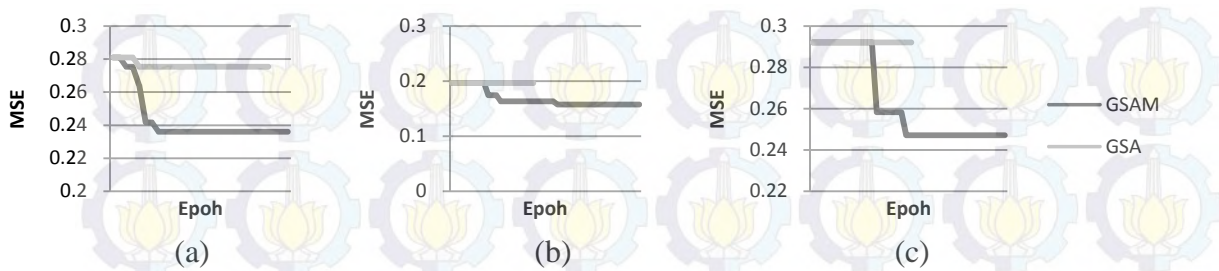
Pengujian pertama pada skenario ini dilakukan pada data urine dengan pengelompokan faktor usia. Hasil rata-rata pengujian dapat dilihat di Lampiran 10. Pengujian kedua dilakukan pada data urine pada dataset dengan pengelompokan faktor jenis kelamin. Hasil rata-rata pengujian dapat dilihat di Lampiran 11. Untuk pengujian ketiga, yakni pengujian yang bertujuan untuk menyimpan rekam medis pasien.

#### **4.6 Analisis Hasil Pengujian menggunakan GSA dan GSAM**

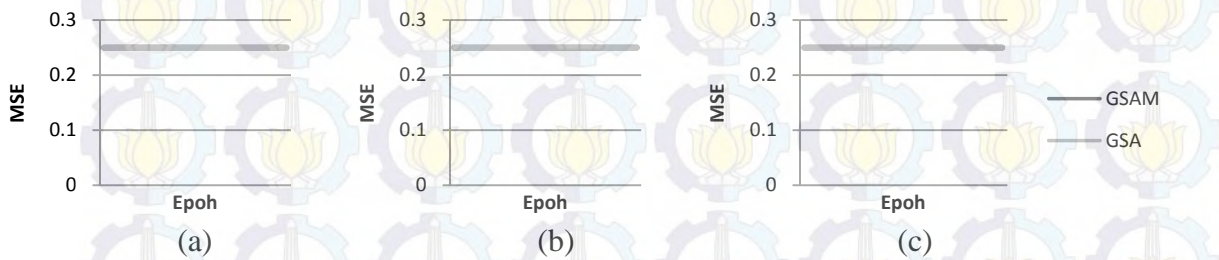
Pada tesis ini, hasil perbandingan kemampuan klasifikasi metode ANN-GSAM dan ANN-GSA dianalisis dari sisi kecepatan konvergensi, kemampuan metode dalam akurasi, sensitivitas, serta spesifisitas, dan waktu komputasi.

##### **4.6.1. Hasil perbandingan kecepatan konvergensi**

Penambahan operator momentum pada GSA bertujuan untuk menghindari konvergensi dini, oleh karena itu ukuran keberhasilan GSAM dapat dilihat dari nilai MSE yang lebih optimal dan diperoleh pada saat iterasi yang tidak terlalu rendah. Gambar 4.2(a), 4.2(b), dan 4.2(c) menunjukkan hasil uji coba skenario pertama yakni klasifikasi dataset Wine dengan maksimum iterasi masing-masing 50, 100, dan 200. Selanjutnya Gambar 4.3(a), 4.3(b), dan 4.3(c) menunjukkan hasil uji coba klasifikasi dataset XOR dengan maksimum iterasi yang sama seperti sebelumnya.

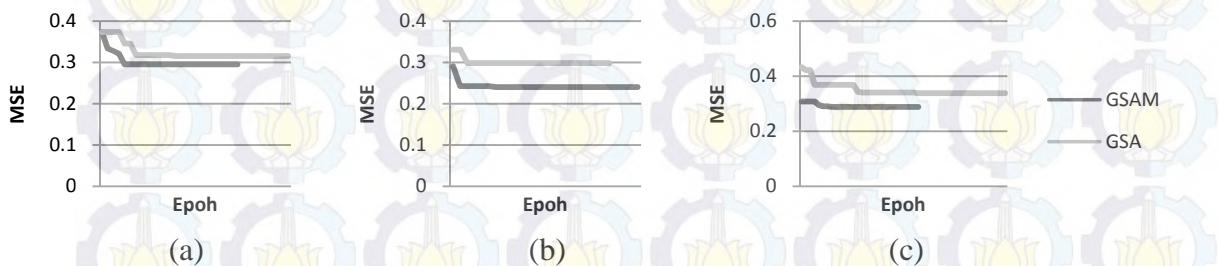


Gambar 4. 2 Perbandingan Kecepatan Konvergensi pada Dataset Wine



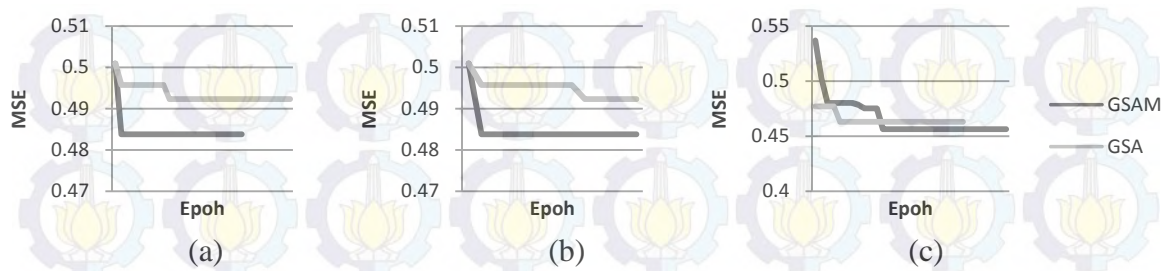
Gambar 4. 3 Perbandingan Kecepatan Konvergensi pada Dataset XOR 3 bit

Berdasarkan Gambar 4.2(a), pada kasus klasifikasi dataset Wine, GSAM menghasilkan MSE sebesar 0.236 pada iterasi ke-9 sedangkan GSA menghasilkan MSE sebesar 0.275 yang diperoleh pada iterasi ke-6. Nilai yang diperoleh GSA pada iterasi ke-6 merupakan nilai minimum yang dapat ditemukan sampai solusi tersebut konvergen. Dengan demikian terlihat bahwa GSA mengalami konvergensi yang lebih cepat daripada GSAM. Hal serupa juga ditunjukkan pada Gambar 4.2(b) dan Gambar 4.2(c). Lain halnya pada klasifikasi data XOR, konvergensi yang ditunjukkan GSAM dan GSA adalah sama yakni dengan MSE 0.25. Uji coba selanjutnya adalah uji coba screen test hiperglikemia berdasarkan data urinalysis. Uji coba skenario kedua yakni dengan melakukan klasifikasi dataset urinalysis 2 kelas dan 3 kelas. Gambar 4.4(a), 4.4(b), dan 4.4(c) adalah hasil uji coba pada kasus klasifikasi dataset urinalysis 2 kelas dengan maksimum iterasi masing-masing 50, 100, dan 200. Sedangkan hasil uji coba pada dataset urinalysis untuk 3 kelas dengan maksimum iterasi yang sama ditunjukkan pada Gambar 4.5(a), 4.5(b), dan 4.5(c).



Gambar 4. 4 Perbandingan Kecepatan Konvergensi pada Kasus Urinalysis 2 Kelas





Gambar 4. 5 Perbandingan Kecepatan Konvergensi pada Kasus Urinalysis 3 Kelas

Gambar 4.4, Gambar 4.5(a), dan (b) menunjukkan hasil yang sama seperti pada klasifikasi Wine yakni GSAM menghasilkan nilai yang lebih kecil dan GSA mengalami konvergensi lebih cepat. Namun Gambar 4.5(c) menunjukkan bahwa adakalanya GSA menghasilkan nilai yang lebih kecil yakni 0.456 dibandingkan GSAM yang mendapatkan 0.463.

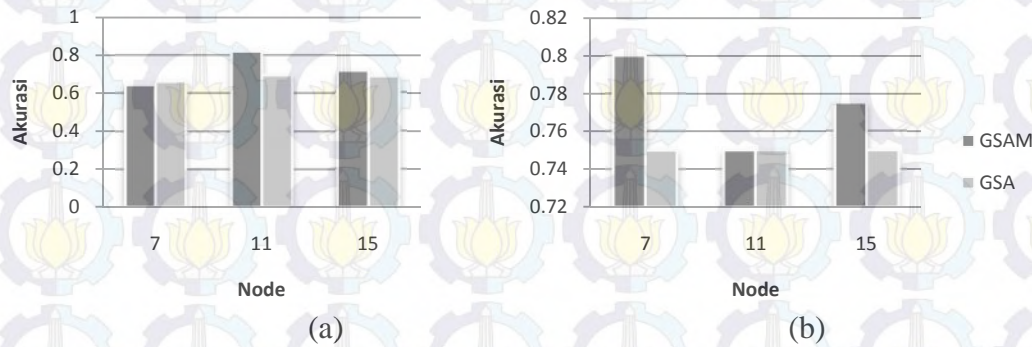
Dari beberapa uji coba tersebut dapat dilihat bahwa pada kasus XOR, konvergensi dan nilai MSE yang dihasilkan ANN-GSA dan ANN-GSAM sama. Sedangkan pada kasus selain klasifikasi XOR, metode ANN-GSA mengalami konvergensi yang lebih cepat dan MSE yang dihasilkan bukan merupakan MSE yang minimum. Hal ini menunjukkan bahwa operator momentum berpengaruh signifikan pada dataset berdimensi tinggi seperti Wine (13 fitur) dan urinalysis (10 fitur). Akan tetapi pengaruh adanya operator ini tidak terlihat signifikan pada dataset berdimensi rendah seperti XOR (3 fitur). Sehingga dapat disimpulkan bahwa operator momentum akan berpengaruh jika dataset berdimensi tinggi.

Pada banyak pengujian, kedua metode ini menemukan solusi optimal sebelum iterasi ke-50. Hal ini menunjukkan bahwa hasil yang diberikan pada saat maksimum iterasi sebesar 100 dan 200 akan sama dengan hasil yang diberikan ketika maksimum iterasi sebesar 50. Sehingga dapat disimpulkan bahwa dalam hal ini maksimum iterasi tidak berpengaruh dalam penentuan solusi yang optimum.

#### 4.6.2. Hasil perbandingan akurasi, sensitifitas, dan spesifisitas

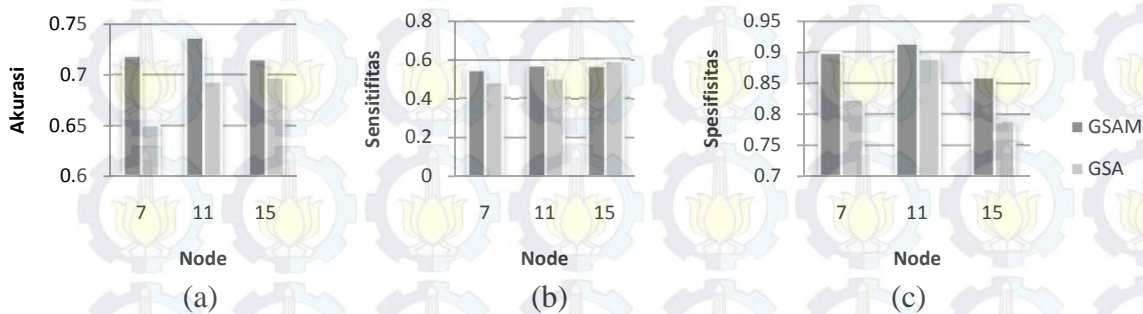
Dengan terhindarnya konvergensi dini, diharapkan GSAM menemukan bobot dan bias ANN yang optimum sehingga memberikan hasil yang lebih baik dari pada GSA. Ukuran evaluasi yang digunakan adalah akurasi, sensitifitas, dan spesifisitas. Gambar 4.6(a) dan 4.6(b) adalah perbandingan akurasi untuk skenario pertama yakni klasifikasi dataset Wine, dan XOR. Dari uji coba tersebut dapat dilihat bahwa

kemampuan metode yang diusulkan memberikan hasil klasifikasi yang lebih baik. Hasil selengkapnya dapat dilihat pada Lampiran 1.

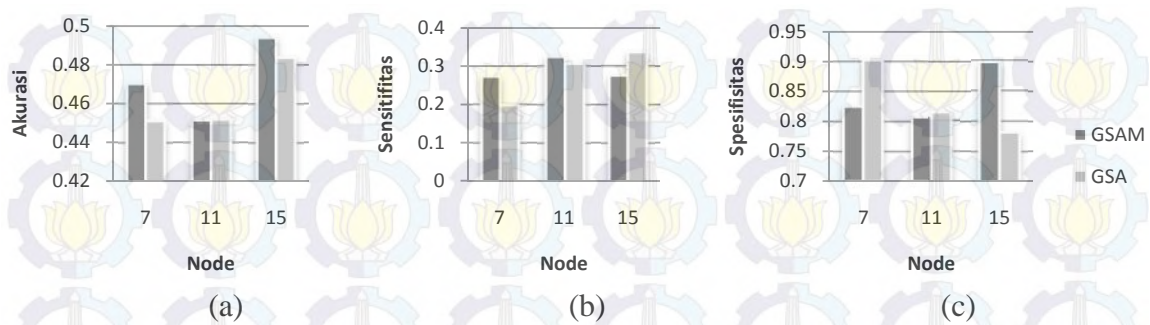


Gambar 4. 6 Perbandingan hasil klasifikasi pada Dataset Umum

Berdasarkan Gambar 4.6, secara umum GSAM memberikan akurasi yang lebih tinggi dibandingkan dengan GSA. Namun pada dataset wine, hasil uji GSAM untuk node sebesar 7 hanya mencapai 64.48% lebih kecil dibandingkan dengan GSA yang memberikan akurasi sebesar 66.17%. Uji coba selanjutnya adalah skenario kedua yakni screen test hiperglikemia pada dataset urinalysis 2 kelas dan 3 kelas. Gambar 4.7(a), 4.7(b), dan 4.7(c) adalah perbandingan akurasi, sensitifitas, dan spesifisitas untuk klasifikasi dataset urinalysis untuk 2 kelas. Sedangkan perbandingan akurasi, sensitifitas, dan spesifisitas untuk klasifikasi dataset urinalysis untuk 3 kelas ditunjukkan pada Gambar 4.8(a), 4.8(b), dan 4.8(c). Hasil selengkapnya dapat dilihat pada Lampiran 2 dan Lampiran 5.



Gambar 4. 7 Perbandingan hasil klasifikasi pada Kasus Urinalysis 2 Kelas



Gambar 4. 8 Perbandingan hasil klasifikasi pada Kasus Urinalysis 3 Kelas

Berdasarkan Gambar 4.7(a) dan Gambar 4.8(a) nilai akurasi GSAM lebih baik daripada akurasi GSA. Namun pada nilai sensitifitas dan spesifisitas, GSAM tidak selalu memberikan hasil yang lebih baik. Misalnya, pada klasifikasi urin 2 kelas dengan node hidden sebesar 15, nilai sensitifitas GSAM lebih rendah yakni 53.33% dibandingkan dengan GSA yakni sebesar 55.02%.

Dari uji coba tersebut dapat dilihat bahwa secara umum nilai akurasi yang diberikan oleh ANN-GSAM lebih tinggi dari ANN-GSA. Namun adakalanya akurasi yang diberikan ANN-GSAM lebih rendah atau sama dengan ANN-GSA seperti pada kasus klasifikasi XOR dengan node hidden sebesar 7. Hal ini wajar terjadi karena proses yang dilakukan dari tahap inisialisasi sampai penentuan posisi agen melibatkan bilangan random. Namun secara rata-rata, akurasi yang diberikan ANN-GSAM lebih tinggi daripada ANN-GSA. Dan dari beberapa uji coba yang dilakukan, nilai akurasi tertinggi diperoleh ketika mengklasifikasi data urin 2 kelas yakni mencapai 78.2% sedangkan sensitivitas dan spesifisitas masing-masing sebesar 66.5% dan 89.9% dengan node hidden sebesar 15.

Di sisi lain, kemampuan metode yang diusulkan untuk melakukan screen test pada kasus 2 kelas menunjukkan hasil akurasi rata-rata yang lebih baik yakni >65%, sedangkan pada 3 kelas menghasilkan nilai akurasi yang rendah yakni <50%. Di sisi lain untuk nilai sensitifitas kedua metode menunjukkan hasil yang rendah sedangkan nilai spesifisitas menunjukkan yang tinggi. Hal ini menunjukkan bahwa sistem lebih mampu mendeteksi orang yang sehat dari pada mendeteksi orang yang sakit. Oleh karena nilai sensitifitas yang dihasilkan rendah, maka perlu diteliti korelasi ( $r$ ) antara fitur-fitur dengan label data. Tabel 4.4 menunjukkan nilai korelasi antar fitur dengan kelas. Dari hasil perhitungan tersebut dapat disimpulkan bahwa yang paling berelasi adalah glukosa. Namun relasi antara glukosa dan kelas tidak berhubungan erat

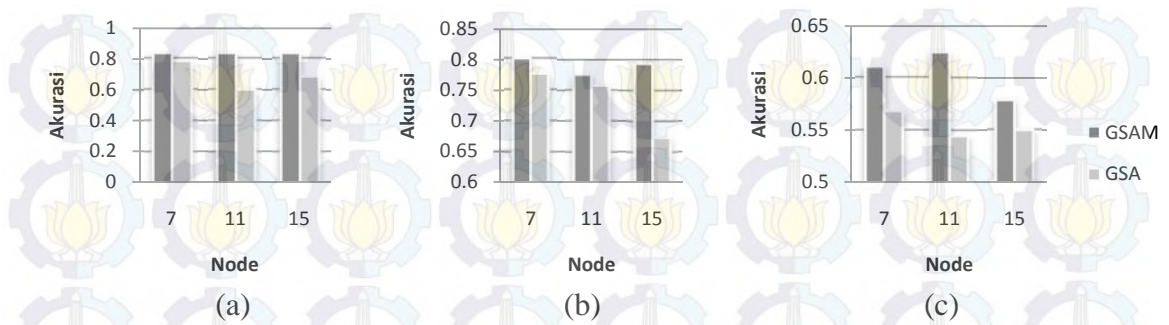
karena nilai  $r$  jauh dari angka 1. Oleh karena itu, kecilnya korelasi antara kandungan dalam urin dengan label penyakit pasien menyebabkan nilai akurasi klasifikasi data urin rendah.

Tabel 4. 4 Korelasi fitur dan label

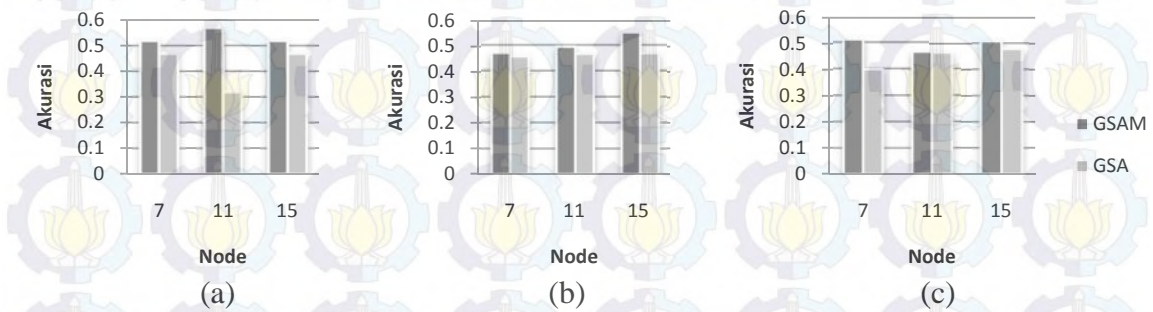
Fitur	Nilai Korelasi ( $r$ )	
	2 kelas	3 kelas
PH	0.072	0.073
BJ	0.069	0.051
protein	0.152	0.183
Glukosa	0.570	0.545
Urobilin	0.071	0.071
Bilirubin	0.082	0.041
Keton	0.057	0.054
Nitrit	0.055	0.051
Eritrosit	0.035	0.006
Lekosit	0	0

#### 4.6.3. Hasil perbandingan pada kasus-kasus khusus

Klasifikasi dilakukan pula pada kasus-kasus khusus kelompok usia dan jenis kelamin seperti yang dijelaskan pada skenario ketiga. Pada kelompok usia, dataset dibagi menjadi 3 kategori sedangkan pada kelompok jenis kelamin, dataset dibagi menjadi 2 kategori seperti dijelaskan pada subbab 3.1. Gambar 4.9(a), 4.9(b), dan 4.9(c) adalah perbandingan akurasi klasifikasi dataset urinalysis 2 kelas pada kelompok usia masing-masing kategori, yakni <20 tahun, 20 s.d. 50 tahun, dan >50 tahun. Sedangkan Gambar 4.10(a), 4.10(b), dan 4.10(c) adalah perbandingan akurasi klasifikasi dataset urinalysis 2 kelas pada kelompok usia masing-masing kategori. Hasil selengkapnya dapat dilihat pada Lampiran 3 dan Lampiran 6.

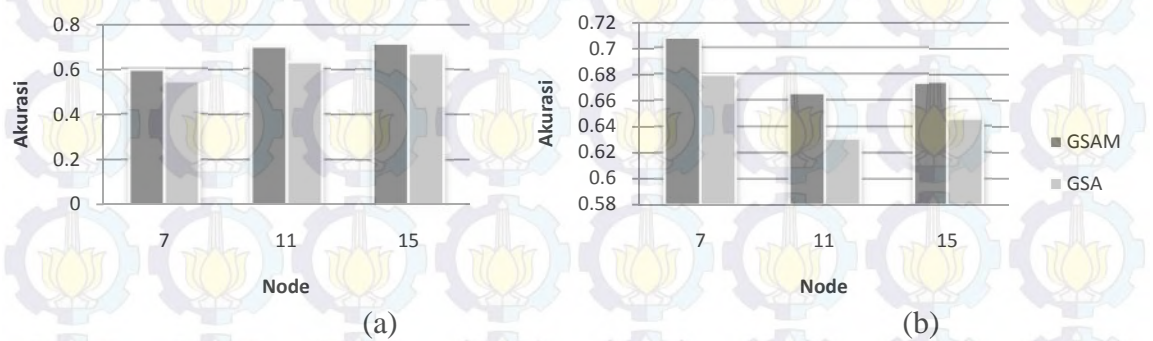


Gambar 4. 9 Perbandingan Hasil Klasifikasi Urin 2 Kelas pada Kelompok Usia

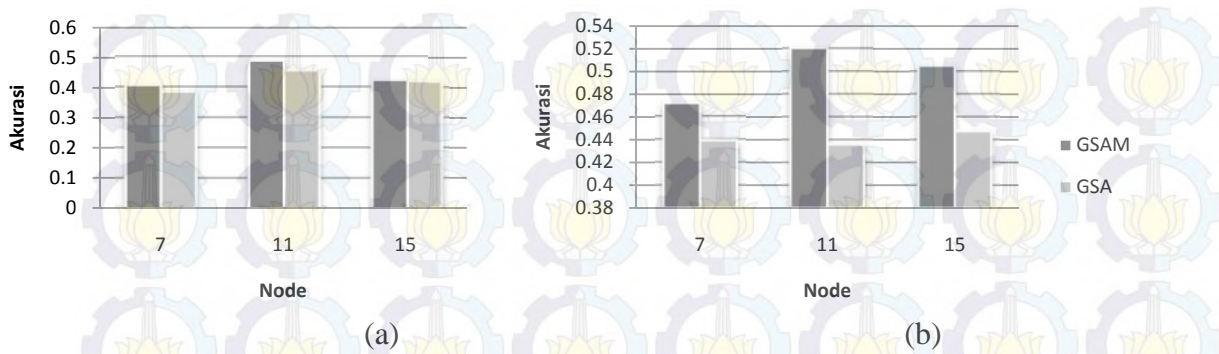


Gambar 4. 10 Perbandingan Hasil Klasifikasi Urin 3 Kelas pada Kelompok Usia

Uji coba selanjutnya adalah uji coba pada kelompok jenis kelamin. Pada kelompok ini, dataset dibagi menjadi 2 kategori seperti dijelaskan pada subbab 3.1. Gambar 4.11(a), 4.11(b), adalah adalah perbandingan akurasi klasifikasi dataset urinalysis 2 kelas pada kelompok usia masing-masing kategori, yakni pria dan wanita. Sedangkan Gambar 4.12(a), 4.12(b), adalah adalah perbandingan akurasi klasifikasi dataset urinalysis 3 kelas pada kelompok usia masing-masing kategori. Hasil selengkapnya dapat dilihat pada Lampiran 4 dan 7.



Gambar 4. 11 Perbandingan hasil klasifikasi pada Jenis Kelamin 2 kelas

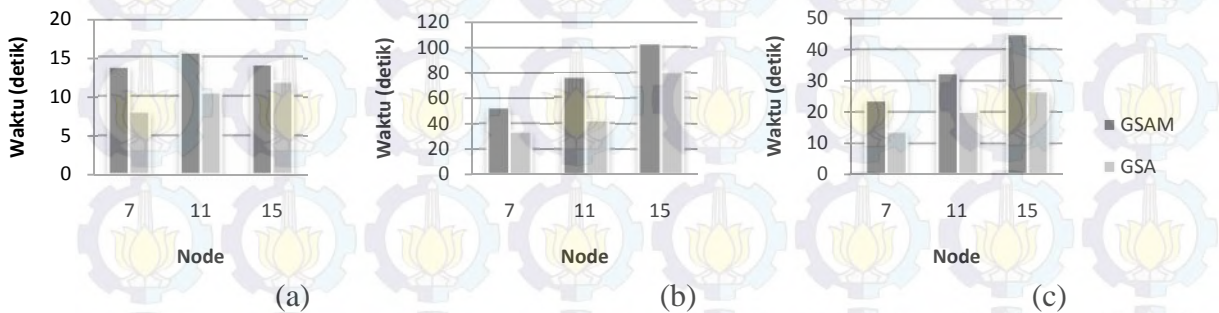


Gambar 4. 12 Perbandingan hasil klasifikasi pada Jenis Kelamin 3 kelas

Pengujian dengan pengelompokan usia dan jenis kelamin menunjukkan hasil yang sama dengan tanpa pengelompokan yakni ANN-GSAM memberikan nilai akurasi yang lebih tinggi daripada ANN-GSA. Akurasi pada proses klasifikasi urin 2 kelas memberikan nilai yang lebih baik yakni diatas 55% jika dibandingkan dengan klasifikasi urin 3 kelas dengan akurasi yang rata-rata kurang dari 55%. Hal ini menunjukkan bahwa pada kasus-kasus khusus pun operator momentum mampu mengatasi konvergensi pada GSA.

#### 4.6.4. Hasil perbandingan waktu komputasi

Selain menghindari terjadinya konvergensi dini dan pencarian solusi optimal, waktu yang dibutuhkan untuk komputasi juga harus dipertimbangkan. Gambar 4.13(a), 4.13(b), dan 4.13(c) masing-masing adalah perbandingan waktu yang dibutuhkan GSAM dan GSA untuk klasifikasi dataset Wine, dataset urinalysis untuk 2 kelas dan 3 kelas.



Gambar 4. 13 Perbandingan Waktu Komputasi

Dari uji coba tersebut dapat dilihat bahwa metode yang diusulkan membutuhkan waktu komputasi yang tinggi. Berdasarkan Gambar 4.13, waktu komputasi GSAM paling lama adalah pada kasus klasifikasi urin 2 kelas dengan node hidden sebesar 15 yang mencapai rata-rata 103 detik jauh lebih lama

dibandingkan dengan GSA yang membutuhkan waktu rata-rata 80 detik dengan node hidden yang sama. Hal ini sesuai dengan perhitungan kompleksitas algoritma pada subbab sebelumnya. Dengan penambahan operator momentum dengan kompleksitas algoritma sebesar  $O(n) = n(n + 1)$ , GSAM memang memberikan hasil yang baik namun menyebabkan banyaknya waktu yang dibutuhkan. Hal ini disebabkan oleh kondisi diversitas algoritma yang selalu dijaga sehingga memperlambat konvergensi.

#### 4.7 Hasil Rekam Medis Pasien Hiperglikemia

Lebih lanjut, metode cerdas yang diusulkan dapat digunakan untuk screen test hasil urinalysis sekaligus memonitoring kondisi pasien. Oleh karena itu dibutuhkan rekam medis hasil urinalysis dari pasien. Berikut ini merupakan rekam medis urinalysis pasien yang telah melakukan pemeriksaan selama lebih dari satu kali pada rentang waktu Januari 2012 s.d. Maret 2013. Terdapat 2 pasien yang melakukan pemeriksaan lebih dari sekali yakni pasien dengan nomor 57 dan 577. Tabel 4.5 menunjukkan data pribadi pasien sedangkan Tabel 4.6 menunjukkan hasil uji urinalysis pasien tersebut.

Tabel 4. 5 Data pasien yang melakukan pemeriksaan lebih dari sekali

Nomor Pasien	Usia	Jenis Kelamin	Tanggal Pemeriksaan
57	59	P	3 Februari 2012
			4 September 2012
577	51	P	19 Maret 2012
			19 Maret 2013

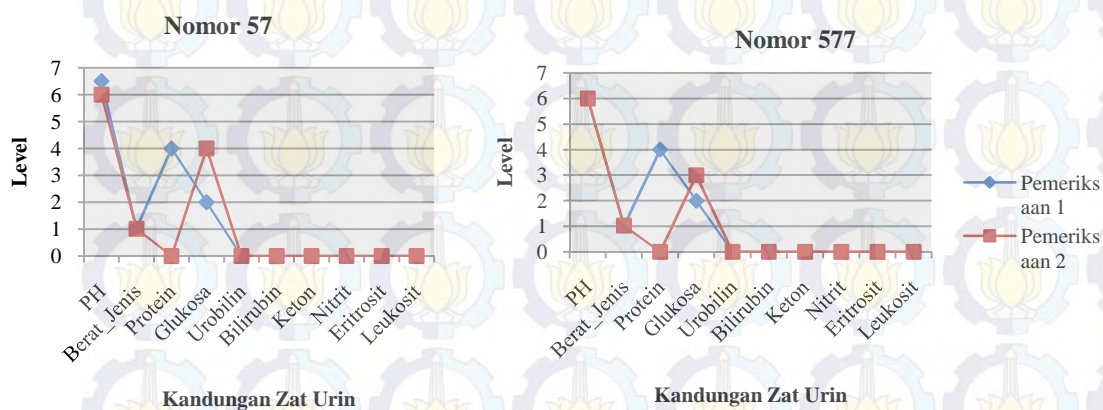
Tabel 4. 6 Rekam medis urine pasien

Nomor	Pemeriksaan	PH	Berat Jenis	Protein		Urobilin		Keton	Nitrit	Eritrosit	Lekosit	Label
				Glukosa	Bilirubin	Bilirubin	Bilirubin					
57	1	6.5	1.02	4	2	0	0	0	0	0	0	Positif
	2	6	1.015	0	4	0	0	0	0	0	0	Positif
577	1	6	1.02	4	2	0	0	0	0	0	0	Positif
	2	6	1.03	0	3	0	0	0	0	0	0	Positif

Contoh hasil implementasi monitoring sistem ini digunakan untuk screen test hiperglikemia dai pasien tersebut pada Tabel 4.7. Sistem ini juga bertujuan untuk dimanfaatkan sebagai alat bantu untuk memonitori hasil uji urine. Sehingga jika hasil uji uin dua pasien tersebut disajikan dalam bentuk grafik, dapat dilihat pada Gambar 4.14.

Tabel 4. 7 Hasil screen test menggunakan sistem

Nomor	Tanggal Pemeriksaan	Target	Hasil Uji	Keterangan
57	3 Februari 2012	Positif	Positif	Benar
	4 September 2012	Positif	Positif	Benar
577	19 Maret 2012	Positif	Positif	Benar
	19 Maret 2013	Positif	Positif	Benar



Gambar 4. 14 Rekam medis hasil uji urine pasien



## BAB 5

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut.

1. Operator momentum pada GSA terbukti dapat menghindari kelemahan algoritma yakni konvergensi dini. Dalam kasus ini, operator momentum dan GSA digunakan sebagai algoritma pembelajaran pada ANN untuk klasifikasi data. Hasil signifikan terlihat pada kasus dataset berdimensi tinggi.
2. Dengan mempunyai GSAM menghindari konvergensi dini, GSAM dapat mengeksplor solusi optimal. Hasil menunjukkan secara rata-rata akurasi GSAM lebih baik dari pada GSA untuk klasifikasi dataset umum dan dataset urine dengan atau tanpa pengeompakan. Pada kasus klasifikasi dataset urin, hasil terbaik yang diperoleh dengan akurasi sebesar 78.2%, sensitivitas sebesar 66.5%, dan spesifisitas sebesar 89.9% dengan arsitektur 10-15-1.
3. Waktu komputasi yang dibutuhkan oleh ANN-GSAM lebih lama dibandingkan dengan waktu komputasi yang dibutuhkan ANN-GSA. Hal ini disebabkan oleh kompleksitas algoritma yakni  $O(n) = n^2 + n$ . Dengan demikian terlihat bahwa algoritma memiliki kompleksitas jenis polinomial yang menandakan bahwa algoritma tidak efisien.

#### 5.2 Saran

Berdasarkan hasil penelitian ini, saran yang dapat diberikan agar diperoleh hasil yang lebih baik dan efisien adalah dengan mempercepat waktu komputasi. Mempercepat waktu komputasi dapat dilakukan dengan cara parallel. Selain itu, prototype yang dibuat dapat dikembangkan secara luas agar dapat digunakan sebagai sistem monitoring kesehatan.

## BAB 5

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut.

1. Operator momentum pada GSA terbukti dapat menghindari kelemahan algoritma yakni konvergensi dini. Dalam kasus ini, operator momentum dan GSA digunakan sebagai algoritma pembelajaran pada ANN untuk klasifikasi data. Hasil signifikan terlihat pada kasus dataset berdimensi tinggi.
2. Dengan mempunyai GSAM menghindari konvergensi dini, GSAM dapat mengeksplor solusi optimal. Hasil menunjukkan secara rata-rata akurasi GSAM lebih baik dari pada GSA untuk klasifikasi dataset umum dan dataset urine dengan atau tanpa pengeompakan. Pada kasus klasifikasi dataset urin, hasil terbaik yang diperoleh dengan akurasi sebesar 78.2%, sensitivitas sebesar 66.5%, dan spesifisitas sebesar 89.9% dengan arsitektur 10-15-1.
3. Waktu komputasi yang dibutuhkan oleh ANN-GSAM lebih lama dibandingkan dengan waktu komputasi yang dibutuhkan ANN-GSA. Hal ini disebabkan oleh kompleksitas algoritma yakni  $O(n) = n^2 + n$ . Dengan demikian terlihat bahwa algoritma memiliki kompleksitas jenis polinomial yang menandakan bahwa algoritma tidak efisien.

#### 5.2 Saran

Berdasarkan hasil penelitian ini, saran yang dapat diberikan agar diperoleh hasil yang lebih baik dan efisien adalah dengan mempercepat waktu komputasi. Mempercepat waktu komputasi dapat dilakukan dengan cara parallel. Selain itu, prototype yang dibuat dapat dikembangkan secara luas agar dapat digunakan sebagai sistem monitoring kesehatan.

## DAFTAR PUSTAKA

American Diabetes Association (ADA), (2012). "Diagnosis and Classification of Diabetes Mellitus", *Diabetes Care*, Vol. 35, hal. S64-S71

Barrat, J., (2007), "What to do with patients with abnormal dipstick urinalysis", *Elsevier Ltd.*

Calva, D., Garcia M., Martinez, C., Salgado, G., Lehman, M., (2009), "Urine and Copro Recognition with Generalized Entropy and Neural Networks", *International Journal of Computer Science and Network Security*, Vol. 9, No. 4, hal. 173-179

Calva, D., Quezada, A.L., Lehman, M., (2004). "Application of Neural Networks and Fractals for Urinalysis", *Proceedings of WSEAS International Conference on Applied Mathematics*, Mexico

Falakaflaki, (2011), "Dipstick Urinalysis Screening of Healthy Neonates", *Pediatrics and Neonatology*, Vol. 52, hal. 161-164

Izzah, A., Ginardi, H., Sarno, R., (2013), "Gravitational Search Algorithm dengan Operator Disruption sebagai Optimasi pada Artificial Neural Network untuk Klasifikasi Data", *Prosiding Konferensi Nasional Informatika ITB*, Vol. 3, hal. 102-107

Izzah, A., Ginardi, H., Saikhu, A., (2013), "Pendekatan Algoritma Heuristik dan Neural Network untuk Screening Test pada Urinalysis", *Jurnal Cybermatika*, Vol. 1, No 2, hal. 29-35

Mastrogiannis, N., Boutsinasa, B., Giannikos, I., (2009), "A Method for Improving The Accuracy of Data Mining Classification Algorithms", *Computers & Operations Research*, Vol. 36, hal. 2829-2839

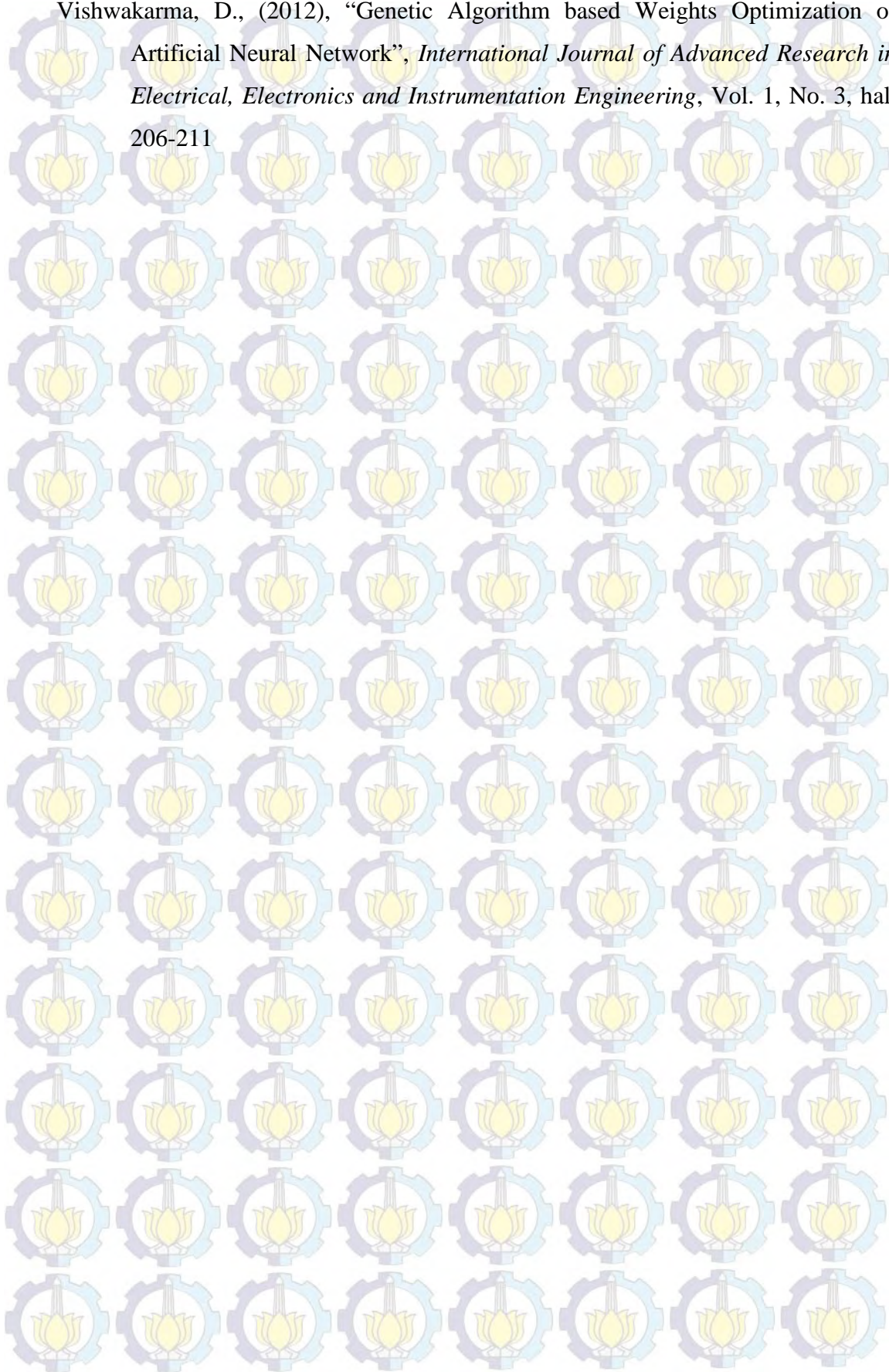
Mirjalili, S., Hashim, S., Sardroudi, H., (2012), "Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm", *Applied Mathematics and Computation*, Vol. 218, hal. 11125-11137

National Institute of Diabetes and Digestive and Kidney Disease (NIDDK). (2008), "Diagnosis of Diabetes", NIH Publication, No. 09 - 4642, hal. 1-12

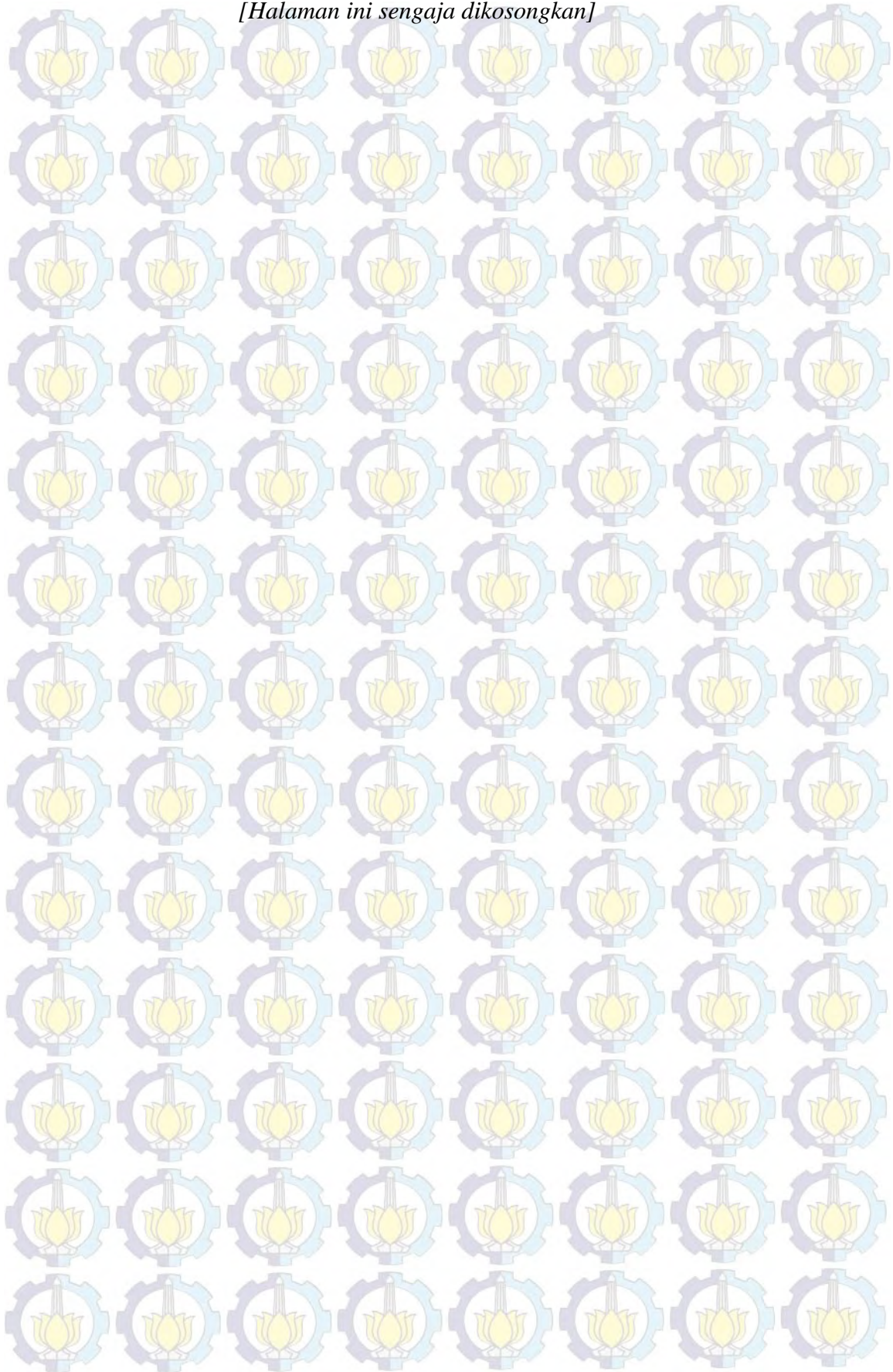
Nicoara, E.S, (2009), "Mechanisms to Avoid the Premature Convergence of Genetic Algorithms", *Matematic - Informatic - Fizic*, Vol. 59, No. 1, hal. 87-96

- Patel, H.P., (2006), "The Abnormal Urinalysis", *Pediatric Clinic of North America*, Vol 53, hal. 325– 337
- Pavelka, A. and Procházka, A. (2004). "Algorithms for Initialization of Neural Network Weights". *International Conference MATLAB*, Vol. 2, hal. 453–459.
- Pugia, (2004), "The uristatin dipstick is useful in distinguishing upper respiratory from urinary tract infections", *Clinica Chimica Acta*, Vol. 341, 73–81
- Rashedi, E., Nezamabadi-pour, H., Saryazdi, S., (2009), "GSA: A Gravitational Search Algorithm", *Information Science*, Vol. 179, 2232–2248
- Rojas, R., (1996), *Neural Networks : A Systematic Introduction*, Springer, Berlin
- Russell, S.J., Norvig, P., (1995), *Artificial Intelligence a Modern Approach*, Prentice Hall, Upper Saddle River, New Jersey
- Sarafrazi S., Nezamabadi-pour H., Saryazdi S. (2011), "Disruption: A new operator in gravitational search algorithm", *Scientia Iranica D*, Vol. 18, No. 3, hal. 539–548
- Sultana, (2001), "Dipstick urinalysis and the accuracy of the clinical diagnosis of urinary tract infection". *The Journal of Emergency Medicine*, Vol. 20, No. 1, hal. 13–19
- Tipler, (1998), *Fisika Untuk Sains Dan Teknik Jilid 1*, Erlangga, Jakarta
- Whiting, P. Westwood, M. Bojke, L. dkk. (2006), "Clinical effectiveness and cost-effectiveness of tests for the diagnosis and investigation of urinary tract infection in children: a systematic review and economic model", *Health Technol Assess*, Vol. 10, hal. 1-154.
- Wijaya, T.A., (2013), *Paduan Elemen Warna Sa\*b\* pada Analisa Urin Dipstick dari Citra Hasil Kamera Smartphone dengan Jaringan Backpropagation*, Tesis Magister, Institut Teknologi Sepuluh Nopember, Surabaya
- Wu, X., (2010), "Urinalysis: A Review of Methods and Procedures", *Critical Care Nursing Clinics of North America*, Vol 22, pp 121-128
- Yin, M., Hu, Y., Yang, F., Li, X., Gu, W., (2011), "A novel hybrid K-harmonic means and gravitational search algorithm approach for clustering", *Expert Systems with Applications*, Vol. 38, hal. 9319–9324

Vishwakarma, D., (2012), "Genetic Algorithm based Weights Optimization of Artificial Neural Network", *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, Vol. 1, No. 3, hal. 206-211



*[Halaman ini sengaja dikosongkan]*



## BIOGRAFI PENULIS

Abidatul Izzah merupakan anak kedua dari tiga bersaudara.

Ia lahir di Kediri pada tanggal 28 Oktober 1990. Izza, sapaan akrabnya, menempuh pendidikan bangku sekolah di

MI Al Irsyad Al Islamiyyah Kediri lulus pada tahun 2002,

MTsN 2 Kediri lulus pada tahun 2005, dan SMAN 2 Kediri

lulus pada tahun 2008. Kemudian ia melanjutkan

pendidikan Strata 1 di Program Studi Matematika,

Universitas Airlangga Surabaya dari tahun 2008 s.d. tahun

2012 dengan mengambil bidang minat Matematika

Terapan. Berminat di dunia pendidikan, pada tahun 2012 s.d. 2014, Izza

melanjutkan pendidikan Strata 2 di Jurusan Teknik Informatika, Fakultas Teknologi

Informasi (FTIF), Institut Teknologi Sepuluh Nopember (ITS) Surabaya dengan

mengambil bidang minat Komputasi Cerdas dan Visualisasi. Bidang penelitian yang

diminati oleh Izza adalah Kecerdasan Buatan, Sistem Cerdas, Optimasi, dan Data

Mining. Abidatul Izzah dapat dihubungi melalui email [aza.syaifa@gmail.com](mailto:aza.syaifa@gmail.com).



## DAFTAR LAMPIRAN

Lampiran 1	Perhitungan GSA (langkah 1 s.d. 3) .....	59
Lampiran 2	Perhitungan GSA (langkah 4) .....	61
Lampiran 3	Perhitungan GSA (langkah 5) .....	63
Lampiran 4	Perhitungan operator momentum (langkah 6) .....	65
Lampiran 5	Hasil pengujian pada kasus umum .....	67
Lampiran 6	Hasil pengujian GSA pada dataset urine .....	69
Lampiran 7	Hasil pengujian GSA pada dataset urine kelompok usia .....	71
Lampiran 8	Hasil pengujian GSA pada dataset urine kelompok jenis kelamin .....	73
Lampiran 9	Hasil pengujian GSAM pada dataset urine .....	75
Lampiran 10	Hasil pengujian GSAM pada dataset urine kelompok usia .....	77
Lampiran 11	Hasil pengujian GSAM pada dataset urine kelompok jenis kelamin .....	79



# LAMPIRAN 1

## PERHITUNGAN GSA (LANGKAH 1 s.d. 3)

### Langkah 1. Pembangkitan Agen

```

agen(:,1) =
-0.1614 0.3062 0.3404 0.2898 -0.3213 0.0597 -0.0548 -0.3155 0.2952 0.1745 -0.5635 0.5460 -0.1984
-0.3799 0.1919 -0.1846 0.4013 -0.3030 0.0840 -0.0301 -0.0766 0.3288 -0.2377 -0.1441 0.4815 -0.1984
0.0386 0.4025 0.1379 -0.4190 -0.0999 0.3363 -0.1004 0.2220 0.2168 -0.2430 0.0803 0.8154 -0.1984
0.3494 -0.2704 0.2723 -0.0735 0.2229 -0.1319 0.3227 0.2931 -0.3035 0.1504 0.0833 -0.0726 -0.1984
agen(:,2) =
0.3950 -0.3599 0.0523 -0.1625 0.3958 0.1058 -0.0317 0.3373 0.2071 0.0794 0.6508 -0.1727 0.2912
-0.3255 -0.3897 -0.3388 -0.0366 -0.2540 -0.3463 -0.1393 -0.2223 -0.1327 0.0427 -0.4937 -0.9934
0.2912
0.3504 -0.2041 -0.0747 0.3672 -0.0284 0.3625 -0.0012 0.3651 0.0285 -0.2746 0.6277 0.0917 0.2912
-0.1664 0.2370 -0.0873 0.4395 -0.4447 -0.1248 0.0674 0.2655 0.2418 -0.1229 -0.3889 -0.3100 0.2912
agen(:,3) =
0.3406 0.0691 -0.1887 0.3152 -0.3804 -0.2144 0.1709 -0.3155 0.1710 -0.2052 0.2908 -0.7429 0.6870
-0.0926 -0.2238 -0.3703 -0.2022 0.1418 -0.5074 0.1320 -0.3187 -0.1094 -0.0363 0.7282 -0.8404 0.6870
0.2269 -0.2122 -0.0895 0.3094 -0.2861 -0.3460 0.1442 -0.1535 0.3777 -0.2404 -0.2149 0.0880 0.6870
0.0620 0.1027 -0.0011 -0.0421 -0.3905 -0.2715 0.2501 0.5582 0.0622 0.1616 -0.0495 0.3446 0.6870
agen(:,4) =
0.0765 -0.3404 0.1848 -0.2694 -0.3227 -0.3508 -0.3264 -0.2230 -0.0867 -0.1649 -0.2001 0.9466 -0.4847
-0.3466 0.1324 -0.3823 0.2320 0.3860 -0.0565 -0.0910 -0.0853 0.3748 0.0269 0.3911 -0.6970 -0.4847
-0.3875 -0.3002 -0.3453 -0.1512 -0.1410 0.0538 -0.3837 0.2190 -0.1509 0.1531 0.6450 -0.7537 -0.4847
-0.0575 0.1980 -0.1978 -0.2398 -0.2673 0.3780 0.3653 0.3248 -0.0375 -0.2295 0.5797 -0.6124 -0.4847
agen(:,5) =
0.2087 -0.2530 0.0887 -0.2981 0.0688 -0.0375 0.3502 -0.4513 -0.1213 -0.3084 0.0376 -0.8028 -0.3390
-0.3472 0.3287 -0.0591 0.0381 -0.3384 0.0728 -0.0755 0.0922 -0.3766 -0.3704 -0.1547 0.2199 -0.3390
-0.0477 0.3577 0.2226 0.0559 -0.3348 0.1545 -0.5090 0.2471 -0.0847 0.0177 -0.7157 -0.5532 -0.3390
0.3902 0.2368 0.4384 0.2442 -0.0587 -0.0422 0.0110 -0.2527 -0.0572 -0.3375 -0.2182 0.1613 -0.3390

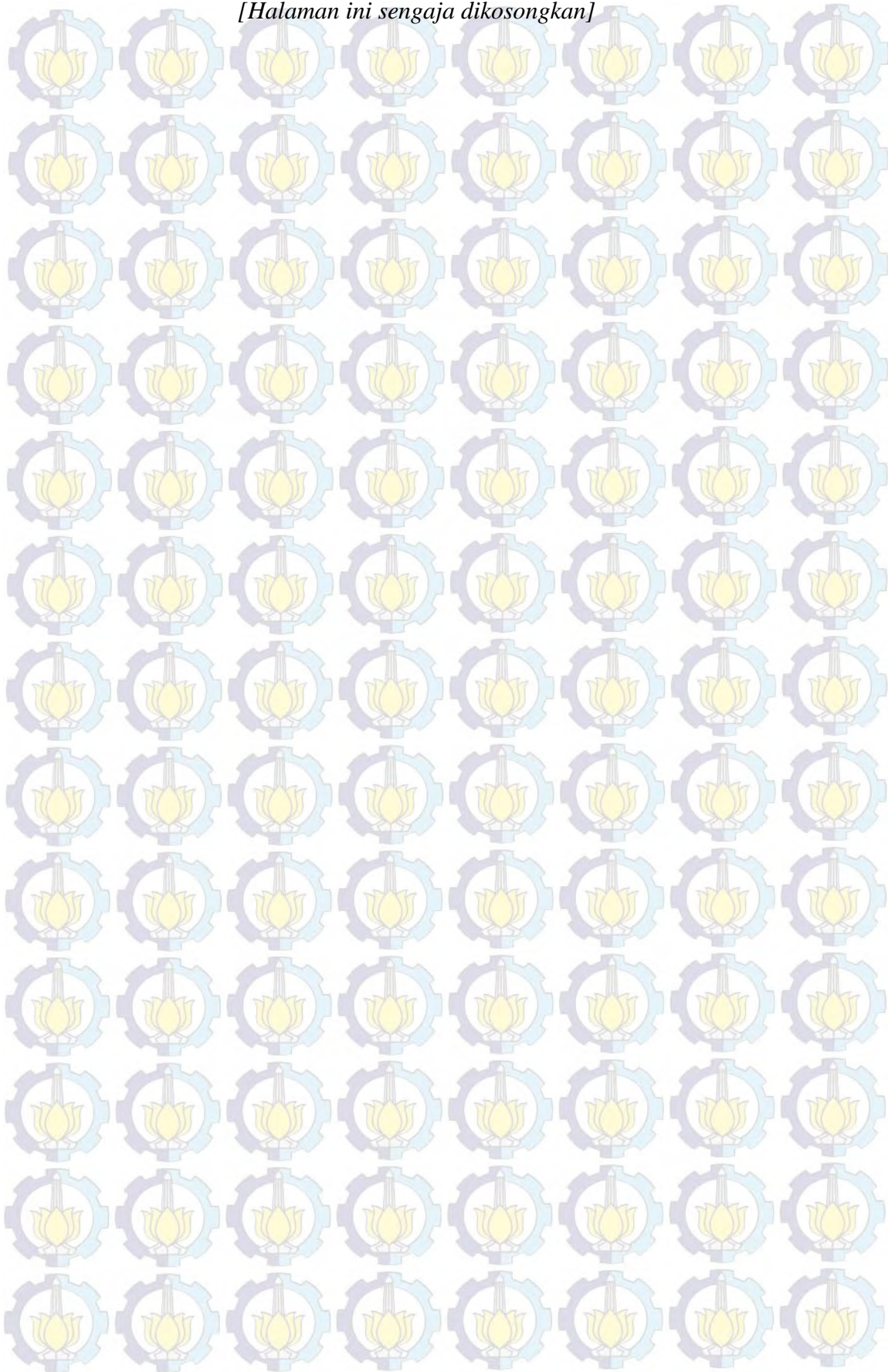
```

### Langkah 2. Evaluasi Minimum Square Error

Agen ke-	MSE
1.0000	0.5032
2.0000	0.4557
3.0000	0.5032
4.0000	0.4778
5.0000	0.4209

Langkah 3. Update G, best, worst  
G(t) pada iterasi ke-1 sebesar 0.12726  
Best MSE 0.42089  
Worst MSE 0.50316

*[Halaman ini sengaja dikosongkan]*



## LAMPIRAN 2

### PERHITUNGAN GSA (LANGKAH 4)

Langkah 4. Hitung massa, gaya, dan percepatan

>>Massa

MSE	m	M
0.5032	0	0
0.4557	0.5769	0.3061
0.5032	0	0
0.4778	0.3077	0.1633
0.4209	1.0000	0.5306

>>Gaya

Fr(:,1) =

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fr(:,2) =

0.0005	-0.0001	-0.0002	0.0002	0.0010	0.0006	0.0003	0.0009	0.0005	0.0004	0.0013	-0.0013	0.0012	
0.0000	-0.0009	-0.0000	-0.0004	-0.0008	-0.0005	-0.0001	-0.0003	-0.0006	0.0001	-0.0013	-0.0007	0.0012	
0.0011	-0.0000	0.0003	0.0008	0.0002	0.0005	0.0006	0.0002	0.0003	-0.0006	0.0003	0.0013	0.0012	
-0.0003	0.0001	0.0000	0.0010	-0.0003	-0.0007	-0.0004	0.0000	0.0004	0.0002	-0.0013	0.0003	0.0012	

Fr(:,3) =

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fr(:,4) =

-0.0005	-0.0003	0.0003	0.0000	-0.0014	-0.0011	-0.0021	0.0005	-0.0000	0.0003	-0.0010	0.0056	-0.0007	
-0.0000	-0.0004	-0.0010	0.0007	0.0024	-0.0003	-0.0000	-0.0005	0.0024	0.0012	0.0020	-0.0026	-0.0007	
-0.0013	-0.0020	-0.0018	-0.0008	0.0005	-0.0004	0.0002	-0.0001	-0.0003	0.0006	0.0041	-0.0009	-0.0007	
-0.0013	-0.0001	-0.0019	-0.0017	-0.0006	0.0014	0.0012	0.0017	-0.0000	0.0003	0.0027	-0.0024	-0.0007	

Fr(:,5) =

-0.0003	0.0007	-0.0002	-0.0006	-0.0000	0.0004	0.0036	-0.0037	-0.0014	-0.0019	-0.0016	-0.0080	-0.0020	
-0.0001	0.0034	0.0021	-0.0003	-0.0026	0.0020	0.0003	0.0018	-0.0033	-0.0028	-0.0004	0.0076	-0.0020	
-0.0004	0.0042	0.0029	-0.0005	-0.0018	-0.0005	-0.0023	-0.0004	-0.0002	0.0007	-0.0095	-0.0018	-0.0020	
0.0036	0.0001	0.0040	0.0008	0.0021	-0.0010	-0.0013	-0.0038	-0.0012	-0.0012	-0.0019	0.0043	-0.0020	

>>Percepatan

acc(:,1) =

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

acc(:,2) =

0.0015	-0.0002	-0.0006	0.0006	0.0034	0.0021	0.0010	0.0031	0.0015	0.0014	0.0042	-0.0044	0.0039	
0.0001	-0.0028	-0.0000	-0.0012	-0.0027	-0.0016	-0.0003	-0.0008	-0.0020	0.0004	-0.0041	-0.0022	0.0039	
0.0035	-0.0000	0.0010	0.0025	0.0007	0.0015	0.0021	0.0007	0.0009	-0.0021	0.0010	0.0042	0.0039	
-0.0009	0.0002	0.0001	0.0031	-0.0011	-0.0023	-0.0013	0.0001	0.0015	0.0006	-0.0044	0.0010	0.0039	

acc(:,3) =

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

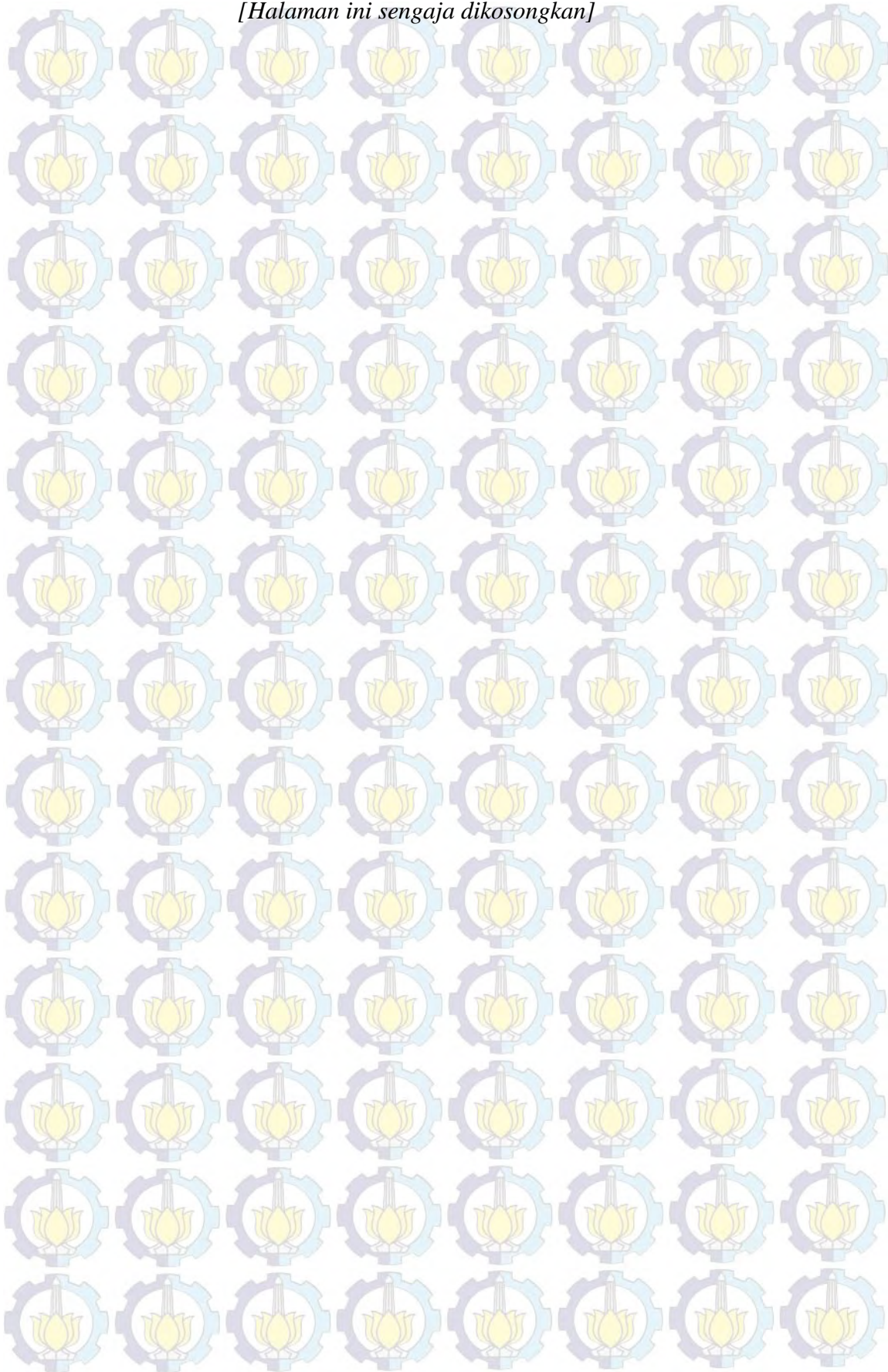
acc(:,4) =

-0.0031	-0.0016	0.0021	0.0003	-0.0088	-0.0067	-0.0130	0.0029	-0.0000	0.0021	-0.0062	0.0345	-0.0044	
-0.0000	-0.0024	-0.0060	0.0041	0.0147	-0.0017	-0.0002	-0.0029	0.0149	0.0072	0.0120	-0.0161	-0.0044	
-0.0079	-0.0123	-0.0110	-0.0049	0.0033	-0.0025	0.0014	-0.0008	-0.0016	0.0034	0.0250	-0.0055	-0.0044	
-0.0080	-0.0008	-0.0119	-0.0104	-0.0034	0.0088	0.0071	0.0107	-0.0003	0.0017	0.0168	-0.0148	-0.0044	

acc(:,5) =

-0.0006	0.0013	-0.0003	-0.0012	-0.0000	0.0008	0.0068	-0.0071	-0.0026	-0.0037	-0.0030	-0.0150	-0.0037	
-0.0002	0.0064	0.0040	-0.0006	-0.0049	0.0038	0.0006	0.0033	-0.0063	-0.0054	-0.0008	0.0143	-0.0037	
-0.0008	0.0080	0.0056	-0.0010	-0.0034	-0.0009	-0.0044	-0.0007	-0.0004	0.0013	-0.0179	-0.0035	-0.0037	
0.0067	0.0002	0.0076	0.0015	0.0040	-0.0019	-0.0025	-0.0072	-0.0023	-0.0022	-0.0035	0.0080	-0.0037	

*[Halaman ini sengaja dikosongkan]*



## LAMPIRAN 3

### PERHITUNGAN GSA (LANGKAH 5)

Langkah 5. Hitung kecepatan dan perpindahan posisi

>>Kecepatan

velocity(:,1) =

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

velocity(:,2) =

```
0.0015 -0.0002 -0.0006 0.0006 0.0034 0.0021 0.0010 0.0031 0.0015 0.0014 0.0042 -0.0044 0.0039
0.0001 -0.0028 -0.0000 -0.0012 -0.0027 -0.0016 -0.0003 -0.0008 -0.0020 0.0004 -0.0041 -0.0022 0.0039
0.0035 -0.0000 0.0010 0.0025 0.0007 0.0015 0.0021 0.0007 0.0009 -0.0021 0.0010 0.0042 0.0039
-0.0009 0.0002 0.0001 0.0031 -0.0011 -0.0023 -0.0013 0.0001 0.0015 0.0006 -0.0044 0.0010 0.0039
```

velocity(:,3) =

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

velocity(:,4) =

```
-0.0031 -0.0016 0.0021 0.0003 -0.0088 -0.0067 -0.0130 0.0029 -0.0000 0.0021 -0.0062 0.0345 -0.0044
-0.0000 -0.0024 -0.0060 0.0041 0.0147 -0.0017 -0.0002 -0.0029 0.0149 0.0072 0.0120 -0.0161 -0.0044
-0.0079 -0.0123 -0.0110 -0.0049 0.0033 -0.0025 0.0014 -0.0008 -0.0016 0.0034 0.0250 -0.0055 -0.0044
-0.0080 -0.0008 -0.0119 -0.0104 -0.0034 0.0088 0.0071 0.0107 -0.0003 0.0017 0.0168 -0.0148 -0.0044
```

velocity(:,5) =

```
-0.0006 0.0013 -0.0003 -0.0012 -0.0000 0.0008 0.0068 -0.0071 -0.0026 -0.0037 -0.0030 -0.0150 -0.0037
-0.0002 0.0064 0.0040 -0.0006 -0.0049 0.0038 0.0006 0.0033 -0.0063 -0.0054 -0.0008 0.0143 -0.0037
-0.0008 0.0080 0.0056 -0.0010 -0.0034 -0.0009 -0.0044 -0.0007 -0.0004 0.0013 -0.0179 -0.0035 -0.0037
0.0067 0.0002 0.0076 0.0015 0.0040 -0.0019 -0.0025 -0.0072 -0.0023 -0.0022 -0.0035 0.0080 -0.0037
```

>>Posisi Baru

agen(:,1) =

```
-0.1614 0.3062 0.3404 0.2898 -0.3213 0.0597 -0.0548 -0.3155 0.2952 0.1745 -0.5635 0.5460 -0.1984
-0.3799 0.1919 -0.1846 0.4013 -0.3030 0.0840 -0.0301 -0.0766 0.3288 -0.2377 -0.1441 0.4815 -0.1984
0.0386 0.4025 0.1379 -0.4190 -0.0999 0.3363 -0.1004 0.2220 0.2168 -0.2430 0.0803 0.8154 -0.1984
0.3494 -0.2704 0.2723 -0.0735 0.2229 -0.1319 0.3227 0.2931 -0.3035 0.1504 0.0833 -0.0726 -0.1984
```

agen(:,2) =

```
0.3950 -0.3599 0.0523 -0.1625 0.3958 0.1058 -0.0317 0.3373 0.2071 0.0794 0.6508 -0.1727 0.2912
-0.3255 -0.3897 -0.3388 -0.0366 -0.2540 -0.3463 -0.1393 -0.2223 -0.1327 0.0427 -0.4937 -0.9934
0.2912
0.3504 -0.2041 -0.0747 0.3672 -0.0284 0.3625 -0.0012 0.3651 0.0285 -0.2746 0.6277 0.0917 0.2912
-0.1664 0.2370 -0.0873 0.4395 -0.4447 -0.1248 0.0674 0.2655 0.2418 -0.1229 -0.3889 -0.3100 0.2912
```

agen(:,3) =

```
0.3406 0.0691 -0.1887 0.3152 -0.3804 -0.2144 0.1709 -0.3155 0.1710 -0.2052 0.2908 -0.7429 0.6870
-0.0926 -0.2238 -0.3703 -0.2022 0.1418 -0.5074 0.1320 -0.3187 -0.1094 -0.0363 0.7282 -0.8404 0.6870
0.2269 -0.2122 -0.0895 0.3094 -0.2861 -0.3460 0.1442 -0.1535 0.3777 -0.2404 -0.2149 0.0880 0.6870
0.0620 0.1027 -0.0011 -0.0421 -0.3905 -0.2715 0.2501 0.5582 0.0622 0.1616 -0.0495 0.3446 0.6870
```

agen(:,4) =

```
0.0765 -0.3404 0.1848 -0.2694 -0.3227 -0.3508 -0.3264 -0.2230 -0.0867 -0.1649 -0.2001 0.9466 -0.4847
-0.3466 0.1324 -0.3823 0.2320 0.3860 -0.0565 -0.0910 -0.0853 0.3748 0.0269 0.3911 -0.6970 -0.4847
-0.3875 -0.3002 -0.3453 -0.1512 -0.1410 0.0538 -0.3837 0.2190 -0.1509 0.1531 0.6450 -0.7537 -0.4847
-0.0575 0.1980 -0.1978 -0.2398 -0.2673 0.3780 0.3653 0.3248 -0.0375 -0.2295 0.5797 -0.6124 -0.4847
```

agen(:,5) =

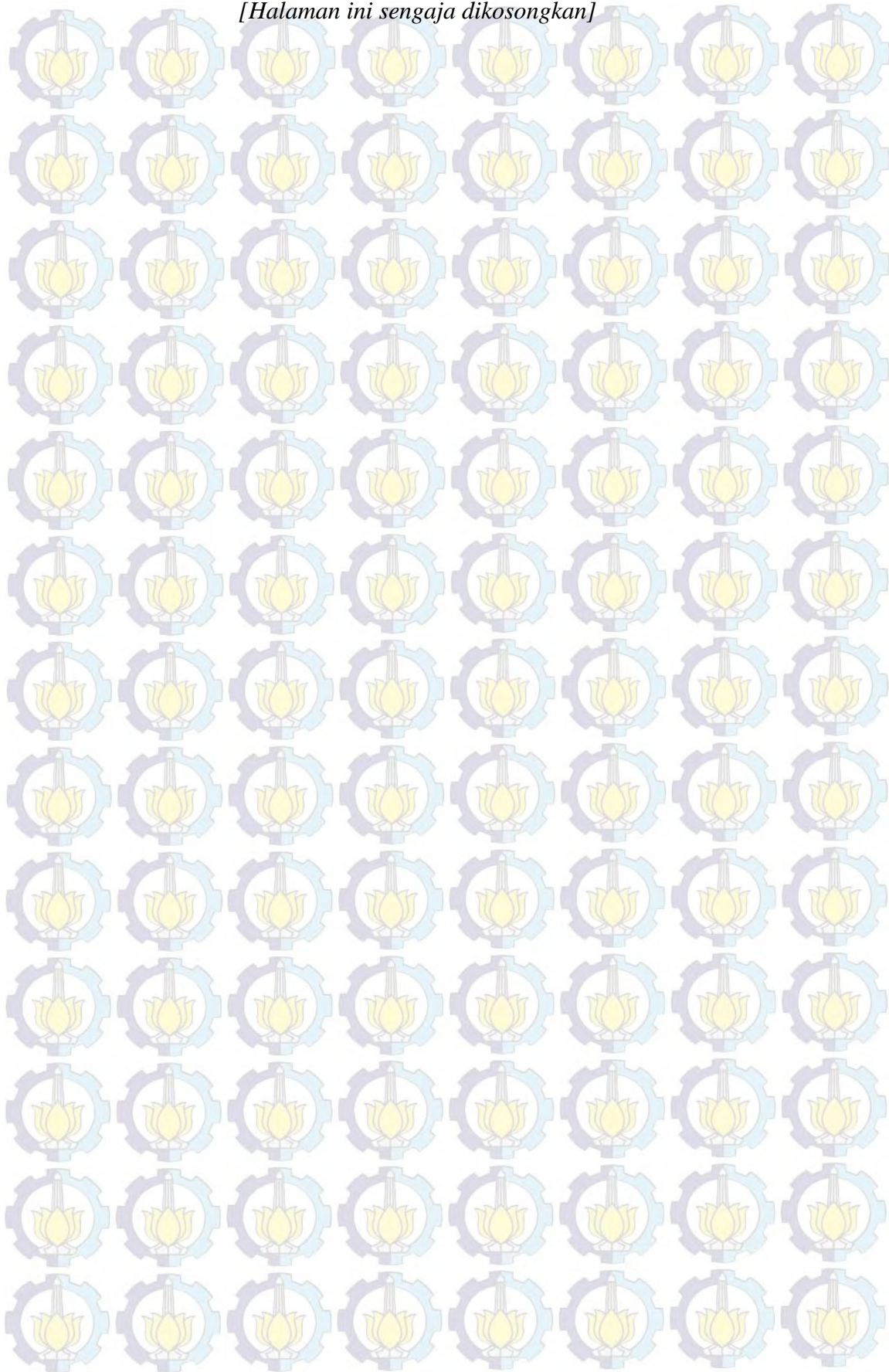
```
0.2087 -0.2530 0.0887 -0.2981 0.0688 -0.0375 0.3502 -0.4513 -0.1213 -0.3084 0.0376 -0.8028 -0.3390
-0.3472 0.3287 -0.0591 0.0381 -0.3384 0.0728 -0.0755 0.0922 -0.3766 -0.3704 -0.1547 0.2199 -0.3390
-0.0477 0.3577 0.2226 0.0559 -0.3348 0.1545 -0.5090 0.2471 -0.0847 0.0177 -0.7157 -0.5532 -0.3390
0.3902 0.2368 0.4384 0.2442 -0.0587 -0.0422 0.0110 -0.2527 -0.0572 -0.3375 -0.2182 0.1613 -0.3390
```

>>Evaluasi Posisi Baru

Agen ke- MSE

```
-----
1.0000 0.5032
2.0000 0.4557
3.0000 0.5032
4.0000 0.4778
5.0000 0.4209
```

*[Halaman ini sengaja dikosongkan]*



## LAMPIRAN 4

### PERHITUNGAN OPERATOR MOMENTUM (LANGKAH 6)

Langkah 6. Operator Momentum

>>Kecepatan setelah tumbukan

velocity(:,1) =

```
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
```

velocity(:,2) =

```
0.0015 -0.0002 -0.0006 0.0006 0.0034 0.0021 0.0010 0.0031 0.0015 0.0014 0.0042 -0.0044 0.0039
0.0001 -0.0028 -0.0000 -0.0012 -0.0027 -0.0016 -0.0003 -0.0008 -0.0020 0.0004 -0.0041 -0.0022 0.0039
0.0035 -0.0000 0.0010 0.0025 0.0007 0.0015 0.0021 0.0007 0.0009 -0.0021 0.0010 0.0042 0.0039
-0.0009 0.0002 0.0001 0.0031 -0.0011 -0.0023 -0.0013 0.0001 0.0015 0.0006 -0.0044 0.0010 0.0039
```

velocity(:,3) =

```
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
```

velocity(:,4) =

```
-0.0031 -0.0016 0.0021 0.0003 -0.0088 -0.0067 -0.0130 0.0029 -0.0000 0.0021 -0.0062 0.0345 -0.0044
-0.0000 -0.0024 -0.0060 0.0041 0.0147 -0.0017 -0.0002 -0.0029 0.0149 0.0072 0.0120 -0.0161 -0.0044
-0.0079 -0.0123 -0.0110 -0.0049 0.0033 -0.0025 0.0014 -0.0008 -0.0016 0.0034 0.0250 -0.0055 -0.0044
-0.0080 -0.0008 -0.0119 -0.0104 -0.0034 0.0088 0.0071 0.0107 -0.0003 0.0017 0.0168 -0.0148 -0.0044
```

velocity(:,5) =

```
-0.0006 0.0013 -0.0003 -0.0012 -0.0000 0.0008 0.0068 -0.0071 -0.0026 -0.0037 -0.0030 -0.0150 -0.0037
-0.0002 0.0064 0.0040 -0.0006 -0.0049 0.0038 0.0006 0.0033 -0.0063 -0.0054 -0.0008 0.0143 -0.0037
-0.0008 0.0080 0.0056 -0.0010 -0.0034 -0.0009 -0.0044 -0.0007 -0.0004 0.0013 -0.0179 -0.0035 -0.0037
0.0067 0.0002 0.0076 0.0015 0.0040 -0.0019 -0.0025 -0.0072 -0.0023 -0.0022 -0.0035 0.0080 -0.0037
```

>>Update Posisi Baru

agen(:,1) =

```
-0.1625 0.3088 0.3397 0.2875 -0.3213 0.0614 -0.0412 -0.3296 0.2900 0.1672 -0.5695 0.5159 -0.2057
-0.3802 0.2047 -0.1767 0.4001 -0.3129 0.0916 -0.0290 -0.0699 0.3163 -0.2485 -0.1458 0.5100 -0.2057
0.0370 0.4185 0.1490 -0.4210 -0.1067 0.3345 -0.1093 0.2207 0.2160 -0.2404 0.0446 0.8084 -0.2057
0.3628 -0.2699 0.2875 -0.0705 0.2310 -0.1357 0.3176 0.2786 -0.3081 0.1460 0.0762 -0.0565 -0.2057
```

agen(:,2) =

```
0.3950 -0.3599 0.0523 -0.1625 0.3958 0.1058 -0.0317 0.3373 0.2071 0.0794 0.6508 -0.1727 0.2912
-0.3255 -0.3897 -0.3388 -0.0366 -0.2540 -0.3463 -0.1393 -0.2223 -0.1327 0.0427 -0.4937 -0.9934
0.2912
```

agen(:,3) =

```
0.3504 -0.2041 -0.0747 0.3672 -0.0284 0.3625 -0.0012 0.3651 0.0285 -0.2746 0.6277 0.0917 0.2912
-0.1664 0.2370 -0.0873 0.4395 -0.4447 -0.1248 0.0674 0.2655 0.2418 -0.1229 -0.3889 -0.3100 0.2912
0.3406 0.0691 -0.1887 0.3152 -0.3804 -0.2144 0.1709 -0.3155 0.1710 -0.2052 0.2908 -0.7429 0.6870
-0.0926 -0.2238 -0.3703 -0.2022 0.1418 -0.5074 0.1320 -0.3187 -0.1094 -0.0363 0.7282 -0.8404 0.6870
0.2269 -0.2122 -0.0895 0.3094 -0.2861 -0.3460 0.1442 -0.1535 0.3777 -0.2404 -0.2149 0.0880 0.6870
0.0620 0.1027 -0.0011 -0.0421 -0.3905 -0.2715 0.2501 0.5582 0.0622 0.1616 -0.0495 0.3446 0.6870
```

agen(:,4) =

```
0.0773 -0.3375 0.1833 -0.2713 -0.3181 -0.3459 -0.3090 -0.2354 -0.0906 -0.1716 -0.2014 0.9053 -0.4880
-0.3469 0.1434 -0.3731 0.2289 0.3706 -0.0498 -0.0900 -0.0786 0.3573 0.0149 0.3835 -0.6666 -0.4880
-0.3846 -0.2815 -0.3310 -0.1501 -0.1479 0.0538 -0.3912 0.2184 -0.1507 0.1533 0.6045 -0.7560 -0.4880
-0.0430 0.1988 -0.1799 -0.2320 -0.2593 -0.3704 0.3577 0.3082 -0.0408 -0.2338 0.5654 -0.5922 -0.4880
```

agen(:,5) =

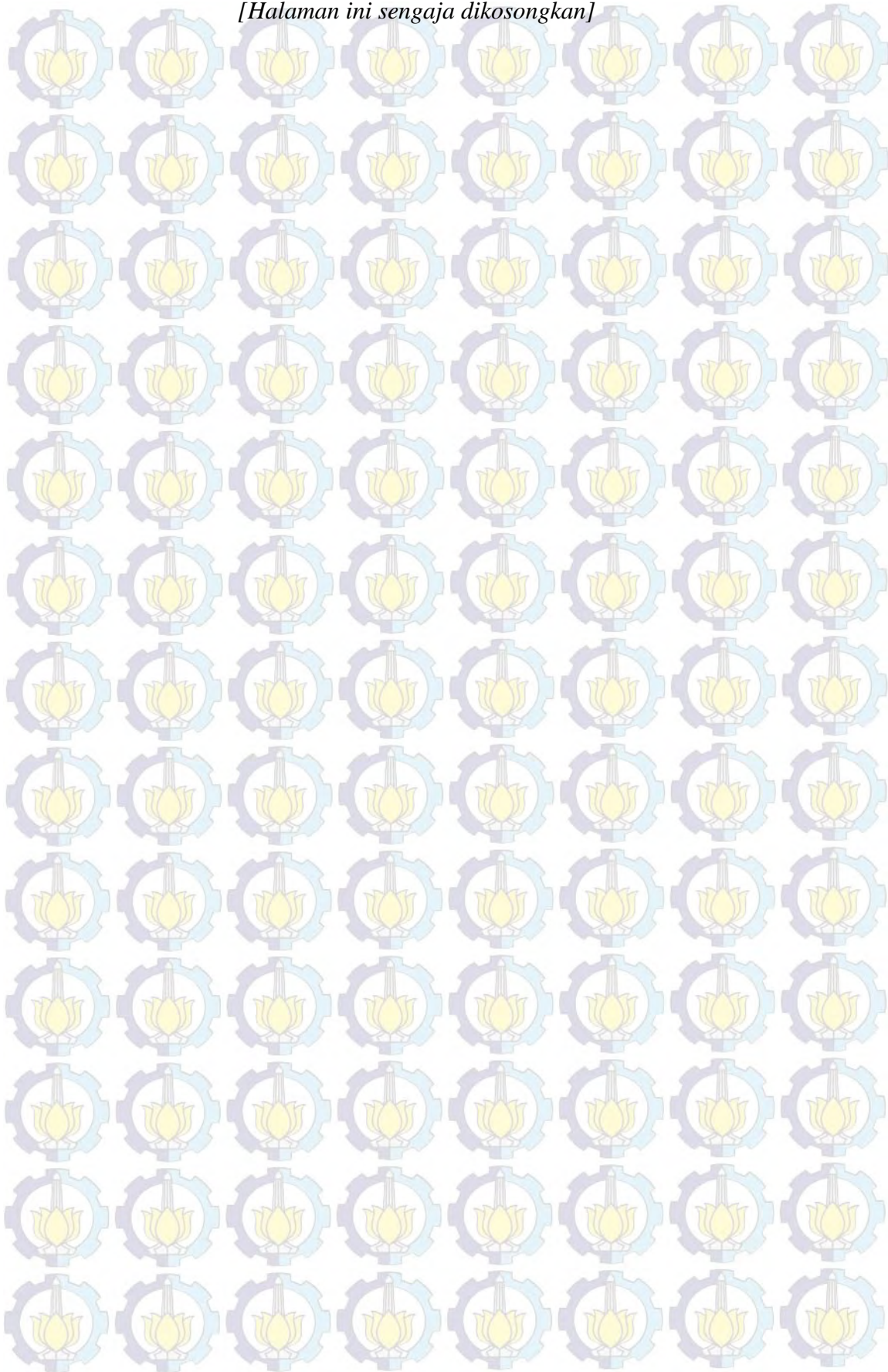
```
0.2087 -0.2530 0.0887 -0.2981 0.0688 -0.0375 0.3502 -0.4513 -0.1213 -0.3084 0.0376 -0.8028 -0.3390
-0.3472 0.3287 -0.0591 0.0381 -0.3384 0.0728 -0.0755 0.0922 -0.3766 -0.3704 -0.1547 0.2199 -0.3390
-0.0477 0.3577 0.2226 0.0559 -0.3348 0.1545 -0.5090 0.2471 -0.0847 0.0177 -0.7157 -0.5532 -0.3390
0.3902 0.2368 0.4384 0.2442 -0.0587 -0.0422 0.0110 -0.2527 -0.0572 -0.3375 -0.2182 0.1613 -0.3390
```

>>Evaluasi Posisi Baru

Agen ke- MSE

```
-----
1.0000 0.5000 ---> terupdate
2.0000 0.4557
3.0000 0.5032
4.0000 0.4747 ---> terupdate
5.0000 0.4209
```

*[Halaman ini sengaja dikosongkan]*

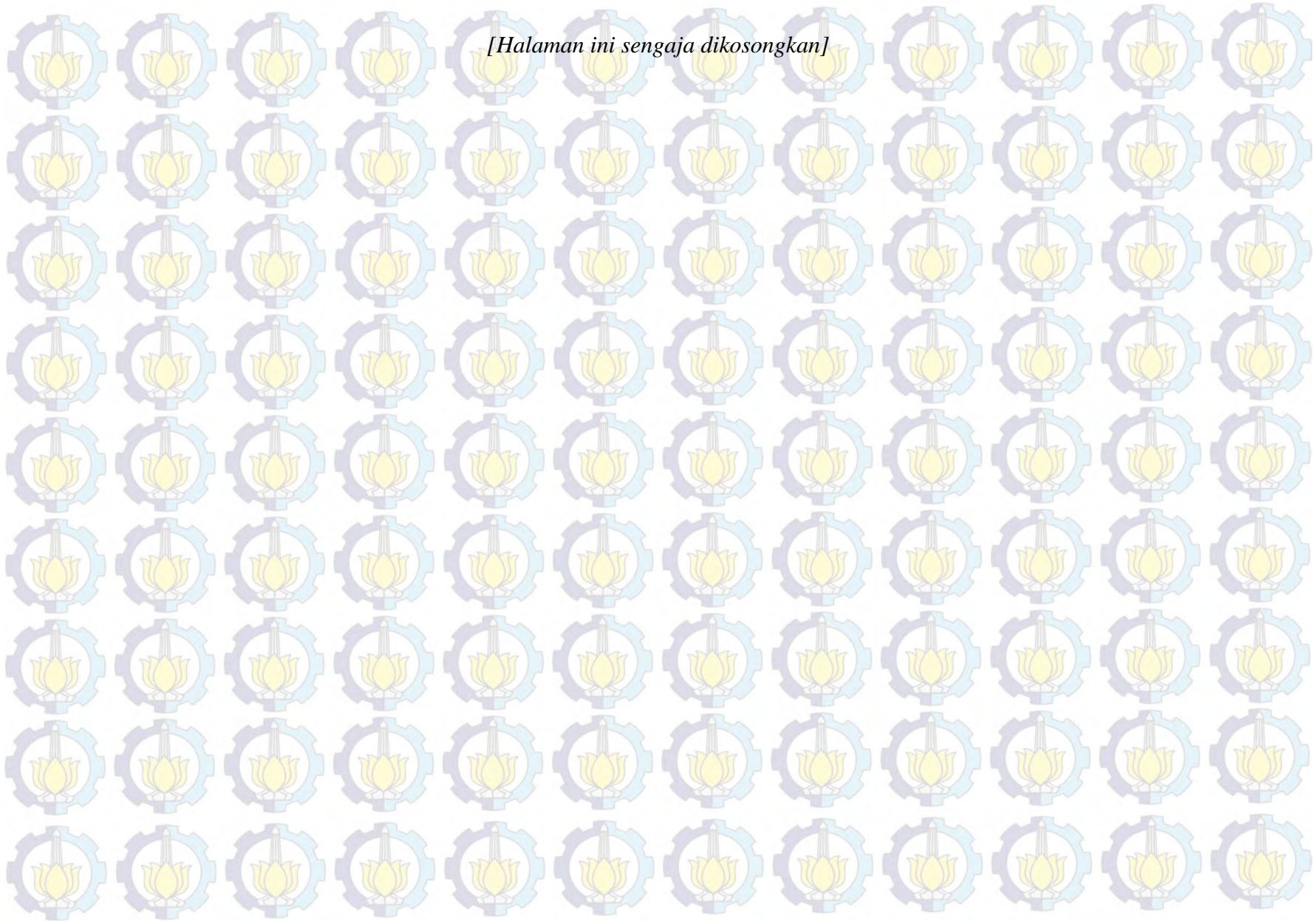




**LAMPIRAN 5**  
**HASIL PENGUJIAN PADA KASUS UMUM**

Dataset	Maksimum Iterasi	Jumlah Node Hidden	GSA		GSAM	
			Akurasi	Waktu Komputasi	Akurasi	Waktu Komputasi
Wine	50	7	0.66174	7.99752	0.64476	13.7235
		11	0.69571	10.4562	0.81968	15.5046
		15	0.68571	11.8086	0.71428	13.9792
	100	7	0.73111	13.1261	0.74714	11.3204
		11	0.62508	12.9934	0.65254	18.8536
		15	0.73587	16.8458	0.76428	31.0256
XOR 3 BIT	200	7	0.65603	8.38834	0.76381	14.7589
		11	0.70793	16.0400	0.70206	21.7647
		15	0.59603	17.8633	0.70222	17.8633
	50	7	0.75000	0.91015	0.80000	2.03334
		11	0.75000	1.24624	0.75000	2.02084
		15	0.75000	1.38332	0.77500	2.60370
100	7	0.75000	0.86524	0.80000	2.05214	
	11	0.75000	1.12570	0.87500	2.62530	
	15	0.82500	1.51328	0.77500	2.76538	
200	7	0.75000	0.88975	0.80000	1.74528	
	11	0.77500	1.14404	0.80000	2.23236	
		15	0.80000	1.57026	0.80000	1.57026

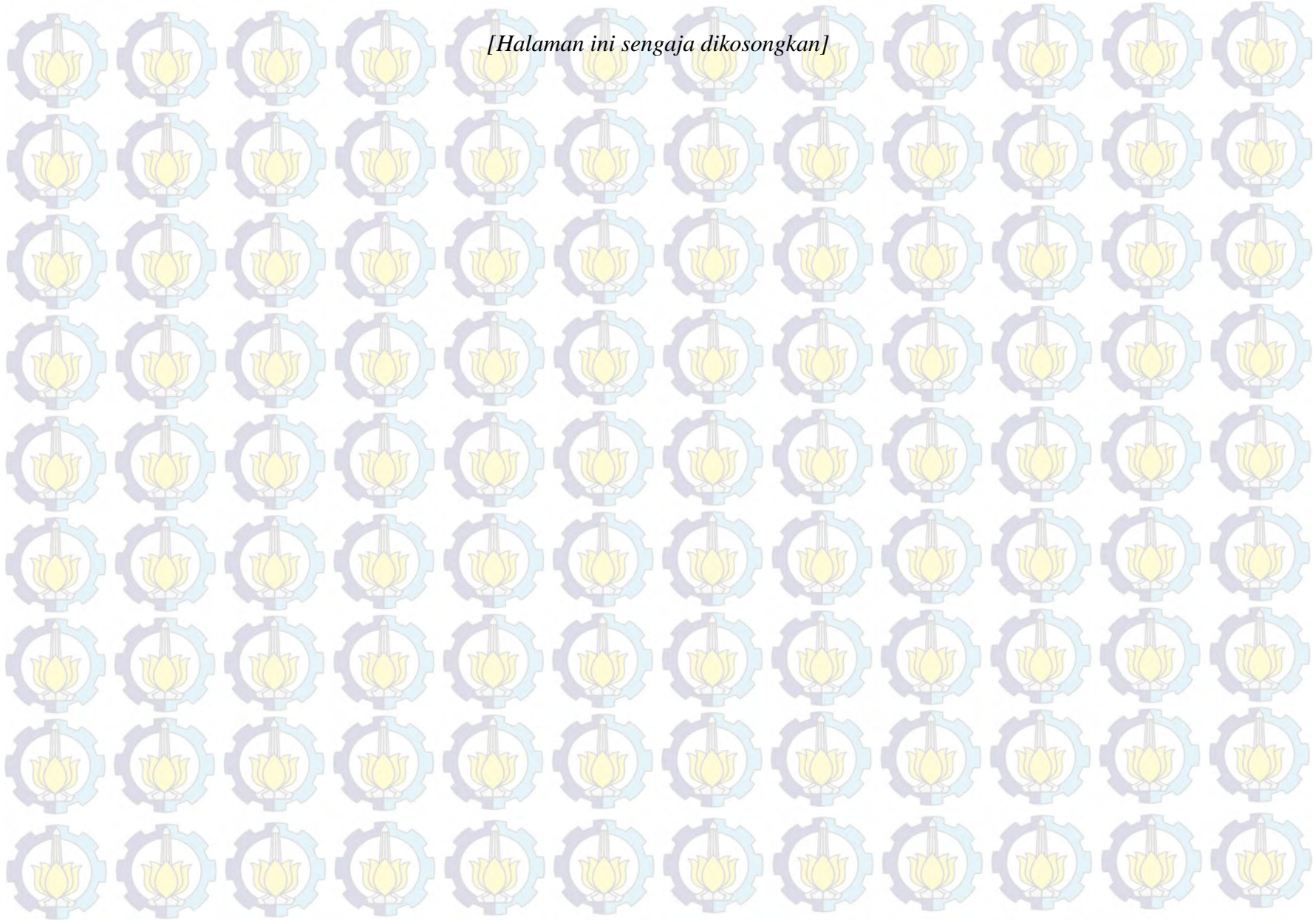
*[Halaman ini sengaja dikosongkan]*



**LAMPIRAN 6**  
**HASIL PENGUJIAN GSA PADA DATASET URINE**

Metode	Maksimum Iterasi	Jumlah Node Hidden	Rule Pelabelan 1				Rule Pelabelan 2			
			Akurasi	Sensitifitas	Spesifisitas	Waktu Komputasi	Akurasi	Sensitifitas	Spesifisitas	Waktu Komputasi
<b>3 fold</b>	50	7	0.69191	0.41919	0.96465	6.13626	0.47863	0.22067	0.96060	12.6107
		11	0.66414	0.42424	0.90404	15.6513	0.50427	0.29239	0.87359	19.0254
		15	0.64898	0.51515	0.78282	19.7029	0.46154	0.38702	0.67922	29.5210
		7	0.70454	0.53030	0.87879	11.0967	0.46837	0.22295	0.91998	12.7574
	100	11	0.73232	0.53535	0.92929	17.8805	0.48205	0.21711	0.96649	25.1916
		15	0.61111	0.67171	0.55050	19.3927	0.48204	0.26920	0.85557	37.9077
		7	0.70959	0.55555	0.86363	9.33466	0.45470	0.16435	0.90993	12.9385
		11	0.69444	0.56565	0.82323	15.9721	0.44102	0.45619	0.56439	17.0334
	200	15	0.70959	0.54040	0.87878	17.8670	0.48034	0.21400	0.93993	21.0960
		7	0.62609	0.43243	0.81974	30.2902	0.40655	0.13222	0.92459	19.6472
		11	0.67442	0.61641	0.73243	20.8491	0.43063	0.18409	0.87470	31.2001
		15	0.65615	0.45525	0.85705	28.8078	0.45616	0.19587	0.90138	35.9934
<b>5 fold</b>	50	7	0.68237	0.63038	0.73436	13.2797	0.39841	0.44875	0.55457	19.1682
		11	0.69698	0.48948	0.90448	25.0824	0.47994	0.21624	0.96698	33.7774
		15	0.67634	0.57641	0.77628	31.7305	0.39375	0.35481	0.58248	50.9393
		7	0.63609	0.60653	0.66564	10.2809	0.43776	0.38139	0.58794	19.8387
	100	11	0.68410	0.51974	0.84846	13.4255	0.43091	0.26500	0.80944	26.3191
		15	0.65205	0.52000	0.78410	18.4242	0.45297	0.17933	0.95433	22.4528
		7	0.65071	0.48228	0.82447	33.8103	0.45124	0.19775	0.90190	13.7918
		11	0.69334	0.50087	0.88947	42.4488	0.45169	0.30684	0.81246	20.1447
	200	15	0.69744	0.59026	0.78763	80.5285	0.48380	0.33574	0.78113	26.5364
		7	0.70171	0.55254	0.83921	32.5191	0.44351	0.23737	0.83863	16.1809
		11	0.66952	0.58657	0.75947	42.1366	0.45990	0.29156	0.83841	26.1037
		15	0.67507	0.48026	0.86289	55.1775	0.43567	0.25214	0.85690	33.8991
<b>10 fold</b>	100	7	0.69510	0.54552	0.84368	35.3311	0.45146	0.18813	0.95161	27.5635
		11	0.64031	0.51780	0.76315	33.3257	0.43889	0.29011	0.77711	31.9392
	200	11	0.64031	0.51780	0.76315	33.3257	0.43889	0.29011	0.77711	31.9392
		15	0.68260	0.59210	0.77210	48.3645	0.50258	0.29325	0.89825	59.9979

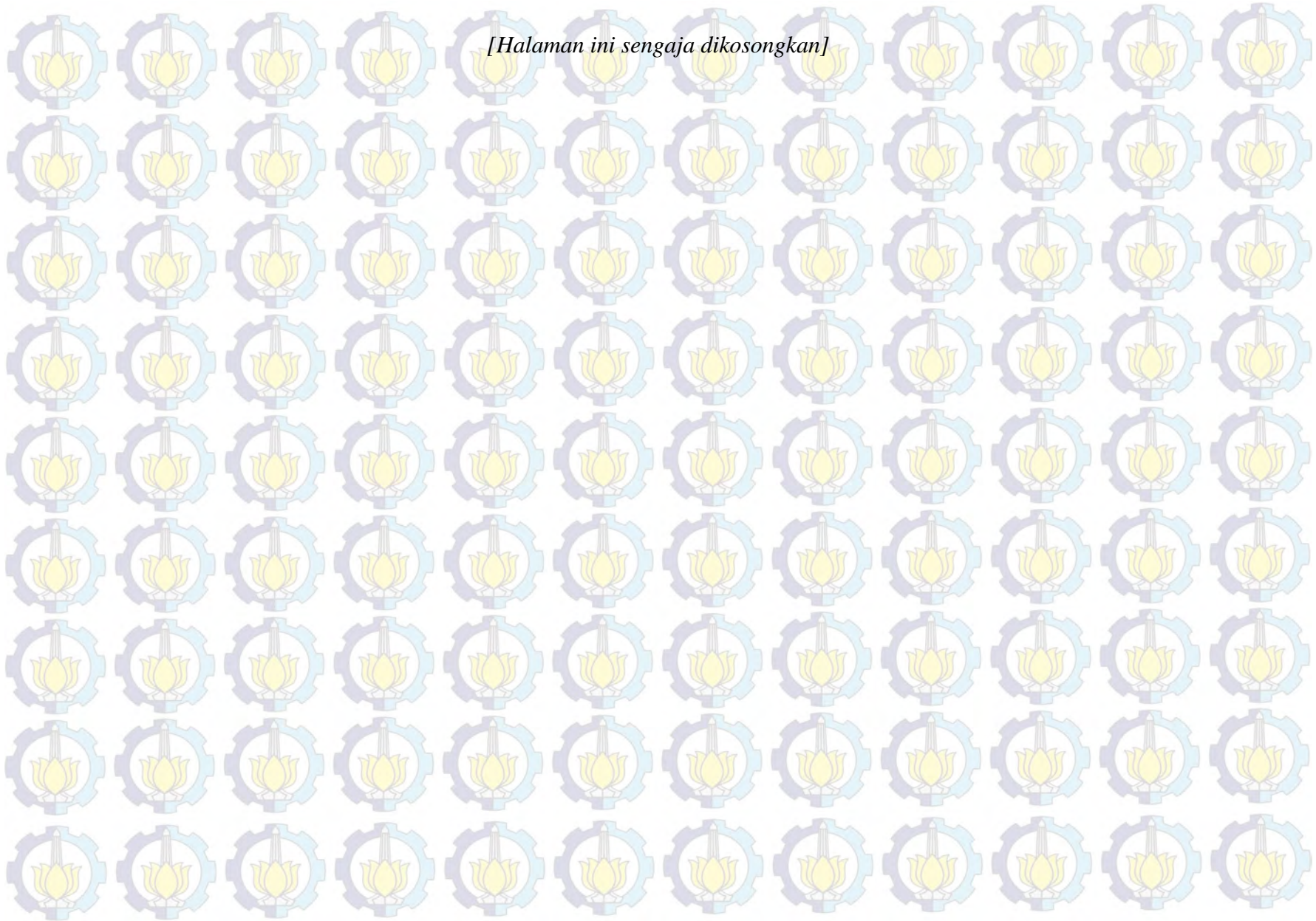
*[Halaman ini sengaja dikosongkan]*



**LAMPIRAN 7**  
**HASIL PENGUJIAN GSA PADA DATASET URINE KELOMPOK USIA**

Kelompok Usia	Maksimum Iterasi	Jumlah Node Hidden	Rule Pelabelan 1				Rule Pelabelan 2			
			Akurasi	Sensitifitas	Spesifisitas	Waktu Komputasi	Akurasi	Sensitifitas	Spesifisitas	Waktu Komputasi
< 20 tahun	50	7	0.78333	0.60000	1.00000	1.12164	0.47000	0.29999	0.70000	1.54690
		11	0.60000	0.50000	0.70000	1.34408	0.32000	0.18333	0.62500	1.55508
		15	0.68333	0.60000	0.80000	1.48121	0.47000	0.39999	0.60000	1.81606
	100	7	0.73333	0.60000	0.90000	1.15434	0.42000	0.51666	0.37500	1.51756
		11	0.78333	0.70000	0.90000	1.47068	0.42000	0.31666	0.75000	1.71498
		15	0.61666	0.50000	0.80000	1.88430	0.27000	0.28333	0.25000	2.09744
200	7	0.68333	0.60000	0.80000	0.67303	0.57000	0.50000	0.70000	1.73874	
	11	0.74000	0.62500	0.88571	0.58652	0.42000	0.26666	0.80000	2.01418	
		15	0.50000	0.50000	0.50000	1.06372	0.67000	0.43333	1.00000	1.80764
	20 tahun.d 50 tahun	50	7	0.77568	0.60769	0.88047	5.08218	0.45455	0.14099	0.91033
11			0.75748	0.68461	0.80190	7.22752	0.46465	0.12734	0.92009	10.9719
15			0.67335	0.62307	0.70381	10.0565	0.46822	0.19738	0.84110	16.8567
100		7	0.78743	0.63717	0.88047	4.78452	0.41958	0.34652	0.68286	10.2858
		11	0.72680	0.59230	0.81142	10.7962	0.45579	0.28270	0.77637	16.0896
		15	0.73891	0.68333	0.77285	12.4737	0.51330	0.19192	0.93938	19.0964
200	7	0.74020	0.54487	0.86047	3.89358	0.40985	0.08079	0.91225	9.79308	
	11	0.67939	0.53077	0.77476	7.00406	0.48571	0.20405	0.91756	23.0861	
		15	0.70257	0.55769	0.79142	9.14024	0.40849	0.17434	0.76881	24.1991
	> 50 tahun	50	7	0.56799	0.56466	0.57777	5.60682	0.39798	0.19716	0.90726
11			0.54474	0.49500	0.62026	9.41586	0.46038	0.34396	0.78840	13.2569
15			0.54961	0.49366	0.63202	11.8201	0.47805	0.35364	0.73818	13.4878
100		7	0.62392	0.54300	0.74313	6.12376	0.48155	0.34260	0.82727	12.4271
		11	0.61506	0.57166	0.67516	11.1230	0.42694	0.28159	0.80806	12.6877
		15	0.61495	0.64733	0.57647	15.4513	0.51077	0.33085	0.86136	23.9110
200	7	0.63640	0.40064	0.82690	6.77470	0.48876	0.30398	0.85552	5.74332	
	11	0.61858	0.37820	0.79142	11.1628	0.58013	0.20461	0.79587	20.5976	
		15	0.61894	0.43974	0.75333	16.2132	0.48889	0.17976	0.78623	14.3165

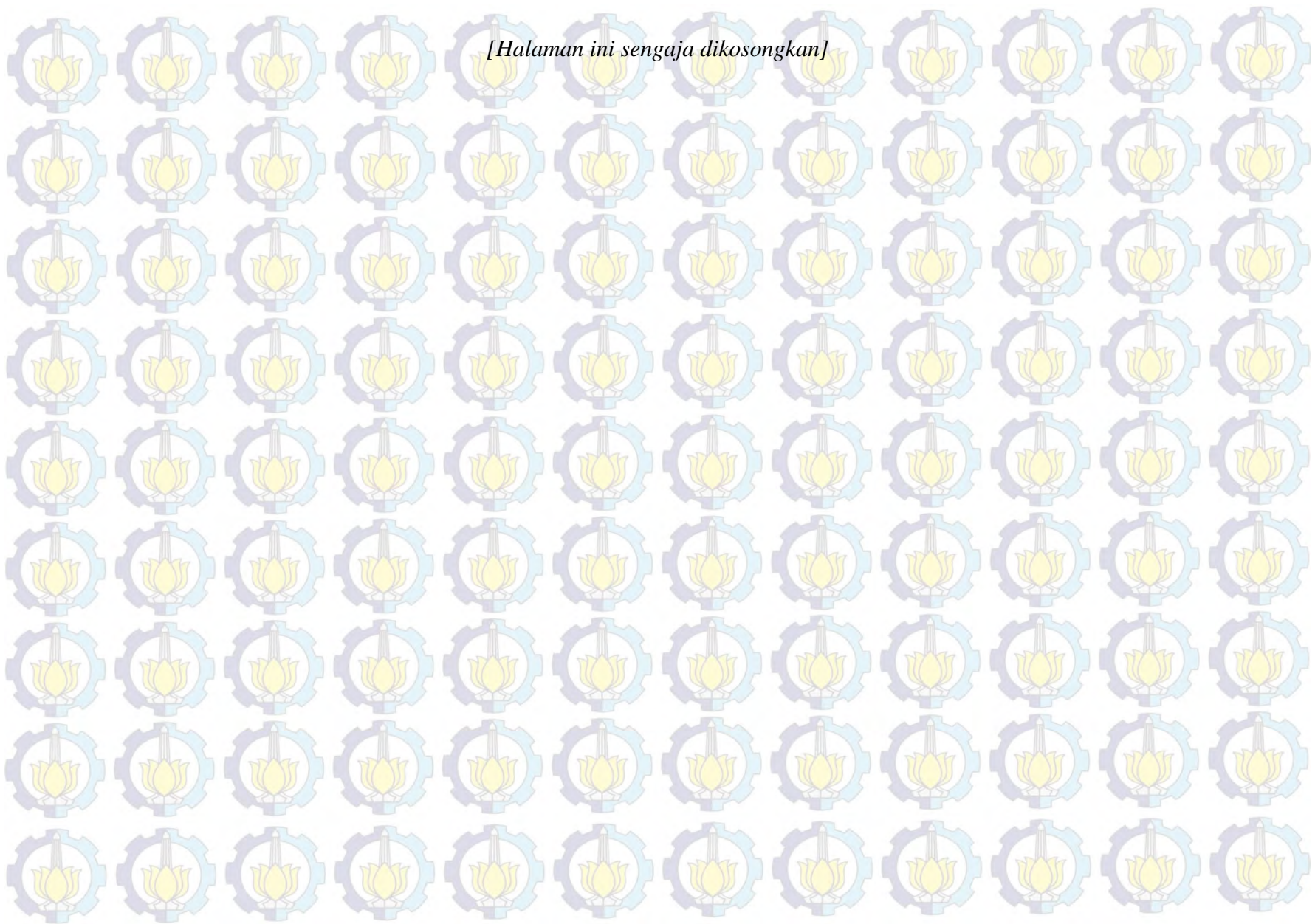
*[Halaman ini sengaja dikosongkan]*



**LAMPIRAN 8**  
**HASIL PENGUJIAN GSA PADA DATASET URINE KELOMPOK JENIS KELAMIN**

Kelompok Jenis Kelamin	Maksimum Iterasi	Jumlah Node Hidden	Rule Pelabelan 1				Rule Pelabelan 2			
			Akurasi	Sensitifitas	Spesifisitas	Waktu Komputasi	Akurasi	Sensitifitas	Spesifisitas	Waktu Komputasi
Pria	50	7	0.54486	0.50850	0.66161	15.1309	0.38483	0.12256	0.89262	12.0178
		11	0.63058	0.38827	0.86541	27.4160	0.45364	0.16337	0.93465	19.3800
		15	0.67066	0.44106	0.89285	26.5651	0.41801	0.15149	0.88129	24.8753
	100	7	0.74690	0.60434	0.86084	26.4887	0.39850	0.21345	0.76818	69.2231
		11	0.64196	0.60135	0.72857	21.9965	0.46537	0.13764	0.90658	22.2950
		15	0.68337	0.48300	0.84814	50.1428	0.39111	0.12487	0.93307	27.0506
		7	0.63882	0.61580	0.65423	16.1152	0.42027	0.14175	0.93248	11.9711
		11	0.67874	0.44861	0.86375	17.3089	0.40762	0.14543	0.84409	24.9663
		15	0.63858	0.52964	0.72777	15.6403	0.42144	0.17403	0.92381	37.4132
Wanita	50	7	0.67931	0.62352	0.76060	7.91320	0.43909	0.22559	0.86941	7.63852
		11	0.63079	0.60000	0.67878	7.27926	0.43594	0.23878	0.82681	7.59784
		15	0.64532	0.65882	0.63182	22.0637	0.44746	0.32650	0.74605	12.0892
	100	7	0.70763	0.77647	0.61060	12.8669	0.39995	0.21900	0.88863	14.8632
		11	0.57635	0.54117	0.63181	10.8321	0.42978	0.29509	0.88363	12.4696
		15	0.67241	0.65882	0.69242	17.9059	0.44935	0.25644	0.90000	30.9994
		7	0.69359	0.68235	0.71515	8.09408	0.49119	0.34892	0.86878	14.0743
		11	0.66625	0.58823	0.78181	8.89534	0.45657	0.31997	0.81846	12.8490
		15	0.73472	0.70588	0.78182	9.75356	0.45946	0.40008	0.70611	21.6274

*[Halaman ini sengaja dikosongkan]*

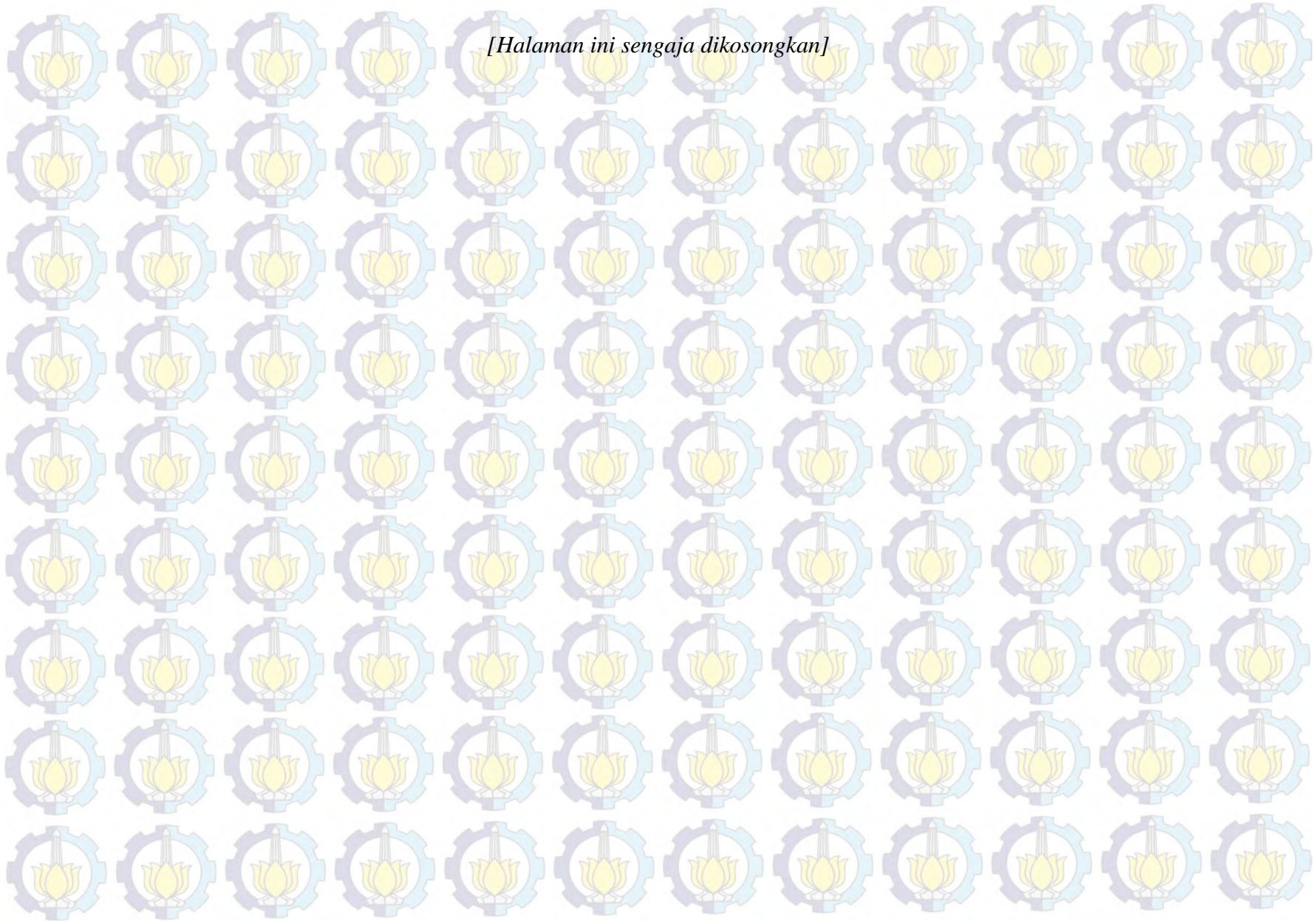




**LAMPIRAN 9**  
**HASIL PENGUJIAN GSAM PADA DATASET URINE**

Metode	Maksimum Iterasi	Jumlah Node Hidden	Rule Pelabelan 1				Rule Pelabelan 2				
			Akurasi	Sensitifitas	Spesifisitas	Waktu Komputasi	Akurasi	Sensitifitas	Spesifisitas	Waktu Komputasi	
<b>3 fold</b>	50	7	0.72979	0.50000	0.95960	10.2555	0.49572	0.24856	0.96029	21.7451	
		11	0.67929	0.45454	0.90404	24.4279	0.51282	0.29271	0.86878	32.9397	
		15	0.68181	0.52525	0.83838	34.6934	0.50085	0.22770	0.98571	43.4296	
	100	7	0.73232	0.56060	0.90404	17.6756	0.48718	0.25065	0.93660	21.3448	
		11	0.74747	0.55050	0.94444	31.4442	0.48034	0.23568	0.92366	47.7043	
		15	0.70202	0.57576	0.82828	30.9840	0.50256	0.26451	0.94112	63.0063	
	200	7	0.71717	0.56060	0.87374	16.0911	0.49059	0.24012	0.94501	22.8827	
		11	0.74242	0.55555	0.92929	19.3891	0.44957	0.45555	0.62226	38.0901	
		15	0.71212	0.59596	0.82828	17.8670	0.52136	0.32098	0.87560	21.0960	
	<b>5 fold</b>	50	7	0.70737	0.65012	0.76461	48.9973	0.48356	0.22406	0.95416	34.5102
			11	0.68429	0.48948	0.87910	31.7823	0.50408	0.28993	0.84638	44.4044
			15	0.69660	0.61974	0.77346	46.3024	0.51304	0.28992	0.86973	60.8478
100		7	0.68474	0.62012	0.74935	26.7088	0.50788	0.26395	0.95500	39.8152	
		11	0.71718	0.50474	0.92961	36.8776	0.46318	0.20994	0.92230	48.1706	
		15	0.71211	0.56551	0.85871	43.1661	0.45851	0.31100	0.77712	78.6136	
200		7	0.74743	0.57589	0.91897	15.0720	0.51656	0.34036	0.81900	29.1330	
		11	0.67936	0.46435	0.89435	21.8967	0.50934	0.30242	0.86612	44.0322	
		15	0.64903	0.64577	0.65230	18.4242	0.48707	0.27212	0.88784	22.4528	
<b>10 fold</b>		50	7	0.71847	0.54359	0.89868	52.3387	0.47019	0.27186	0.82501	23.7542
			11	0.73673	0.56579	0.91368	76.0531	0.45140	0.32476	0.80698	32.4728
			15	0.71536	0.56386	0.85921	103.091	0.49404	0.27607	0.89651	44.7631
	100	7	0.73347	0.54000	0.93894	53.2501	0.47497	0.34990	0.77066	27.2654	
		11	0.73373	0.58219	0.89394	73.9832	0.49222	0.28777	0.87231	44.0230	
		15	0.72644	0.53447	0.92842	99.4302	0.47334	0.23951	0.90454	63.2466	
	200	7	0.73150	0.57052	0.88947	68.2364	0.47191	0.21437	0.92575	60.2121	
		11	0.68821	0.48780	0.88894	62.1883	0.47969	0.37206	0.78579	70.9030	
		15	0.7380	0.59052	0.89447	48.3645	0.51714	0.28822	0.94860	59.9979	

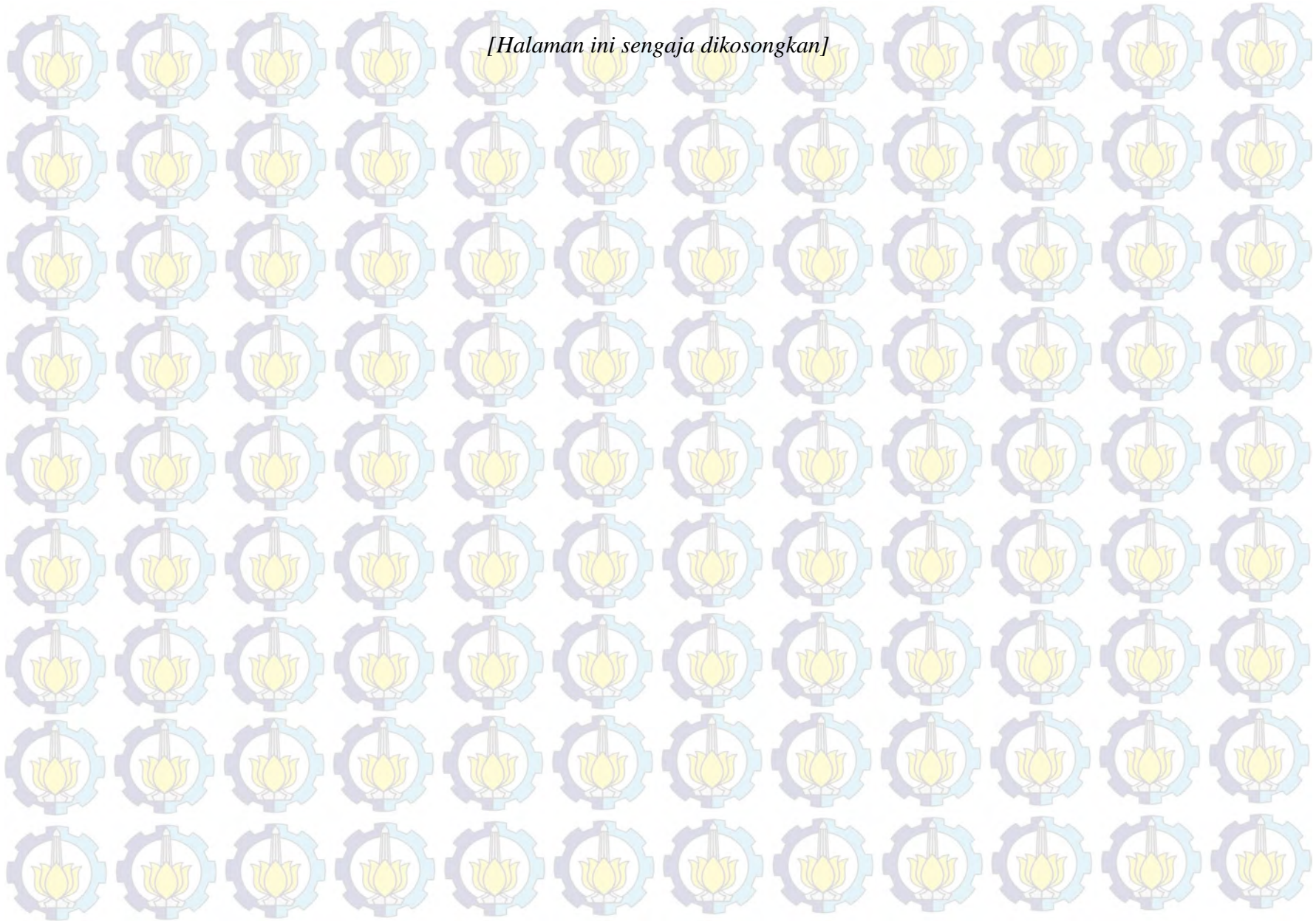
*[Halaman ini sengaja dikosongkan]*



**LAMPIRAN 10**  
**HASIL PENGUJIAN GSAM PADA DATASET URINE KELOMPOK USIA**

Kelompok Usia	Maksimum Iterasi	Jumlah Node Hidden	Rule Pelabelan 1				Rule Pelabelan 2				
			Akurasi	Sensitifitas	Spesifisitas	Waktu Komputasi	Akurasi	Sensitifitas	Spesifisitas	Waktu Komputasi	
< 20 tahun	50	7	0.83333	0.70000	1.0000	2.00674	0.5200	0.46666	0.70000	2.51606	
		11	0.83333	0.70000	1.0000	2.41832	0.5700	0.43333	0.90000	3.26400	
		15	0.83333	0.70000	1.0000	3.26468	0.5200	0.39999	0.80000	3.34006	
	100	7	0.73333	0.60000	0.90000	1.84808	0.5200	0.53333	0.50000	2.88636	
		11	0.83333	0.70000	1.0000	3.70446	0.4700	0.33333	1.0000	3.12146	
		15	0.73333	0.60000	0.90000	3.71584	0.4700	0.49999	0.50000	2.95161	
	200	7	0.73333	0.70000	0.80000	1.46708	0.6200	0.46666	0.90000	2.96276	
		11	0.74000	0.55000	0.97142	1.33748	0.5700	0.66666	0.56666	3.24426	
		15	0.55000	0.50000	0.60000	1.06372	0.5700	0.43333	0.80000	1.80764	
	20 tahun.s.d 50 tahun	50	7	0.79992	0.60769	0.92047	8.61164	0.46834	0.14929	0.95200	13.6525
			11	0.77550	0.71666	0.81142	12.4201	0.49273	0.17934	0.93886	15.6426
			15	0.79386	0.53076	0.96000	14.8032	0.54803	0.22409	0.91815	29.3334
100		7	0.79367	0.62179	0.90095	10.6797	0.46157	0.21497	0.80186	16.1374	
		11	0.79898	0.60384	0.92142	15.4815	0.53128	0.19469	0.96926	33.3188	
		15	0.81791	0.59102	0.96142	20.0066	0.54088	0.23952	0.94808	29.1060	
200		7	0.76445	0.46794	0.95047	7.83456	0.45504	0.14326	0.89720	19.0809	
		11	0.75211	0.68461	0.79571	9.25516	0.49950	0.21006	0.88629	37.2186	
		15	0.74519	0.68461	0.78095	9.14024	0.47229	0.14267	0.94194	24.1991	
> 50 tahun		50	7	0.61030	0.53200	0.72156	10.2187	0.51071	0.40658	0.78472	13.0572
			11	0.62425	0.64800	0.59869	14.9513	0.46396	0.36888	0.79887	20.3609
			15	0.57763	0.57966	0.57646	16.9546	0.50675	0.36042	0.81203	22.4968
	100	7	0.67065	0.63100	0.73202	10.1024	0.48538	0.26616	0.97518	15.5212	
		11	0.65238	0.62033	0.69738	16.9898	0.44870	0.34357	0.69841	23.9097	
		15	0.64318	0.54033	0.78692	20.2370	0.48155	0.29180	0.88514	37.3248	
	200	7	0.66084	0.46346	0.80190	9.66284	0.53954	0.39601	0.85714	16.1272	
		11	0.61157	0.42051	0.79642	21.4113	0.67441	0.25251	0.94491	34.5824	
		15	0.72146	0.44807	0.94000	16.2132	0.51435	0.22078	0.77867	14.31654	

*[Halaman ini sengaja dikosongkan]*



## LAMPIRAN 11

### HASIL PENGUJIAN GSAM PADA DATASET URINE KELOMPOK JENIS KELAMIN

Kelompok Jenis Kelamin	Maksimum Iterasi	Jumlah Node Hidden	Rule Pelabelan 1				Rule Pelabelan 2			
			Akurasi	Sensitifitas	Spesifisitas	Waktu Komputasi	Akurasi	Sensitifitas	Spesifisitas	Waktu Komputasi
Pria	50	7	0.59592	0.52890	0.73694	21.1768	0.40568	0.16362	0.84374	20.8524
		11	0.69804	0.46456	0.92142	43.1350	0.48525	0.27619	0.78364	29.1469
		15	0.71404	0.55099	0.91428	40.9119	0.42239	0.12219	0.98014	44.1197
	100	7	0.80227	0.73596	0.85661	40.9270	0.43479	0.20272	0.82274	96.3942
		11	0.68509	0.53432	0.79857	48.7549	0.39624	0.11854	0.91337	33.1896
		15	0.71466	0.52648	0.86984	84.0070	0.41081	0.29377	0.75665	57.8835
	200	7	0.70274	0.55573	0.82116	23.9148	0.46694	0.16506	0.93744	29.3124
		11	0.72203	0.46679	0.92830	20.1297	0.39946	0.21796	0.74136	68.4825
		15	0.63074	0.49565	0.74206	15.6403	0.42941	0.15308	0.88755	37.4132
Wanita	50	7	0.70739	0.68235	0.74697	17.6526	0.47155	0.27332	0.87878	11.5892
		11	0.66527	0.64706	0.69545	13.4786	0.52046	0.35668	0.83750	14.1168
		15	0.67315	0.75294	0.56666	38.7160	0.50489	0.36130	0.83916	22.8271
	100	7	0.74211	0.71764	0.77727	24.6008	0.49089	0.33370	0.96000	21.0704
		11	0.68029	0.64705	0.73333	14.1257	0.50535	0.37780	0.83907	19.9127
		15	0.72069	0.71764	0.72727	29.0437	0.49706	0.30502	0.93333	43.1350
	200	7	0.74187	0.71764	0.78181	9.30438	0.53898	0.34242	0.98181	22.3435
		11	0.76280	0.65882	0.91515	15.9245	0.50553	0.32918	0.90411	31.1666
		15	0.74286	0.76470	0.70454	9.75356	0.54120	0.38439	0.90571	21.6274

