

**PENGGUNAAN ALGORITMA *SUPERVISED LEARNING*
IN QUEST UNTUK PENGLASIFIKASIAN STATUS GIZI
BALITA PADA KECAMATAN KERTOSONO**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Armeilya Rahmanis Shokikah

NIM: 115060807113029



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

MALANG

2015

PENGESAHAN

PENGGUNAAN ALGORITMA *SUPERVISED LEARNING IN QUEST* UNTUK
PENGKLASIFIKASIAN STATUS GIZI BALITA PADA KECAMATAN KERTOSONO

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Armeilya Rahmanis Shokikah

NIM: 115060807113029

Skripsi ini telah diuji dan dinyatakan lulus pada
3 Desember 2015

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Indriati, ST., M.Kom.
NIK. 831013 06 1 2 0035

Drs. Achmad Ridok, M.Kom
NIP. 19680825 199403 1 004

Mengetahui
Ketua Program Studi Informatika/Illmu Komputer

Drs. Marji, M.T
NIP: 1670801 199203 1 001

PERNYATAAN ORISINALITAS

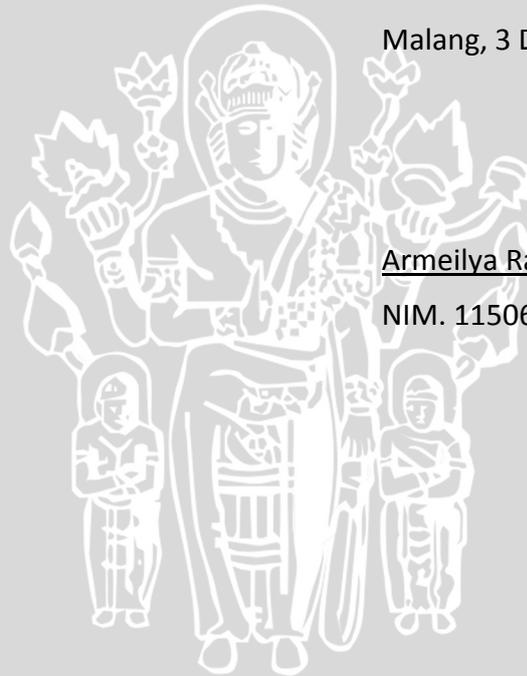
Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan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

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

Malang, 3 Desember 2015

Armeilya Rahmanis Shokikah

NIM. 115060807113029



KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Penggunaan *Algoritma Supervised Learning In Quest* untuk Pengklasifikasian Status Gizi Balita pada Kecamatan Kertosono” ini dapat terselesaikan.

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

1. Ibu Indriati, S.T., M.Kom dan Bapak Drs. Achmad Ridok, M.Kom. selaku dosen pembimbing skripsi satu dan dua yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Ibu Yayuk Fatimah, Ibu Mayulin, dan Bpk. Masrukhi terimakasih atas dukungan dan motivasi untuk segera menyelesaikan skripsi ini.
3. Seluruh Guru Besar saya di Al-Ishlah Kediri, terimakasih atas segala bimbingan dan ilmu yang diberikan. Anda sebagai penyemangat untuk segera menyelesaikan skripsi saya.
4. Teman-teman yang selalu saya repotkan Elliya Lestari, Yasmin Ghasani, Dewi Enggar, Zain Nuril, Lusy Dwi, Ayu Nanda, Ermi Novita, Zhanela Aziza, Julia Ika Ratna, terimakasih banyak atas bantuannya dalam menyelesaikan skripsi saya.

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

Malang, 3 Desember 2015

Penulis

shokikah@gmail.com

ABSTRAK

Balita-balita di Indonesia memiliki status gizi yang berbeda-beda dan kecenderungan untuk memiliki gizi di bawah rata-rata jika tidak ditangani serta dirawat dengan baik oleh pemerintah dan pihak terkait. Penggunaan alat bantu untuk mengetahui balita dengan status gizi pada suatu wilayah sangat diperlukan untuk memudahkan mengklasifikasikan status gizi balita pada kelompok-kelompok status gizi (gizi buruk, gizi kurang, gizi baik, gizi lebih), seperti sebuah aplikasi pengklasifikasian dengan algoritma klasifikasi yang dirasa baik diterapkan untuk sebuah kasus seperti ini. Algoritma klasifikasi yang dipilih untuk diterapkan pada penelitian ini adalah algoritma *Supervised Learning In Quest* (SLIQ) karena SLIQ memiliki tingkat akurasi yang cukup baik diterapkan untuk beberapa penelitian yang telah dilakukan sebelumnya. SLIQ merupakan sebuah metode *decision tree* pada data mining yang dapat mengolah data numerikal dan kategorikal dengan jumlah data yang besar. SLIQ membentuk aturan-aturan dari *tree* untuk mengklasifikasikan sebuah data. Prosentase akurasi yang dihasilkan pada kasus ini adalah 74.59 %. Hasil ini didapatkan dari pengujian tingkat akurasi sistem pada nilai batas dan data pembentuk aturan klasifikasi.

Kata kunci: *status gizi, data mining, klasifikasi, decision tree, SLIQ.*



ABSTRACT

Indonesia has toddlers with different nutritional status and tendency to have nutritional below average if not handle and care by the government and related parties. The use of tools to determine the infant with nutritional status in an area is necessary to facilitate classify the nutritional status of children in groups nutrition (severely underweight, underweight, well nourished, over weight) as an application of the classification with a classification algorithm that good applied to a case such as this. Classification algorithms will chose for this study is Supervised Learning In Quest (SLIQ) algorithms, because SLIQ has a good degree of accuracy that applied to some of the research. SLIQ is a decision tree method in data mining that can process data with numerical and categorical large amount of data. SLIQ establish the rules of the tree to classify data. The percentage accuracy of SLIQ classification in this case is 74.59 %. These results obtained from testing the accuracy of the system at the limit value and data training to rules classification.

Key word : *nutritional status, data mining, classification, decision tree, SLIQ.*



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Balita.....	7
2.3 Gizi dan Status Gizi.....	8
2.4 Penilaian Status Gizi	8
2.5 Klasifikasi Status Gizi Balita	10
2.6 Data Mining.....	11
2.7 Klasifikasi	12
2.8 <i>Supervised Machine Learning</i>	13
2.9 Pohon keputusan (<i>Decision Tree</i>)	14
2.10 Algoritma Supervised Learning in Quest (SLIQ)	15
2.11 Perhitungan Akurasi	18
BAB 3 METODOLOGI DAN PERANCANGAN.....	20

3.1.	Metodologi Penelitian.....	20
3.1.1.	Studi Literatur	21
3.1.2.	Pengumpulan Data.....	21
3.1.3.	Analisa dan Perancangan	21
3.1.4.	Implementasi.....	23
3.1.5.	Pengujian.....	23
3.1.6.	Evaluasi (Kesimpulan dan Saran)	23
3.2.	Perancangan Sistem	23
3.3.	Perancangan Stuktur Tabel pada Basisdata.....	32
3.4.	Perancangan Antarmuka.....	36
3.5.	Perhitungan Manual.....	38
BAB 4 IMLEMENTASI		49
4.1.	Lingkungan Implementasi	49
4.1.1.	Lingkungan <i>Hardware</i>	49
4.1.2.	Lingkungan <i>Software</i>	49
4.2.	Persiapan Data	49
4.3.	Implementasi Antarmuka	49
4.3.1.	Modul Data Latih.....	49
4.3.2.	Modul Data Uji	50
4.3.3.	Modul Klasifikasi	51
4.4.	Implementasi Program.....	51
4.4.1.	Program Proses Algoritma SLIQ	51
4.4.2.	Program perhitungan <i>z-score</i> pada modul klasifikasi	61
4.4.3.	Program <i>Testing</i> Pada Modul Klasifikasi	63
BAB 5 PENGUJIAN DAN PEMBAHASAN.....		66
5.1.	Sisitematika Pengujian	66
5.1.1.	Sisitematika Pengujian Dengan Data Latih dan Uji Sama	66
5.1.2.	Sisitematika Pengujian Pekaruh Perubahan Data Latih dan Data Uji	66
5.1.3.	Sisitematika Pengujian Perubahan Nilai Batas.....	67
5.2.	Hasil Pengujian	69
5.2.1.	Hasil Pengujian Dengan Data Latih dan Data Uji Sama.....	69



5.2.2. Hasil Uji Pengaruh Perubahan Data Latih dan Data Uji 69

5.2.3. Hasil Uji Perubahan Nilai Batas 70

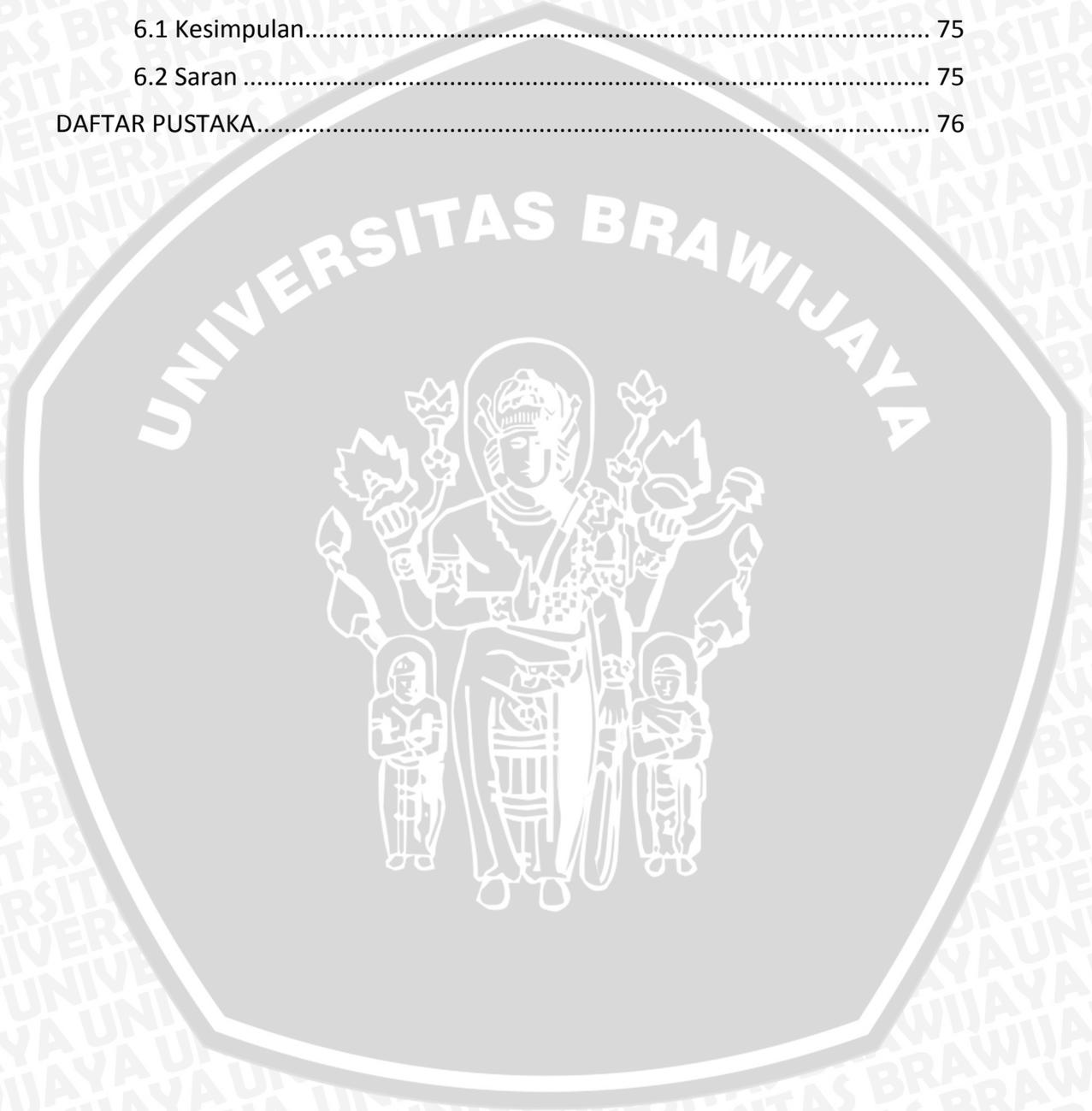
5.3. Analisa Hasil 72

BAB 6 Penutup 75

6.1 Kesimpulan..... 75

6.2 Saran 75

DAFTAR PUSTAKA..... 76



DAFTAR TABEL

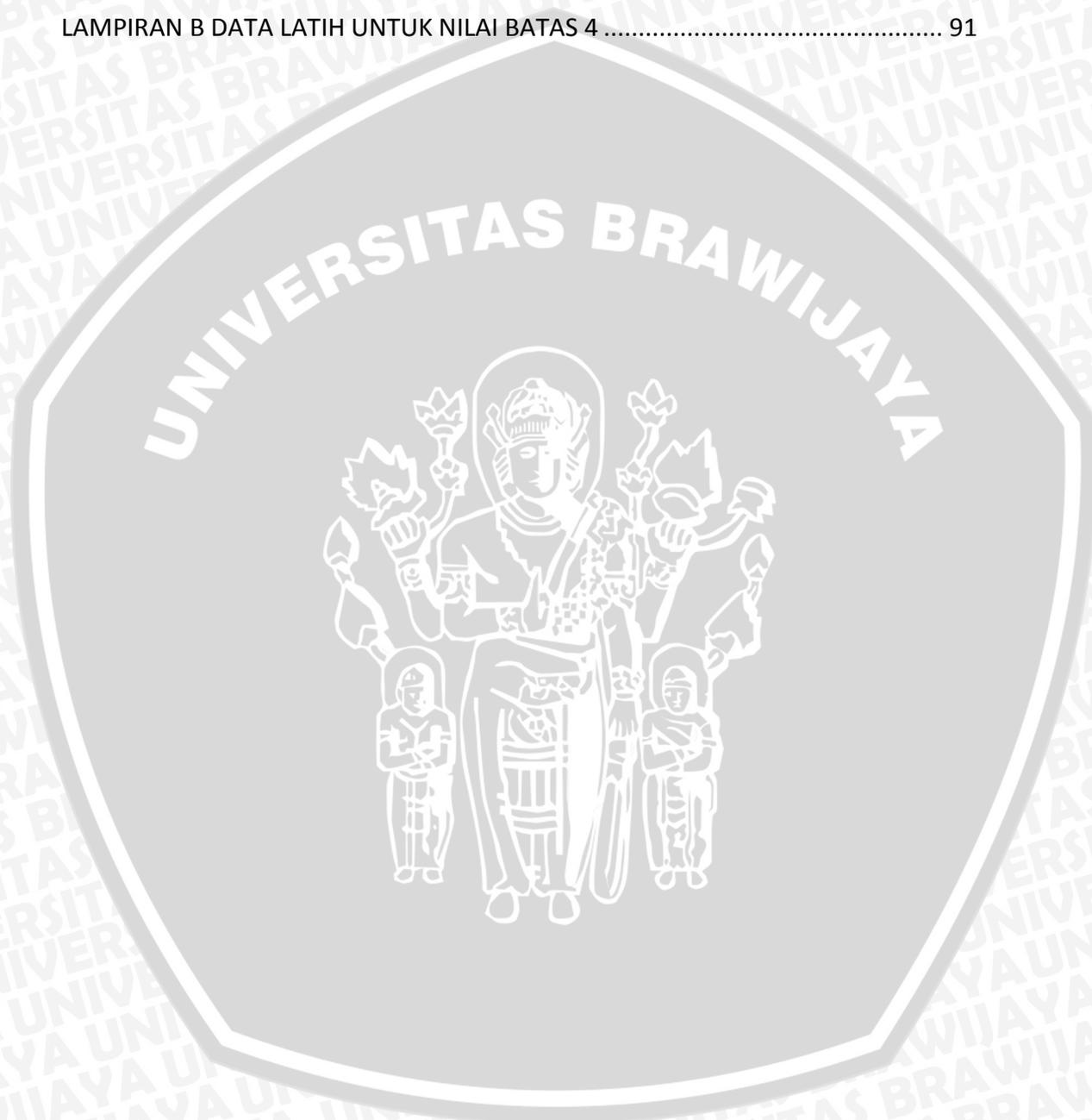
Tabel 2. 1 Kajian pustaka.....	5
Tabel 2. 2 Kategori dan ambang batas status gizi balita.....	10
Tabel 2. 3 Contoh Pre-Sorting.....	16
Tabel 3. 1 Spesifikasi Perangkat Keras.....	22
Tabel 3. 2 Tabel data_balita.....	32
Tabel 3. 3 Tabel data_testing.....	33
Tabel 3. 4 struktur tabel tbperumur di database.....	34
Tabel 3. 5 Tabel bbpertb.....	34
Tabel 3. 6 Tabel bbperumur.....	35
Tabel 3. 7 Tabel sampel data untuk perhitung manual.....	38
Tabel 3. 8 Tabel data balita.....	40
Tabel 3. 9 Tabel Attibut list.....	41
Tabel 3. 10 Tabel Class List.....	42
Tabel 3. 11 Tabel histogram jenis kelamin.....	43
Tabel 3. 12 Tabel histogram umur.....	43
Tabel 3. 13 Tabel histogram berat badan.....	44
Tabel 3. 14 Tabel Histogram tinggi badan.....	44
Tabel 3. 15 Tabel hasil perhitungan gini indeks.....	46
Tabel 3. 16 Tabel subset pada atribut Berat Badan ≤ 13	46
Tabel 3. 17 Tabel subset pada atribut Berat Badan > 13	47
Tabel 5. 1 Tabel nilai batas untuk pengujian.....	67
Tabel 5. 2 Data latihan pembentuk rule.....	68
Tabel 5. 3 Hasil uji dengan data latihan dan uji sama.....	69
Tabel 5. 4 Hasil pengujian akurasi perubahan jumlah data.....	69
Tabel 5. 5 Hasil pengujian data uji sama.....	70
Tabel 5. 6 Hasil pengujian nilai batas 1.....	70
Tabel 5. 7 Hasil pengujian nilai batas 2.....	70
Tabel 5. 8 Hasil pengujian niai batas 3.....	71
Tabel 5. 9 Hasil pengujian nilai batas 4.....	71

DAFTAR GAMBAR

Gambar 2. 1 Proses Supervised Machine Learning.....	13
Gambar 2. 2 <i>Tree Building Algoritm</i>	14
Gambar 2. 3 Model Pohon Keputusan.....	15
Gambar 2. 4 Pseudocode untuk meng-update daftar kelas.....	17
Gambar 3. 1 Alur metodologi penelitian	20
Gambar 3. 2 Gambar Diagram blok sistem pengklasifikasian data bayi menggunakan SLIQ.....	22
Gambar 3. 3 Diagram alir sistem pengklasifikasian status bayi menggunakan SLIQ	24
Gambar 3. 4 Diagram alir sub proses proses data latih	25
Gambar 3. 5 Diagram alir sub proses algoritma SLIQ	26
Gambar 3. 6 Diagram alir sub program histogram	27
Gambar 3. 7 Sub Program menghitung gini indeks	28
Gambar 3. 8 Sub Proses sorting gini indeks dari kecil ke besar	29
Gambar 3. 9 Sub Proses sorting data menurut gini terkecil	30
Gambar 3. 10 Sub Proses Spit	31
Gambar 3. 11 Diagram alir perhitungan Z-Score	32
Gambar 3. 12 Gambar desain tabel data training pada database	33
Gambar 3. 13 Gambar struktur data tabel data_testing	33
Gambar 3. 14 Desain tabel tbperumur pada basisdata	34
Gambar 3. 15 Desain tabel bbpertb pada basisdata.....	35
Gambar 3. 16 Gambar desain tabel bbperumur pada database	36
Gambar 3. 17 Modul data latih	36
Gambar 3. 18 Antarmuka modul data uji	37
Gambar 3. 19 Modul Klasifikasi.....	37
Gambar 3. 20 Hasil <i>Decision Tree</i>	47
Gambar 4. 1 Modul Data Latih.....	50
Gambar 4. 2 Modul Data Uji	50
Gambar 4. 3 Modul Klasifikasi.....	51

DAFTAR LAMPIRAN

LAMPIRAN A DATA BALITA.....	79
LAMPIRAN B DATA LATIH UNTUK NILAI BATAS 4	91



BAB 1 PENDAHULUAN

1.1 Latar belakang

Indonesia masih memiliki angka kematian bayi dan balita yang cukup tinggi. Masalah tersebut terutama dalam periode neonatal dan dampak dari penyakit menular, terutama pneumonia, malaria, dan diare ditambah dengan masalah gizi yang dapat mengakibatkan lebih dari 80% kematian anak (WHO,2002). Balita adalah anak bawah lima tahun (balita) merupakan kelompok yang menunjukkan pertumbuhan badan pesat, sehingga memerlukan zat-zat gizi yang tinggi setiap kilogram berat badannya. Masa balita adalah masa pertumbuhan sehingga memerlukan gizi yang baik, namun anak balita ini justru merupakan kelompok umur yang paling sering menderita akibat kekurangan gizi (Paath, dkk., 2010). Gizi kurang dan gizi buruk adalah status gizi yang didasarkan pada indeks berat badan menurut umur (BB/U) yang merupakan padanan istilah *underweight* (gizi kurang) dan *severely underweight* (gizi buruk) (Keputusan Menteri Kesehatan RI nomor : 1995/MENKES/SK/XII/2010). Untuk menekan gizi buruk pada suatu daerah tentu saja Dinas Kesehatan setempat harus ikut bertanggung jawab untuk mengatasi masalah gizi buruk melalui kebijakan dan program kesehatan. Berdasarkan hal ini, mengetahui status gizi pada balita adalah hal yang sangat penting untuk dipahami oleh para orangtua mengingat usia balita merupakan masa yang tergolong rawan dalam pertumbuhan dan perkembangan karena pada masa ini balita mudah terserang penyakit dan mudah kekurangan gizi. Kejadian gizi buruk apabila tidak diatasi akan menyebabkan dampak yang buruk bagi balita. Dampak yang terjadi antara lain kematian dan infeksi kronis (Novitasari, 2012).

Mengingat tingkat kelahiran di Indonesia yang tinggi, tentunya banyak sekali balita yang membutuhkan perhatian khusus terhadap tumbuh kembangnya. Kesehatannya juga perlu diperhatikan baik oleh ibunya ataupun peran serta pemerintah. Cara yang dapat dilakukan pertama adalah mengelompokkan balita-balita dengan gizi buruk, kemudian melakukan cara selanjutnya untuk menanggulangi gizi buruk tersebut.

Salah satu cara untuk melakukan klasifikasi atau pengelompokan adalah dengan menggunakan *decision tree*. Cara *decision tree* ini relatif lebih cepat dibandingkan menggunakan cara lainnya (Sutjiadi, 2009). Pembuatan sistem untuk menggolongkan balita ini sangat diperlukan untuk membantu dinas terkait untuk lebih memperhatikan kesehatan balita tersebut. Metode yang digunakan untuk mengklasifikasi adalah algoritma *Supervised learning in Quest*. Algoritma SLIQ menggunakan modifikasi dari *tree classifier* sehingga bisa dipakai juga untuk *dataset* yang besar. SLIQ bisa dipakai untuk atribut dengan tipe numerik dan kategorikal. *Decision tree* dalam SLIQ menggunakan teknik novel (teknik terbaru) untuk mempersingkat *learning time* dengan tetap mempertahankan tingkat akurasi yang tinggi (Sutjiadi, 2009). Oleh karena itu jika data atribut yang digunakan berupa numerik atau kategorikal, maka algoritma ini cocok untuk digunakan.

Oleh sebab itu penulis tertarik untuk melakukan penelitian pada bidang ini dengan menerapkan SLIQ sebagai metode penyelesaian masalah dengan

mengangkat judul “ *Penggunaan Algoritma Supervised Learning In Quest untuk Pengklasifikasian Status Gizi Balita pada Kecamatan Kertosono*” sebagai topik tugas akhir penulis.

Di dalam penentuan gizi bayi dinas kesehatan telah memiliki standar dan perhitungan untuk menentukan status gizi pada balita. Perhitungan tersebut menggunakan indeks Antropometri, penulis sendiri juga ingin membandingkan bagaimana menggolongkan status gizi balita dengan data-data yang telah ada dan menerapkan teori dari bidang kesehatan. Nantinya apakah metode ini juga baik dan cocok diterapkan untuk mengklasifikasikan data bayi akan diteliti lebih lanjut.

Data yang penulis peroleh dari jurnal SLIQ : *A Fast Scalable Classifier for Data Mining* oleh Manish Mehta, Rakesh Agrawal dan Jorma Rissanen, jurnal IBM *Almaden Research Center* menunjukkan performa antara algoritma SLIQ, IND-Cart dan IND-C4. Pada jurnal tersebut menunjukkan bahwa algoritma SLIQ lebih unggul 5.3 % untuk *Dataset* Diabetes dibanding dengan IND-Cart dan IND-C4. Dengan tingkat akurasi mencapai 75.4 % untuk dataset tersebut. Sedangkan untuk *dataset* Shuttle mencapai 99.9% (Mehta, Agrawal, dan Rissanen, 1996). Penelitian yang lain juga telah dilakukan oleh Arif Jananto dengan judul penelitiannya *Penggunaan Algoritma SLIQ untuk Pengklasifikasian Kinerja Akademik Mahasiswa* menunjukkan prediksi tingkat akurasinya sebesar 41,52% (Jananto,2010).

1.2 Rumusan masalah

Permasalahan yang ada pada penelitian ini dapat dirumuskan sebagai berikut :

1. Bagaimana penerapan Algoritma *Supervised Learning in Quest* (SLIQ) untuk penentuan status gizi pada balita di Kecamatan Kertosono ?
2. Bagaimana hasil akurasi dari penerapan Algoritma *Supervised Learning in Quest* (SLIQ) untuk penentuan status gizi pada balita di Kecamatan Kertosono ?

1.3 Tujuan

Tujuan penelitian tugas akhir ini adalah menentukan status gizi balita pada Kecamatan Kertosono dengan menerapkan Algoritma *Supervised Learning In Quest* dan mengetahui tingkat akurasi sistem jika algoritma ini diterapkan untuk kasus pengklasifikasian status gizi balita serta untuk mengetahui penerapan algoritma SLIQ pada sistem.

1.4 Manfaat

Manfaat yang dapat dihasilkan dari penelitian ini adalah :

1. Bagi Penulis
 1. memenuhi syarat kelulusan sebagai Sarjana Komputer dan untuk penerapan ilmu yang diperoleh selama perkuliahan.
 2. Mendapatkan pemahaman mengenai Algoritma *Supervised Learning in Quest*.

2. Bagi Pengguna.

1. Memudahkan pengguna untuk mengklasifikasikan dan menentukan seorang balita tersebut termasuk ke dalam balita gizi buruk, gizi normal, atau gizi lebih.

Penelitian ini diharapkan dapat memberikan manfaat berupa sumbangan ilmu pengetahuan bagi universitas khususnya Fakultas FILKOM.

1.5 Batasan masalah

Penelitian ini dibatasi oleh hal-hal berikut:

1. Data yang digunakan adalah data gizi balita dari Puskesmas Kecamatan Kertosono.
2. Parameter yang akan dioptimasi adalah beberapa parameter *inputan* berupa Berat Badan (BB), Tinggi Badan (TB), Umur (U), Jenis Kelamin (JK).
3. Hasil status gizi ada 4 kategori yaitu gizi buruk, gizi kurang, gizi baik, dan gizi lebih.
4. Inputan (data latih dan data uji) yang diterima berupa tabel-tabel yang telah tersimpan di dalam basisdata
5. Parameter output yang dihasilkan dari sistem adalah hasil klasifikasi yang ditentukan oleh algoritma SLIQ, serta nilai z-score dari indeks antropometri. Nilai z-score berupa BB/U, TB/U, BB/TB.
6. Menerapkan *tree building*.
7. Tidak menangani tahap *pre-processing (data cleaning)*.
8. Bahasa pemrograman yang digunakan adalah Java, dengan *compiler* Netbean 8.0.2, DBMS menggunakan phpmyadmin dan Xampp.

1.6 Sistematika pembahasan

Sistematika penyusunan laporan ini secara garis besar meliputi beberapa bab, yaitu sebagai berikut :

BAB 1 PENDAHULUAN

Membahas latar belakang masalah, tujuan penelitian, rumusan masalah, batasan masalah dan sistematika penulisan yang akan digunakan dalam penelitian ini.

BAB 2 LANDASAN KEPUSTAKAAN

Membahas kajian pustaka dan landasan teori yang berhubungan dengan penelitian ini.

BAB 3 METODOLOGI DAN PERANCANGAN

Membahas langkah-langkah serta perancangan sistem yang akan dilakukan dalam penelitian ini.

BAB 4 IMPLEMENTASI

Membahas tentang penerapan Algoritma *Supervised Learning in Quest (SLIQ)* untuk penentuan status gizi balita pada Kecamatan Kertosono pada sistem.

BAB 5 PENGUJIAN DAN PEMBAHASAN

Membahas tentang hasil pengujian dari implementasi sistem

BAB 6 PENUTUP

Bagian ini berisi kesimpulan yang diperoleh dari penelitian dan saran untuk penelitian selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepustakaan berisi uraian dan pembahasan tentang teori, konsep, dan metode dari literatur ilmiah, yang berkaitan dengan klasifikasi status gizi pada balita menggunakan algoritma SLIQ. Dalam landasan kepustakaan terdapat landasan teori dari berbagai sumber pustaka yang terkait dengan teori dan metode yang digunakan dalam penelitian ini.

2.1 Kajian Pustaka

Sebelum ini telah ada kajian pustaka atau penelitian yang dilakukan terkait penggunaan Algoritma *Supervised Learning In Quest* (SLIQ). Dasar atau acuan untuk melakukan sebuah penelitian berupa teori-teori ataupun penemuannya merupakan hal yang sangat perlu untuk dijadikan sebagai data pendukung. Di bawah ini merupakan daftar beberapa penelitian yang telah dilakukan terkait dengan penelitian menggunakan Algoritma SLIQ.

Tabel 2. 1 Kajian pustaka

No.	Judul	Peneliti	Metode terkait	Hasil Penelitian	Tingkat Akurasi (penggunaan SLIQ)	Tahun
1	<i>SLIQ : A Fast Scalable Classifier for Data Mining</i>	Manish Mehta, Rakesh Agrawal dan Jorma Rissanen	SLIQ, IND-Cart, IND-C4	Performa dan tingkat akurasi yang dihasilkan antara algoritma SLIQ, IND-Cart, dan IND-C4 terhadap beberapa <i>Dataset</i> yang diujikan. Menjelaskan tentang bagaimana penggunaan	Nilai rata-rata akurasi SLIQ dari 8 <i>Dataset</i> yang diujikan : 86.01 %	2000

				<p>algoritma SLIQ.</p> <p>Tingkat akurasi yang dihasilkan SLIQ yang cukup tinggi.</p>		
2	<p>Penerapan Algoritma <i>SLIQ</i> untuk Pengklasifika sian Jurusan pada SMK bagi Siswa Baru (Studi Kasus : Data pendaftaran pada PSB online jenjang SMK Provinsi DKI Jakarta tahun 2008-2010)</p>	<p>Hendrawan Kuncoro</p>	<p>SLIQ</p>	<p>Algoritma <i>SLIQ</i> yang telah diimplementasik an untuk mengklasifikasika n siswa SMK yang diterima dalam jurusan dalam bentuk aplikasi desktop berbasis Java. Nilai <i>precision</i>, <i>recall</i> dan <i>f-measure</i> yang cukup baik sebesar 75 %</p>	<p>Rata-rata nilai akurasi data siswa tahun 2008-2010 adalah 93%. Sedangkan untuk data siswa tahun 2011 sebesar 95%</p>	<p>2012</p>
3	<p>Penggunaan Algoritma <i>SLIQ</i> untuk Pengklasifika sian Kinerja Akademik Mahasiswa</p>	<p>Arief Jananto</p>	<p>SLIQ</p>	<p>Algoritma <i>Supervised Learning In Quest</i> dapat digunakan untuk memprediksi kinerja (evaluasi belajar)</p>	<p>Nilai rata-rata tingkat akurasi yang dihasilkan dari 2 Dataset yang digunakan antara data</p>	<p>2010</p>

				mahasiswa baru. Tingkat akurasi yang cukup baik tanpa menggunakan perhitungan yang rumit.	mahasiswa tahun 2005 dan 2006 adalah : 42.52 %	
4	Perbandingan Performasi Algoritma <i>Nearest Neighbor</i> dan SLIQ untuk Prediksi Kinerja Akademik Mahasiswa Baru	Arief Jananto	SLIQ, <i>Nearest Neighbor</i>	Menghaislkan sebuah aplikasi data mining dengan menggunakan algoritma <i>Nearest Neighbor</i> dan SLIQ yang dapat digunakan untuk memprediksi kinerja akademik mahasiswa baru.	Nilai rata-rata dari data latih dan testing yang diuji antara SLIQ dan NN adalah untuk SLIQ : 52.39 % sedangkan untuk <i>Nearest Neighbor</i> : 82.5 %	2010

Dari beberapa kajian pustaka yang telah didapat pada Tabel 2.1 dapat disimpulkan bahwa algoritma SLIQ mempunyai tingkat akurasi yang cukup baik untuk *dataset* dengan jumlah yang besar serta dataset yang memiliki atribut kategorikal maupun numerikal. Oleh karena beberapa penelitian yang telah dilakukan seperti tertera pada Tabel 2.1 tersebut, dapat disimpulkan bahwa algoritma ini dapat diterapkan pada kasus penentuan status gizi balita. Dimana status gizi balita ini mempunyai nilai atribut kategorikal dan numerikal.

2.2 Balita

Balita adalah anak bawah lima tahun (balita) merupakan kelompok yang menunjukkan pertumbuhan badan pesat, sehingga memerlukan zat-zat gizi yang tinggi setiap kilogram berat badannya. Masa balita adalah masa pertumbuhan sehingga memerlukan gizi yang baik, namun anak balita ini justru merupakan

kelompok umur yang paling sering menderita akibat kekurangan gizi (Paath, dkk., 2010).

Kondisi kecukupan gizi sangatlah berpengaruh dengan kondisi kesehatannya secara berkesinambungan pada masa mendatang (Muaris, 2006).

2.3 Gizi dan Status Gizi

Nutrisi atau gizi adalah substansi organik yang dibutuhkan organisme untuk fungsi normal dari sistem tubuh, pertumbuhan, pemeliharaan kesehatan (Wikipedia, 2015).

Penelitian di bidang nutrisi mempelajari hubungan antara makanan dan minuman terhadap kesehatan dan penyakit, khususnya dalam menentukan diet yang optimal. Pada masa lalu, penelitian mengenai nutrisi hanya terbatas pada pencegahan penyakit kurang gizi dan menentukan standar kebutuhan dasar nutrisi pada makhluk hidup. Angka kebutuhan nutrisi (zat gizi) dasar ini dikenal di dunia internasional dengan istilah *Recommended Daily Allowance* (RDA) (Wikipedia, 2015).

Status gizi merupakan keadaan tubuh sebagai akibat konsumsi makanan dan penggunaan zat-zat gizi. Dibedakan menjadi status gizi buruk, status gizi kurang, status gizi baik dan status gizi lebih. Status gizi baik atau status gizi optimal terjadi bila tubuh memperoleh cukup zat-zat gizi yang digunakan secara efisien, sehingga memungkinkan pertumbuhan fisik, perkembangan otak, kemampuan kerja dan kesehatan secara umum pada tingkat setinggi mungkin. Status gizi kurang terjadi bila tubuh mengalami kekurangan satu atau lebih zat-zat gizi esensial. Status gizi lebih terjadi bila tubuh memperoleh zat-zat gizi dalam jumlah berlebihan sehingga menimbulkan efek toksin atau membahayakan (Almatsier, 2005). Status gizi buruk terjadi bila kondisi gizi kurang berlangsung lama, maka akan berakibat semakin berat tingkat kekurangannya. Pada keadaan ini dapat menimbulkan berbagai penyakit pada balita. Menurut Notoatmodjo (2003) yang dikutip dari karya ilmiah Aaltje E. Manampiring (2008) kelompok bayi 0-23 bulan dan balita 1-5 tahun termasuk ke dalam kelompok rentan gizi yang merupakan kelompok di dalam masyarakat yang paling mudah menderita gangguan kesehatan atau rentan karena kekurangan gizi (Manampiring, 2008).

Status gizi merupakan salah satu determinan utama status kesehatan penduduk. Salah satu indikator status gizi penduduk yang rendah adalah tingginya prevalensi gizi kurang dan gizi buruk pada anak bawah lima tahun (balita) yang didasarkan pada Berat Badan menurut Umur (BB/U) (Depkes RI 1999 tentang Status Gizi dan Imunisasi Ibu dan Anak di Indonesia).

2.4 Penilaian Status Gizi

Menurut Baliwati dkk dalam dr. Aaltje Manampiring, M. Kes. status gizi dapat diketahui melalui penilaian konsumsi makanan berdasarkan data kuantitatif maupun kualitatif. Cara lain yang sering digunakan untuk mengetahui status gizi

yaitu dengan cara biokimia, antropometri ataupun secara klinis (Manampiring, 2008).

Status gizi pada balita dan anak dapat diukur dengan menggunakan indeks antropometri. Antropometri adalah pengukuran dari dimensi fisik tubuh manusia. Antropometri adalah teknik yang sangat berguna untuk mengestimasi komposisi tubuh sehingga membutuhkan ketelitian dalam pengukuran serta keahlian dan alat-alat yang sudah distandarisasi, hal ini dikutip dari Mitchell dalam dr. Aaltje Manampiring, M. Kes. (Manampiring, 2008).

Menurut Baliwati dkk dalam dr. Aaltje Manampiring, M. Kes. adapun indeks antropometri tersebut antara lain (Manampiring, 2008) :

- a. Indeks berat badan menurut umur (BB/U)
- b. Indeks berat badan menurut panjang atau tinggi badan (BB/PB atau BB/TB)
- c. Indeks panjang atau tinggi badan menurut umur (PB/U atau TB/U)
- d. Indeks gabungan (BB/U ; BB/TB ; TB/U)
- e. Indeks lingkar lengan atas (LILA)
- f. Indeks lingkar kepala menurut umur (LK/U)
- g. Tebal lipatan lemak bawah kulit (TLBK)

2.4.1 Model Pernyataan Indeks Antropometri

Penelitian baik untuk individu atau populasi, indeks antropometri dapat dibandingkan antar populasi menggunakan *percentiles* dan *Z-score* yang berasal dari data yang ada. Sebagian besar negara industri menggunakan perhitungan Indeks Antropometri secara *percentiles*, namun untuk negara berkembang lebih menyukai menggunakan *Z-score* untuk menghitung indeks Antropometri. Dalam beberapa keadaan, *percentiles* dan *Z-score* tidak dapat dihitung dan sebagai gantinya indeks antropometri dihitung menggunakan istilah yang disebut *percent-of-median* (simpang baku). Namun, *percent-of-median* (simpang baku) hanya memberikan informasi yang terbatas pada nilai yang berada pada posisi relatif dari data populasi (Gibson, 2005). Pada aplikasi yang akan menerapkan metode SLIQ ini nantinya perhitungan status gizi per anak juga akan menerapkan perhitungan dengan nilai *Z-score* nya. Dimana nilai *Z-score* ini bertujuan untuk memperoleh nilai dari BB/U, TB/U dan BB/TB. Perhitungan *Z-score* ditunjukkan pada persamaan (2.1) (Gibson, 2005) :

$$Z - score = \frac{wt\ of\ subject - median\ reference\ value\ of\ wt - for - ht}{1\ SD\ below\ median\ reference\ value\ of\ wt - for - ht} \quad (2.1)$$

Sedangkan *percent-of-median* adalah rasio yang dipertimbangkan pada nilai antropometri (contoh : berat badan). Pada individu, nilai tengah dari data yang ada untuk tinggi badan dan umur, dinyatakan dalam nilai presentase. Misalnya, 60% dari median *weight-of-age* (BB/U) merepresentasikan keadaan gizi buruk pada balita lebih parah dari pada anak-anak di atasnya (Gibson, 2005).

2.5 Klasifikasi Status Gizi Balita

Dalam SK Menkes RI No. 920 tahun 2002 ditetapkan klasifikasi status gizi anak bawah lima tahun (balita) yang didasarkan pada baku rujukan WHO NCHS (Manampiring, 2008). Pada Tabel 2.2 menjelaskan tentang kategori dan ambang batas status gizi anak yang menjadi standar Kementerian Kesehatan RI.

Tabel 2. 2 Kategori dan ambang batas status gizi balita

Indeks	Kategori Status Gizi	Ambang Batas (Z-Score)
Berat Badan menurut Umur (BB/U) Anak Umur 0-60 Bulan	Gizi Buruk	<-3 SD
	Gizi Kurang	-3 SD sampai dengan -2 SD
	Gizi Baik	-2 SD sampai dengan 2 SD
	Gizi Lebih	>2 SD
Panjang Badan menurut Umur (PB/U) atau Tinggi Badan menurut Umur (TB/U) Anak Umur 0-60 Bulan	Sangat Pendek	<-3 SD
	Pendek	-3 SD sampai dengan -2 SD
	Normal	-2 SD sampai dengan 2 SD
	Tinggi	>2 SD
Berat Badan menurut Panjang Badan (BB/PB) atau Berat Badan menurut Tinggi Badan Anak Umur 0-60 Bulan	Sangat Kurus	<-3 SD
	Kurus	-3 SD sampai dengan -2 SD
	Normal	-2 SD sampai dengan 2 SD
	Gemuk	>2 SD

Sumber: Keputusan MENKES RI No. 920 Tahun 2002

Adapun penjelasan menurut Harvard dalam Supriasa 2002 klasifikasi status gizi dapat dibedakan menjadi 4 yaitu (Supriasa, 2002) :

a. Gizi lebih (*Over weight*)

Gizi lebih terjadi bila tubuh memperoleh zat-zat gizi dalam jumlah berlebihan sehingga menimbulkan efek toksis atau membahayakan (Almatsier, 2005). Kelebihan berat badan pada balita terjadi karena ketidakmampuan antara energi yang masuk dengan keluar, terlalu banyak makan, terlalu sedikit olahraga atau keduanya. Kelebihan berat badan anak tidak boleh diturunkan, karena penyusutan berat akan sekaligus menghilangkan zat gizi yang diperlukan untuk pertumbuhan (Arisman, 2002).

b. Gizi baik (*well nourished*)

Status gizi baik atau status gizi optimal terjadi bila tubuh memperoleh cukup zat-zat gizi yang digunakan secara efisien sehingga memungkinkan

pertumbuhan fisik, perkembangan otak, kemampuan kerja dan kesehatan secara umum pada tingkat setinggi mungkin (Almatsier, 2005).

c. Gizi kurang (*under weight*)

Status gizi kurang terjadi bila tubuh mengalami kekurangan satu atau lebih zat-zat esensial (Almatsier, 2005).

d. Gizi buruk (*severe PCM*)

Gizi buruk adalah suatu kondisi di mana seseorang dinyatakan kekurangan nutrisi, atau dengan ungkapan lain status nutrisinya berada di bawah standar rata-rata. Nutrisi yang dimaksud bisa berupa protein, karbohidrat dan kalori. Di Indonesia, kasus KEP (Kurang Energi Protein) adalah salah satu masalah gizi utama yang banyak dijumpai pada balita. Gizi buruk adalah bentuk terparah dari proses terjadinya kekurangan gizi menahun (Nency, 2005).

2.6 Data Mining

Data mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam *database* (Kusrini, Taufiq, 2009). Sedangkan menurut Turban data mining adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengindektifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai *database* besar (Kusrini, Taufiq, 2009). Hal penting yang terkait dengan data mining adalah :

1. Data mining merupakan suatu proses otomatis terhadap data yang sudah ada.
2. Data yang akan diproses berupa data yang sangat besar.
3. Tujuan data mining adalah mendapatkan hubungan atau pola yang mungkin memberikan indikasi yang bermanfaat.

2.6.1 Pengelompokan Data Mining

Menurut Larose data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan (Kusrini, Taufiq, 2009) :

a. Deskripsi

Terkadang peneliti dan analis secara sederhana ingin mencoba melakukan percobaan mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Sebagai contoh, petugas pengumpulan suara mungkin tidak dapat menemukan keterangan atau fakta bahwa siapa yang tidak cukup profesional akan sedikit didukung dalam pemilihan presiden. Deskripsi pola dan kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

b. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih ke arah numeric daripada ke arah kategori. Model dibangun menggunakan *record* lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan variabel prediksi.

c. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada dimasa mendatang. Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi.

d. Klasifikasi

Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan menjadi tiga kategori yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah.

e. Pengklusteran

Pengklusteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. Kluster adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lain dan memiliki ketidakmiripan dengan *record-record* dalam kluster lain. Pengklusteran berbeda dengan klasifikasi yaitu tidak adanya variabel target dalam pengklusteran. Pengklusteran tidak mencoba untuk melakukan klasifikasi, mengestimasi atau memprediksi nilai variabel target. Akan tetapi, algoritma pengklusteran mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan (homogen), yang mana kemiripan *record* dalam suatu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal.

f. Asosiasi

Tugas asosiasi dalam *data mining* adalah menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja.

2.7 Klasifikasi

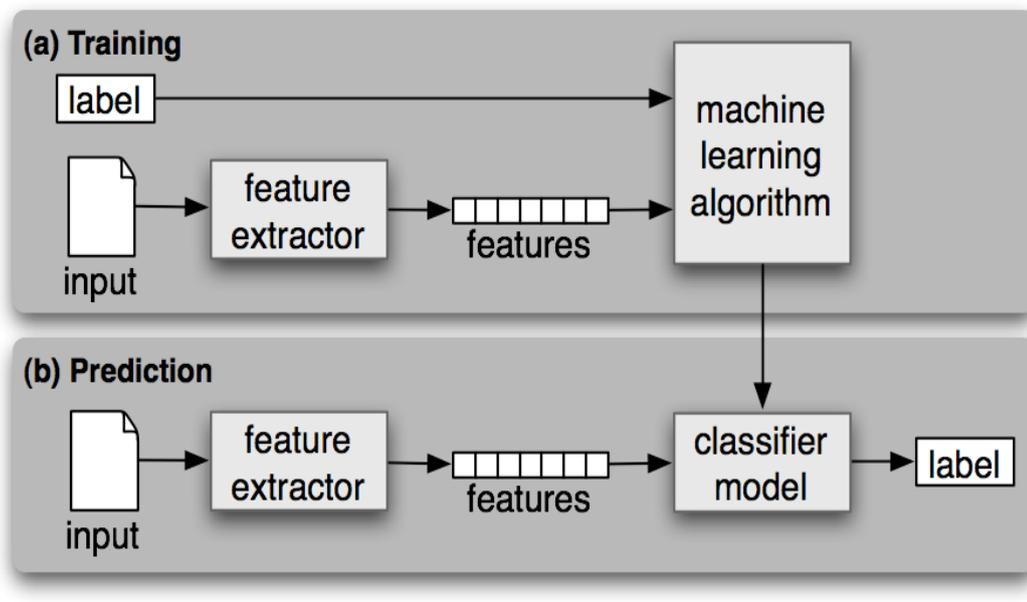
Klasifikasi adalah proses untuk menemukan sebuah model (atau fungsi) yang menggambarkan dan membedakan kelas data atau konsep, dengan tujuan agar mampu menggunakan model untuk memprediksi kelas objek pada label kelas yang tidak diketahui. Model didasarkan pada analisis dari satu set data pelatihan (yaitu, objek data yang label kelasnya diketahui) (Han, Kamber, 2006).

"Bagaimana sebuah model dapat disajikan?" Model ini berasal dapat diwakili dalam berbagai bentuk, seperti klasifikasi (IF-THEN) *rule*, *decision tree*,

rumus matematika, atau jaringan saraf. Sebuah pohon keputusan merupakan sebuah struktur pohon aliran – seperti grafik, di mana setiap node menunjukkan tes pada nilai atribut, setiap cabang merupakan hasil dari pengujian, dan *leaf* mewakili kelas atau distribusi kelas. Pohon keputusan dengan mudah dapat dikonversi ke aturan klasifikasi (Han, Kamber, 2006).

2.8 Supervised Machine Learning

Supervised Machine Learning merupakan pencarian algoritma yang berdasarkan contoh yang diberikan dari luar dan menghasilkan hipotesa umum, yang kemudian dapat menjadi prediksi untuk data berikutnya. Dimana contoh atau data latih yang diberikan dalam *Supervised Machine Learning* memiliki *class* yang diketahui, berkebalikan dengan *Unsupervised Machine Learning*, dimana contoh yang diberikan tidak memiliki *class* (Kotsiantis, 2007).



Gambar 2. 1 Proses Supervised Machine Learning.

Sumber : (Bird, Klein & Loper, 2009)

Pada Gambar 2.1 menggambarkan tentang (a) Selama pelatihan, ekstraktor fitur digunakan untuk mengubah setiap nilai masukan untuk satu set fitur. Set fitur ini, yang menangkap informasi dasar tentang setiap masukan yang harus digunakan untuk mengklasifikasikan. Set fitur dan label dimasukkan ke dalam algoritma *machine learning* untuk menghasilkan model. Sedangkan gambar (b) Selama prediksi (pengujian), fitur ekstraktor yang sama digunakan untuk mengkonversi input yang tak terlihat untuk fitur set. Fitur set ini kemudian dimasukkan ke dalam model, yang menghasilkan label diprediksi (Bird, Klein & Loper, 2009).

2.9 Pohon keputusan (*Decision Tree*)

Pohon keputusan merupakan metode klasifikasi dan prediksi yang sangat kuat dan terkenal. Metode pohon keputusan mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan aturan. Aturan dapat dengan mudah dipahami dengan Bahasa alami. Dan mereka juga dapat diekspresikan dalam bentuk bahasa basisdata seperti *structured Query Language* untuk mencari *record* pada kategori tertentu (Kusrini, Taufiq, 2009).

Menurut Berry dan Linoff, sebuah pohon keputusan adalah sebuah struktur yang dapat digunakan untuk membagi kumpulan data yang besar menjadi himpunan-himpunan *record* yang lebih kecil dengan menerapkan serangkaian aturan keputusan. Dengan masing-masing rangkaian pembagian, anggota himpunan hasil menjadi mirip satu dengan yang lain (Kusrini, Taufiq, 2009).

2.9.1 Klasifikasi Pohon Keputusan

Kebanyakan penggolongan *decision-tree* (seperti CART, DC.5) dilakukan dalam dua tahapan, yaitu : *Tree Building* dan *Tree Prunning*. Namun untuk implementasi pada kasus klasifikasi menggunakan SLIQ ini, hanya menggunakan *Tree Building*

Tree Building : Awalnya pohon keputusan dibentuk pada fase ini dengan pengulangan perhitungan pada data latih yang telah di bagi. Data latih dibagi menjadi dua bagian atau lebih menggunakan atribut yang telah ditentukan. Pada proses ini dilakukan pengulangan rekursif hingga semua contoh data dalam setiap partisi masuk ke dalam satu kelas (Mehta, Agrawal, dan Rissanen, 1996). Setiap simpul dalam menyatakan keputusan dan daun menyatakan solusi. *Pseudocode* untuk *Tree Building* ditunjukkan pada Gambar 2.2.

```
Buat tree (Training Data T)
    partisi (T);
Partisi (Data S)
    If (semua poin di S sama dalam satu
    kelas)) then return;
    Bagi evaluasi untuk setiap atribut A
    Gunakan best split yang didapat untuk
    membagi S menjadi S1 dan S2;
```

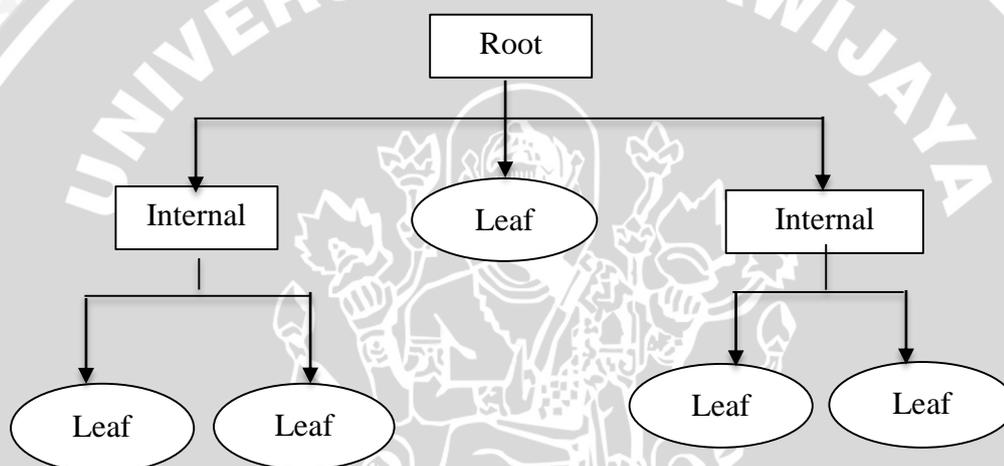
Gambar 2. 2 Tree Building Algorithm.

Sumber : (Mehta, Agrawal, dan Rissanen, 1996).

2.9.2 Model Pohon Keputusan

Pohon keputusan adalah model prediksi menggunakan struktur pohon atau struktur berhirarki. *Decision tree* merupakan metode klasifikasi yang paling populer digunakan. Selain karena pembangunannya relatif cepat, hasil dari model yang dibangun mudah untuk dipahami. Pada *decision tree* terdapat 3 jenis *node*, yaitu (Larose, 2005) :

- Root Node*, merupakan *node* paling atas, pada *node* ini tidak ada *input* dan bisa tidak mempunyai *output* atau mempunyai *output* lebih dari satu.
- Internal Node*, merupakan *node* percabangan, pada *node* ini hanya terdapat satu *input* dan mempunyai *output* minimal dua.
- Leaf node* atau *terminal node*, merupakan *node* akhir, pada *node* ini hanya terdapat satu *input* dan tidak mempunyai *output*. Model pohon keputusan ditunjukkan pada gambar 2.2.



Gambar 2. 3 Model Pohon Keputusan.

Sumber : Larose (2005)

2.10 Algoritma Supervised Learning in Quest (SLIQ)

Algoritma SLIQ (*Supervised Learning in Quest*) ini dikenalkan oleh Manish Mehta pada tahun 1996 dan kemudian dikembangkan oleh IBM dan menjadi suatu proyek di IBM. Kemunculan algoritma ini merupakan jawaban dari kekurangan algoritma-algoritma sebelumnya yang memiliki keterbatasan memori untuk *dataset* dalam jumlah yang besar (Mehta, Agrawal, dan Rissanen, 1996).

SLIQ merupakan *decision tree* pengklasifikasian yang dapat menangani atribut numerikal ataupun kategorikal. SLIQ menggunakan teknik *pre-sorting* di fase *tree-growth* untuk mengurangi *cost* evaluasi atribut numerik. Prosedur *sorting* ini diintegrasikan dengan strategi *breadth-first tree growing* untuk memungkinkan SLIQ melakukan klasifikasi pada *dataset* yang ada dalam *disk*. Selain itu SLIQ menggunakan algoritma *fast subnetting* untuk menentukan *split point* pada atribut kategorikal. Algoritma ini murah dan menghasilkan pohon

yang kompak dan akurat. Kombinasi dari teknik-teknik di atas menyebabkan SLIQ mampu untuk menangani *dataset* yang besar dan mengklasifikasikan data sets yang memiliki banyak kelas, atribut, dan record (Mehta, Agrawal, dan Rissanen, 1996).

2.10.1 Tahap Penyelesaian Algoritma SLIQ

2.10.1.1 Pre-Sorting dan Breadth-First Growth

Untuk atribut yang bertipe numerik, maka untuk melakukan *split* pada *decision tree node* harus dilakukan pengurutan (*sorting*) terlebih dahulu. Pada algoritma lainnya, proses pengurutan ini dilakukan pada setiap penentuan *node*-nya. Tetapi pada algoritma SLIQ ini, proses pengurutan data latih dilakukan hanya sekali pada saat awal dari fase pembentukan pohon.

Proses *pre-sorting* ini diawali dengan cara membuat daftar dari setiap atribut yang ada dalam data latih dan sebuah daftar lain yang disebut daftar kelas (*class list*) untuk menampung label kelas dari data tersebut. Setiap entri dari daftar atribut terdiri dari 2 *fields*, yaitu nilai atribut dan indeks dari daftar kelas. Daftar kelas sendiri juga punya 2 *field*, yaitu label kelas dan referensi ke *node* daun dari *decision tree*. Entri ke-*i* dari daftar kelas akan berkorespondensi dengan contoh ke-*i* pada training data. Setiap *node* daun pada *decision tree* merepresentasikan pemisah dari data latih. Dengan demikian daftar kelas setiap waktu bisa mengidentifikasi termasuk dalam pemisah mana sebuah contoh data. Konsep ini mengasumsikan terdapat jumlah memori yang cukup untuk menampung daftar kelas. Sedangkan kelas atribut bisa ditulis ke dalam *disk* jika diperlukan. Mula-mula *field* merujuk pada daun untuk semua masukan di daftar kelas yang mengarah pada *root* dari *decision tree*. Lalu langkah selanjutnya mendistribusikan nilai-nilai dari setiap atribut yang ada pada daftar. Setiap nilai atribut juga dilabeli dengan indeks dari daftar kelas yang berkorespondensi. Daftar atribut untuk nilai numerik kemudian diurutkan terpisah. Pada Tabel 2.3 menunjukkan contoh dari sebuah *pre-sorting*.

Tabel 2. 3 Contoh Pre-Sorting

Training Data			Class List		After Pre-Sorting		Class List		
Age	Salary	Class	Age	Index	Salary	Index	Class	Leaf	
30	65	G	23	2	15	2	1	G	N1
23	15	B	30	1	40	4	2	B	N1
40	75	G	40	3	60	6	3	G	N1
55	40	B	45	6	65	1	4	B	N1
55	200	G	55	5	75	3	5	G	N1
45	60	G	55	4	100	5	6	G	N1

Sumber : Mehta, Agrawal, dan Rissanen (1996)

Proses Split Node : Jika pada klasifikasi *decision-tree* sebelumnya dilakukan strategi *depth first*, maka pada algoritma SLIQ memakai strategi *breadth-first*. Konsekuensinya *split* pada semua *leaf* dari pohon akan dievaluasi secara simultan setiap satu data dimasukkan (Mehta, Agrawal, dan Rissanen, 1996). Untuk menghitung gini *splitting-indeks* dari sebuah atribut dari suatu *node*, maka kita membutuhkan distribusi frekuensi dari nilai kelas yang ada pada pembagian data yang berkorespondensi dengan *node*. Distribusi diakumulasikan dalam sebuah histogram kelas yang terhubung pada setiap *node* daun. Untuk atribut numerik, histogram berbentuk daftar pasangan yang terdiri dari $\langle \text{kelas}, \text{frekuensi} \rangle$. Sedangkan untuk atribut kategorikal, histogram ini berbentuk daftar dari 3 hal yaitu $\langle \text{nilai atribut}, \text{kelas}, \text{frekuensi} \rangle$. Daftar atribut ini dilakukan dalam satu waktu. Pada persamaan (2.2) merupakan formula untuk menghitung *Gini*, di mana p_j adalah frekuensi relative dari kelas j pada D (Han, Kamber, 2006) :

$$gini(D) = 1 - \sum_{i=1}^m p_i^2 \quad (2.2)$$

Sedangkan untuk persamaan gini split diilustrasikan pada persamaan 2.3 (Han, Kamber, 2006) :

$$Gini_A(D) = \frac{|D_1|}{|D|} \cdot Gini(D_1) + \frac{|D_2|}{|D|} \cdot Gini(D_2) \quad (2.3)$$

Di mana $Gini(D_1)$ adalah gini indeks untuk split pertama dan $Gini(D_2)$ adalah gini indeks untuk split kedua. $|D_1|$ dan $|D_2|$ merupakan frekuensi terjadinya split pertama dan kedua pada data latih, sedangkan $|D|$ adalah jumlah data latih.

Meng-update Daftar Kelas : Langkah selanjutnya adalah membuat *node* anak untuk setiap *node* daun dan meng-*update* daftar kelas. Pada gambar 2.3 merupakan *pseudocode* untuk proses *update* :

```

UpdateLabel()
for each attribute A used in a split do
  traverse attribute list of A
  for each value v in the attribute list do
    find the corresponding entry in the class list
    say e)
    find the new class c to which v belongs by
    applying
      the splitting test at node referenced from
      e
    update the class label for e to c
    update node referenced in e to child
  
```

Gambar 2. 4 Pseudocode untuk meng-update daftar kelas

Sumber : Mehta, Agrawal, dan Rissanen (1996)

Optimalisasi : Dalam pembentukan pohon, kedua langkah tersebut di atas dilakukan berulang-ulang sampai setiap node daun menjadi *pure node* (berisi satu *record* yang termasuk dalam satu kelas saja) dan tidak bisa dilakukan *split node* lagi. Perlu diketahui bahwa beberapa *node* mungkin saja menjadi *pure node* terlebih dahulu dibandingkan lainnya dan akan lebih baik jika daftar atribut disederhanakan dengan membuang semua entri yang berkorespondensi dengan *record* yang termasuk dalam *pure node* tadi. Optimisasi ini dengan mudah diimplementasikan dengan cara menulis ulang daftar yang sudah disederhanakan. Informasi yang diperlukan untuk mengambil keputusan ini bisa diambil dari data sebelumnya (Mehta, Agrawal, dan Rissanen, 1996).

Hal penting yang bisa diambil dari proses *pre-sorting* dan *breadth-grow* di atas adalah startegi ini yang memungkinkan algoritma SLIQ bisa menangani *dataset* dalam jumlah besar tanpa kehilangan tingkat akurasi. Ini dikarenakan kumpulan *split* yang dievaluasi dengan atau tanpa *pre-sorting* adalah identik. *Pre-sorting* dilakukan untuk mengeliminasi proses pengurutan data pada setiap node dan menyingkirkan batasan bahwa data latih harus *memory-resident* (Mehta, Agrawal, dan Rissanen, 1996).

2.10.1.2 Men-subset untuk kategori atribut

Split untuk atribut bertipe kategorikal A adalah bentuk dari $A \in S'$, dimana $S' \subset S$ dan S kemungkinan memiliki nilai-nilai dari atribut A . Evaluasi untuk semua *subset* dari S bisa jadi lama prosesnya, apalagi jika jumlah elemen (*cardinality*) dari S besar (Mehta, Agrawal, dan Rissanen, 1996).

SLIQ menggunakan pendekatan *hybrid* untuk mengatasi hal ini. Jika *cardinality* dari S lebih kecil dari *threshold* (MAXSETSIZE), maka semua *subset* dari S akan dievaluasi. Jika terjadi sebaliknya, maka algoritma *greedy* digunakan untuk mendapatkan *subset* yang diinginkan. Algoritma *greedy* dimulai dengan *subset* kosong S' dan menambahkan satu elemen dari S ke dalam S' yang akan memberikan *split* terbaik. Proses ini diulang sampai tidak ada perbaikan lagi dari *split*. Pendekatan *hybrid* ini menemukan *subset* optimal jika S kecil dan memberikan hasil baik pula pada *subset* yang lebih besar (Mehta, Agrawal, dan Rissanen, 1996).

2.11 Perhitungan Akurasi

Perhitungan akurasi dilakukan untuk mengetahui keakuratan sistem dalam mengklasifikasikan studi kasus yang diangkat dalam penelitian. Akurasi merupakan presentasi kebenaran antara jumlah data uji benar dengan total data uji secara keseluruhan. Persamaan untuk menghitung akurasi dinyatakan dalam persamaan (2.4) berikut ini (Kuncoro, 2012) :

$$\text{Akurasi} = \frac{\sum \text{Data Uji Benar Diklasifikasi}}{\sum \text{Data Keseluruhan}} * 100\% \quad (2.4)$$

Jumlah data uji benar merupakan jumlah data yang hasil prediksinya sama dengan kelas awal atau kelas sebenarnya. Sedangkan jumlah data uji adalah jumlah seluruh data uji yang diprediksi kelasnya.

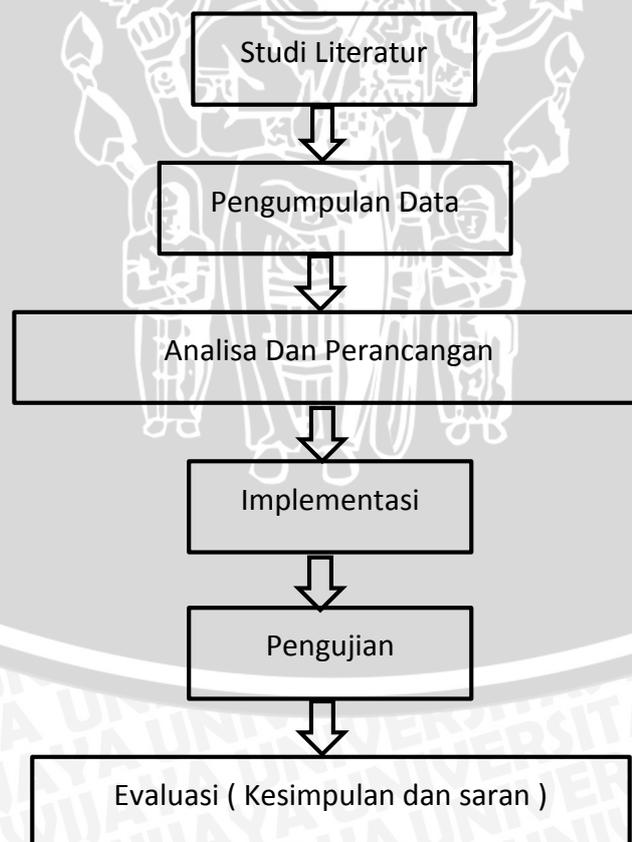


BAB 3 METODOLOGI DAN PERANCANGAN

Makna dari metodologi penelitian dapat dilihat dari dua sudut pandang. Pertama, dari pandangan umum dia bisa berarti sebuah cara sistematis untuk menyelesaikan masalah penelitian. Dalam hal ini dia juga dapat merupakan kumpulan cara (metode) yang lebih spesifik dalam penyelesaian masalah. Kedua, metodologi penelitian dapat dipahami sebagai sebuah ilmu untuk mempelajari bagaimana sebuah penelitian dilakukan secara sistematis. Dalam ilmu ini kita mempelajari berbagai langkah yang umumnya digunakan oleh peneliti ketika mempelajari masalah penelitian beserta alasan-alasan logis di belakangnya. Oleh karena itu di dalam pembahasan metodologi penelitian ini, akan membahas tentang langkah-langkah penelitian, perancangan sistem dan perhitungan manual menggunakan algoritma SLIQ.

3.1. Metodologi Penelitian

Pada bab ini di jelaskan langkah – langkah mengenai metodologi penelitian dan perancangan dalam penggunaan algoritma *Supervised Learning In Quest* untuk pengklasifikasi status gizi balita pada Kecamatan Kertosono Kabupaten Nganjuk. Metodologi penelitian yang diterapkan ditunjukkan pada Gambar 3.1.



Gambar 3. 1 Alur metodologi penelitian

Keterangan Gambar 3.1 :

1. Melakukan studi literatur mengenai penggunaan metode *Supervised Learning in Quest* (SLIQ)
2. Mengumpulkan data perkembangan status balita pada beberapa bidan desa serta Puskesmas Kertosono Kabupaten Nganjuk.
3. Melakukan analisa dan perancangan sistem dengan algoritma SLIQ dan indeks antropometri *z-score*.
4. Mengimplementasi hasil analisa dan perancangan dalam sebuah perangkat lunak.
5. Melakukan pengujian terhadap perangkat lunak.
6. Mengevaluasi hasil dari perangkat lunak tersebut. Nantinya berupa kesimpulan dan saran.

3.1.1. Studi Literatur

Studi literatur digunakan untuk mendukung teori-teori yang digunakan dalam penelitian klasifikasi status gizi balita disuatu wilayah. Teori – teori tersebut meliputi balita, gizi dan status gizi, penilaian status gizi, klasifikasi status gizi balita, data mining, *decision tree*, metode *Supervised Learning in Quest* (SLIQ), perhitungan akurasi yang diperoleh dari berbagai sumber meliputi sumber buku, jurnal, skripsi terdahulu dan *browsing* dari internet. Data yang diperoleh diolah sehingga dapat digunakan untuk analisis dalam memecahkan permasalahan klasifikasi status gizi balita dengan metode *Supervised Learning in Quest* (SLIQ).

3.1.2. Pengumpulan Data

Data penelitian yang digunakan bersumber dari data balita yang berstatus gizi di beberapa desa yang di peroleh dari bidan desa yang bersangkutan serta data gizi buruk akumulasi satu kecamatan pada Puskesmas Kertosono, di mana data ini lebih spesifik diambil dari 4 desa yang ada di Kertosono, kabupaten Nganjuk. Data latih yang digunakan sejumlah 361 data yang terdiri dari status gizi buruk, gizi kurang, gizi baik dan gizi lebih. Sedangkan untuk pengujian data yang digunakan sejumlah 142 berpa jenis kelamin, umur, berat badan dan tinggi badan untuk setiap balita. Data ini merupakan data perkembangan balita yang diperoleh dari beberapa bidan desa serta dari Puskesmas Kertosono, Kabupaten Nganjuk.

3.1.3. Analisa dan Perancangan

Analisa kebutuhan bertujuan untuk menganalisis dan mendapatkan kebutuhan yang diperlukan dalam membangun sebuah aplikasi. Sedangkan perancangan sistem dibangun berdasarkan hasil pengambilan data dan analisis kebutuhan yang telah dilakukan. Pada perancangan sistem ini dilakukan untuk mempermudah implementasi, pengujian, dan analisis.

3.1.3.1. Kebutuhan perangkat

1. Perangkat keras
Perangkat keras (*hardware*) yang digunakan untuk implementasi penelitian ini ada pada Tabel 3.1.

Tabel 3. 1 Spesifikasi Perangkat Keras

Jenis	Notebook
Prosesor	Intel Pentium @ 1.80 GHz 64 bit
Memory	4GB
Hardisk	500GB
Network	WiFi 802.11n, Bloetooth

2. Lingkungan *Software*

Perangkat lunak yang digunakan untuk implementasi penelitian ini adalah :

- Sistem Operasi Windows 8
- Editor Java (Netbean 8.0.2)
- JDK 8 update 45 64bit
- Xampp 5.5.19-0-VC11 installer 32bit

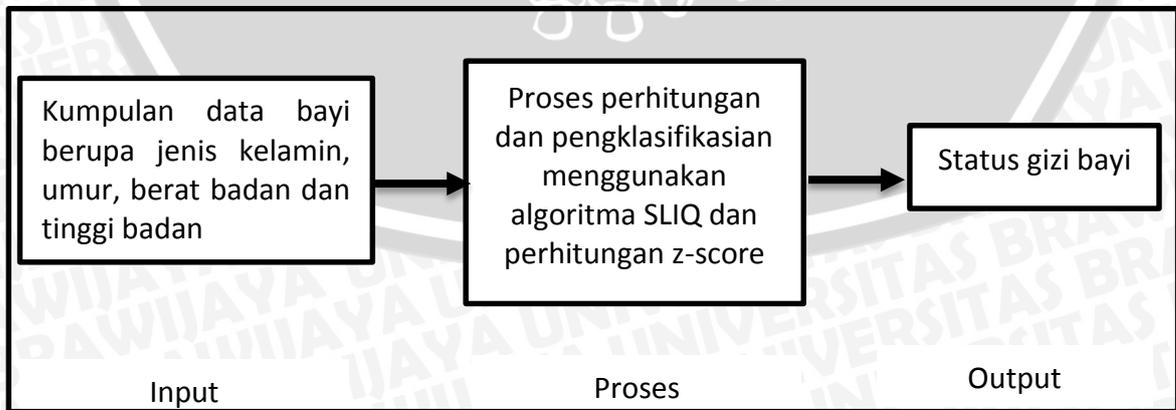
3.1.3.2. **Kebutuhan fungsional**

Kebutuhan fungsional yang digunakan adalah sebagai berikut :

- Data balita berstatus gizi buruk, gizi kurang, gizi baik, gizi lebih
- Data nilai dan data jenis setiap kriteria

3.1.3.3. **Arsitektur Sistem Pengklasifikasian**

Arsitektur sistem pengklasifikasian digambarkan dengan diagram blok. Diagram blok pada sistem ini menggambarkan aliran proses dari sebuah sistem secara terstruktur. Diagram blok menjelaskan cara kerja dari sistem secara umum, mulai dari tahap input, proses, hingga menghasilkan keluaran (*output*). Secara garis besar perancangan diagram blok terlihat seperti pada Gambar 3.2 yang terdiri dari beberapa blok diagram.



Gambar 3. 2 Gambar Diagram blok sistem pengklasifikasian data bayi menggunakan SLIQ



3.1.4. Implementasi

Implementasi ini mengacu pada perancangan sistem yang telah dibentuk. Pada tahap ini menjelaskan tentang penerapan metode *Supervised Learning in Quest (SLIQ)* untuk klasifikasi status gizi bayi. Proses pengimplementasian perangkat lunak ini menggunakan bahasa pemrograman *java* dan Netbeans sebagai *compiler*-nya, manajemen DBMS MySQL, serta Xampp. serta *tools* pendukung lain yang digunakan. Berikut merupakan tahapan yang ada pada implementasi aplikasi :

1. Pembuatan antarmuka.
2. Penerapan metode *SLIQ* pada sistem untuk proses klasifikasi status gizi bayi
3. Keluaran yang dihasilkan oleh aplikasi ini nantinya berupa status gizi individu dari perhitungan *SLIQ*.

3.1.5. Pengujian

Tahap pengujian ini dilakukan untuk mengetahui tingkat akurasi atau keberhasilan sistem yang telah dibuat penerapan metode yang digunakan. Pengujian yang diterapkan dalam aplikasi ini adalah pengujian akurasi. Pengujian akurasi dilakukan untuk menguji tingkat akurasi dari perangkat lunak. Pengujian akurasi dilakukan dengan membandingkan data dari sistem dengan data yang terklasifikasi. Scenario pengujian lebih lanjut dibahas pada Bab 5 tentang pengujian sistem.

3.1.6. Evaluasi (Kesimpulan dan Saran)

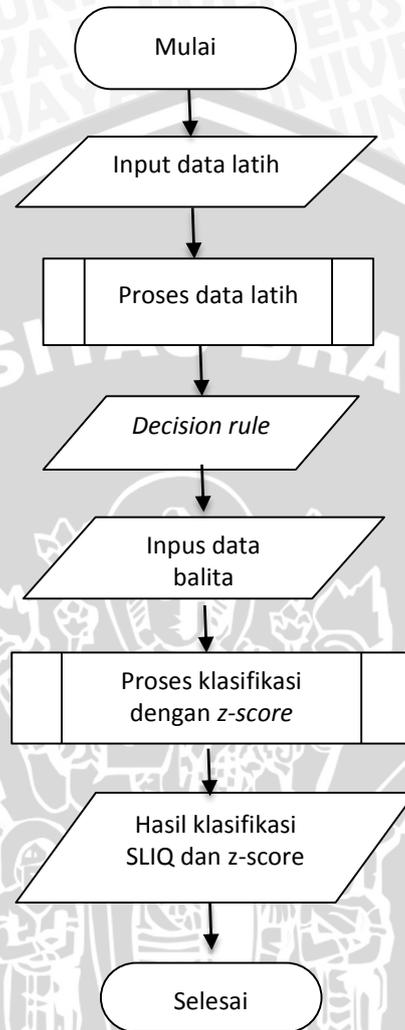
Tahap yang terakhir adalah evaluasi. Evaluasi disini berupa kesimpulan dan saran. Pengambil kesimpulan dan saran dilakukan apabila semua tahapan pada perancangan, implementasi, dan pengujian telah selesai dilakukan. Kesimpulan didapatkan berdasarkan dari hasil pengujian dan analisis terhadap sistem yang dibangun. Penulisan saran berguna untuk memperbaiki kesalahan-kesalahan yang terjadi serta memberikan pertimbangan jika ada pengembangan sistem selanjutnya.

3.2. Perancangan Sistem

Dalam penelitian ini, sistem yang akan dibuat digunakan untuk mengklasifikasikan status gizi balita balita tersebut termasuk ke dalam golongan gizi buruk, gizi kurang, gizi baik atau gizi lebih. Data yang digunakan diperoleh dari data balita di beberapa desa di kecamatan Kertosono. Dalam sistem ini pengguna cukup memasukkan aturan nya kemudian memasukkan data balita yang akan diklasifikasikan status gizi nya saja. Kemudian pengguna bisa langsung mengetahui balita tersebut termasuk ke golongan mana. Selain penerapan algoritma *SLIQ*, sistem ini nantinya juga mempunyai perhitungan *z-score* juga untuk mengetahui status gizi balita.

Berdasarkan analisis yang telah dilakukan, secara umum sistem pengklasifikasian data bayi ini menggunakan metode SLIQ memiliki beberapa tahap seperti pada gambar 3.3.

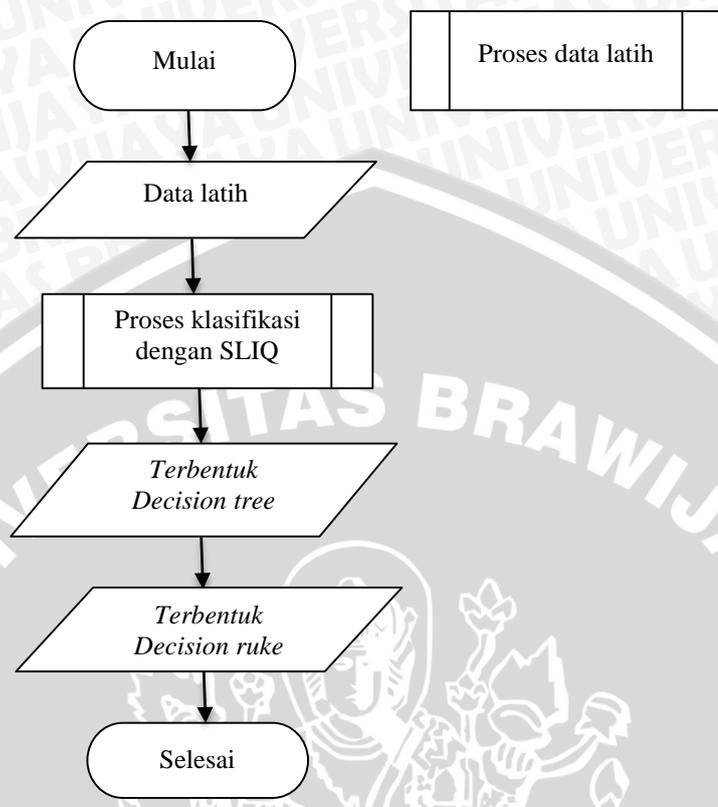
Dari Gambar 3.3 diagram alir sebelum melakukan perhitungan algoritma



Gambar 3. 3 Diagram alir sistem pengklasifikasian status bayi menggunakan SLIQ

SLIQ untuk menentukan klasifikasi, data latih yang telah disiapkan di latih dulu untuk menentukan *decision rule* dari perhitungan manual. Data latih yang telah di proses ini lah nantinya yang akan menjadi acuan pada data uji.

Untuk input data bayi yang digambarkan di atas, berupa nama, jenis kelamin, umur, berat badan dan tinggi badan yang harus dimasukkan untuk memperoleh klasifikasi status gizi bayi.



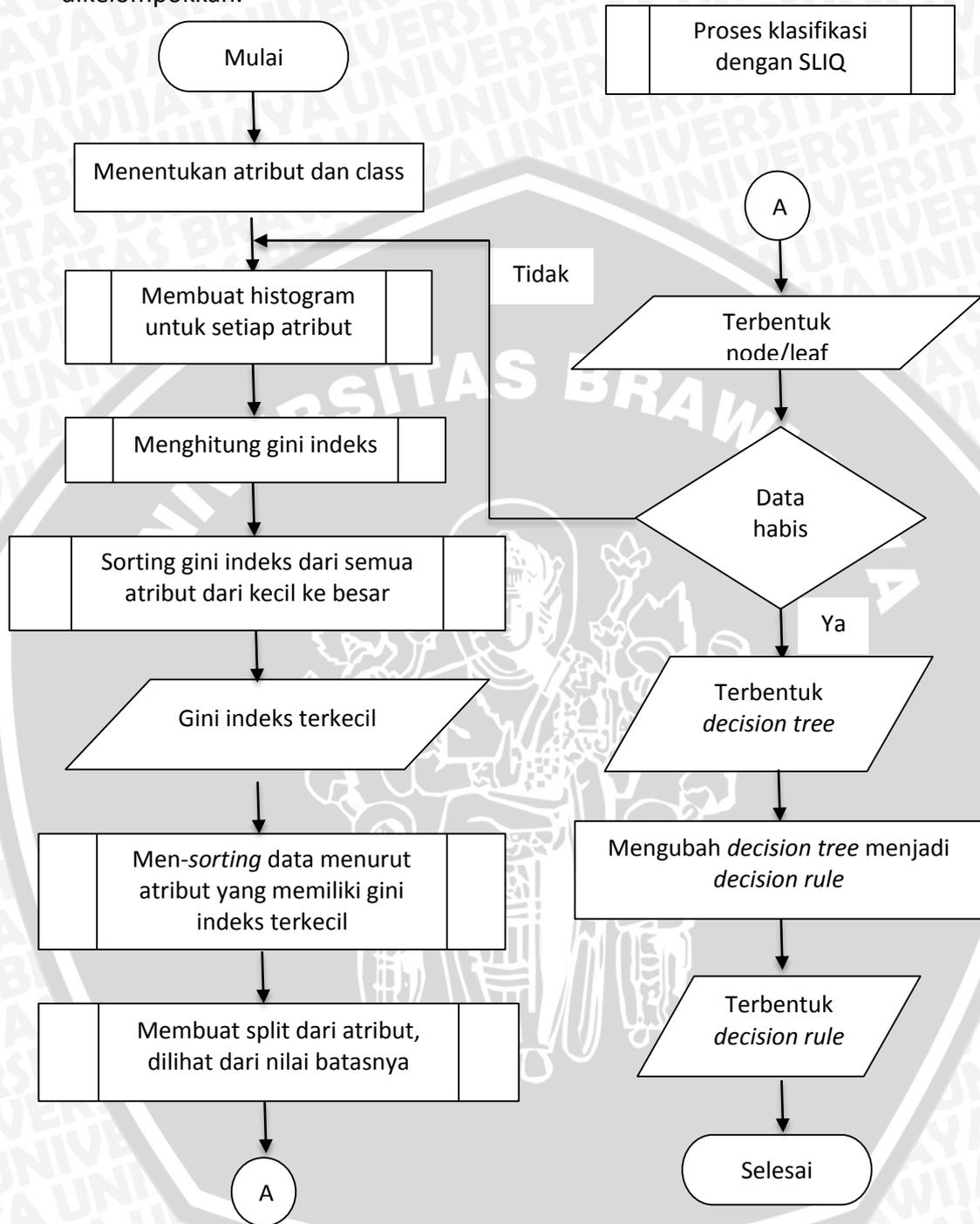
Gambar 3. 4 Diagram alir sub proses proses data latih

Proses data latih di sini berfungsi untuk pengambilan struktur *tree* dan *rule* yang digunakan untuk tahap pengujian selanjutnya pada sistem ini. Subproses untuk data atih ditunjukkan pada diagram alir Gambar 3.4.

Setelah data-data bayi yang dibutuhkan selesai dimasukkan, barulah tahapan selanjutnya yaitu melakukan proses pembuatan *decision tree* menggunakan algoritma SLIQ. Selain menggunakan algoritma SLIQ, pada sistem nantinya juga akan di terapkan perhitungan *z-score* untuk mengetahui status gizi bayi. Perhitungan *z-score* di sini di sertakan sebagai perbandingan untuk algoritma SLIQ, apakah nantinya algoritma ini baik diterapkan untuk kasus seperti ini. Proses *decision tree* menggunakan algoritma SLIQ digambarkan pada gambar 3.5.

Dari gambar 3.4 yang menjelaskan tentang langkah-langkah perhitungan algoritma SLIQ, yang pertama yaitu menentukan *atribut list* dan *class list* dari data. Selanjutnya data dikelompokkan menurut atribut dan *class* yang telah ditentukan, kemudian di-*sorting* dari nilai yang terkecil untuk data numerik, untuk data kategorikal tidak perlu melakukan *sorting*. Langkah selanjutnya yaitu membuat histogram untuk setiap atribut. Histogram adalah tabel yang menyimpan frekuensi dari kemunculan kelas dalam atribut tersebut.

Selanjutnya adalah menghitung nilai gini indeks dari histogram yang telah dikelompokkan.



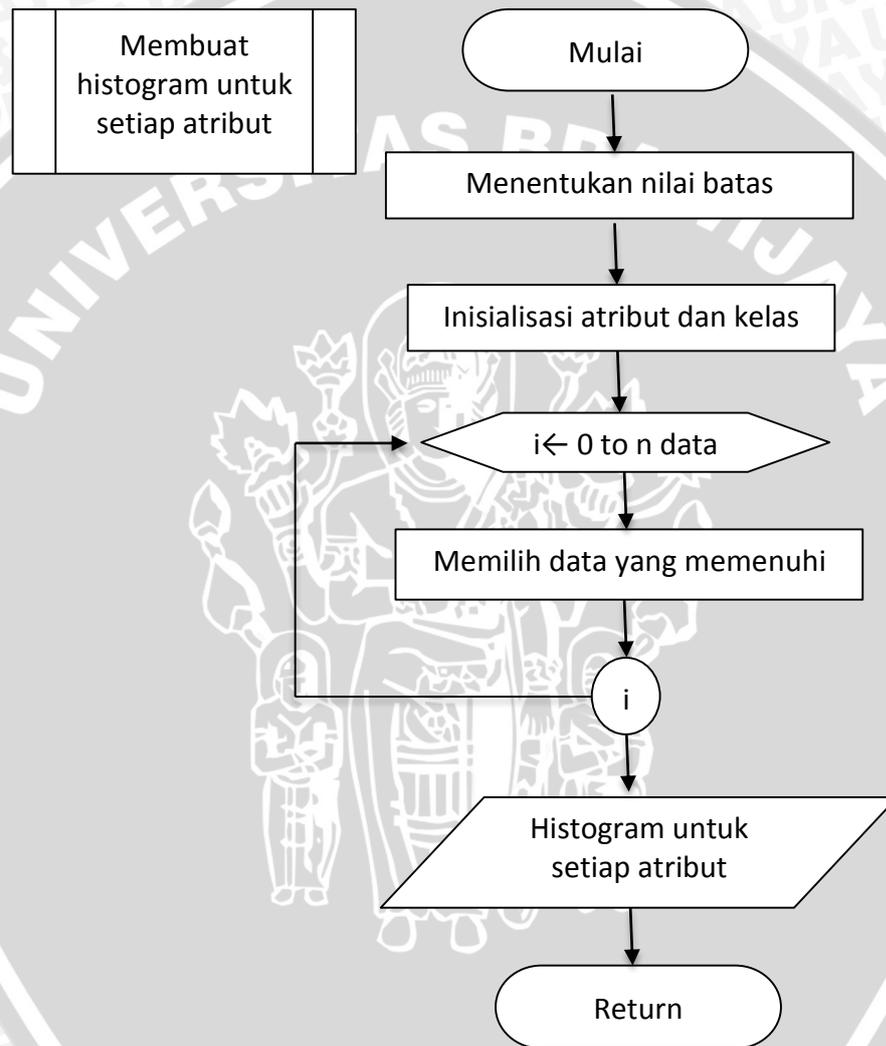
Gambar 3. 5 Diagram alir sub proses algoritma SLIQ

Dari nilai gini yang dihasilkan, nilai terkecil yang dijadikan acuan untuk *sorting*. *Sorting* disini dilakukan menurut atribut yang memiliki gini terkecil. Dari *sorting* ini tadi dilihat, data mana yang memenuhi syarat yang ditentukan dari histogram, kemudian atribut dari data tersebut menjadi *node/leaf* untuk *decision tree*. Proses ini dilakukan berulang hingga didapatkan *decision tree*



untuk keseluruhan data. Langkah terakhir setelah membuat *decision tree* adalah membuat *decision rule* dari *decision tree* tersebut.

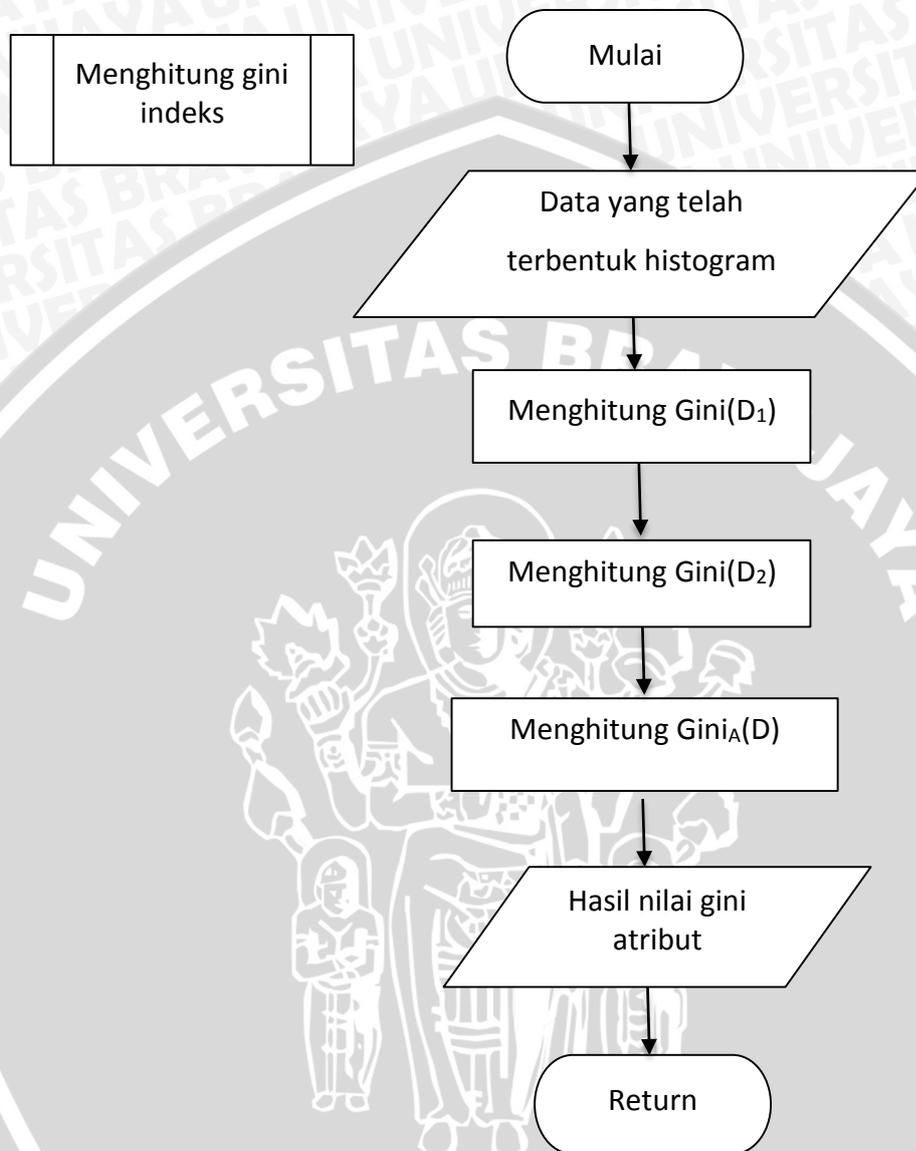
Gambar 3.6 menunjukkan sub proses untuk pembuatan histogram dari setiap atribut. Histogram ini berguna untuk menentukan atau menghitung nilai gini dari jumlah data yang ada. Sebelum menghitung histogram, nilai batas untuk masing-masing atribut harus ditentukan lebih dahulu, baru kemudian menjabarkan nilai histogram menurut atribut, kelas dan batas yang telah ditentukan.



Gambar 3. 6 Diagram alir sub program histogram

Setelah histogram terbentuk, tahapan perhitungan SLIQ selanjutnya adalah menghitung gini indeks dari setiap batas. Diagram alir sub proses untuk menghitung gini indeks ditunjukkan pada Gambar 3.7. Data yang telah terbentuk pada tabel histogram, selanjutnya dihitung menggunakan persamaan Gini(D_1) dan Gini (D_2). Persamaan ini diimplementasikan ke dalam program. Kemudian barulah dihitung melalui persamaan $Gini_A(D)$ untuk satu

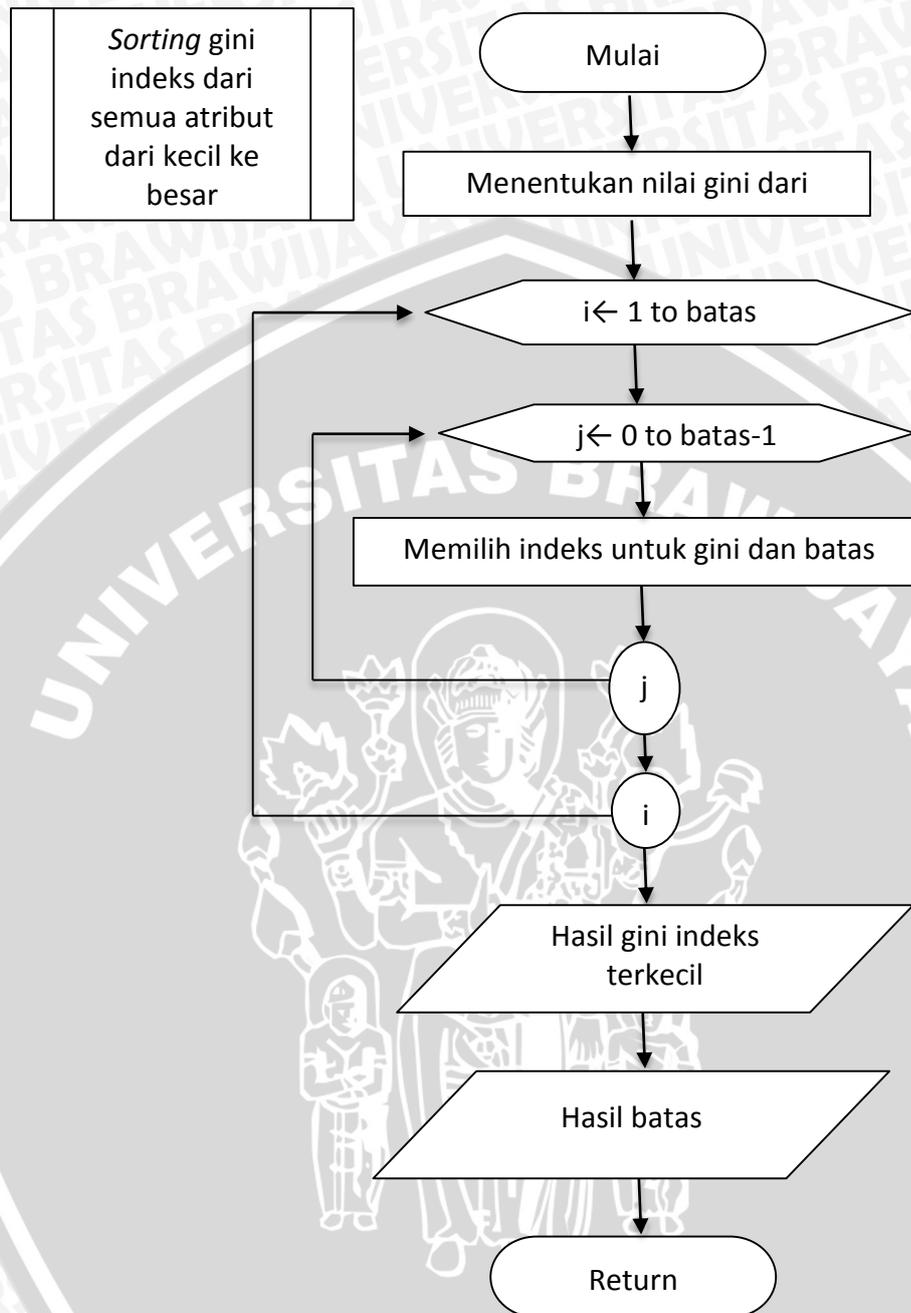
batas pada satu atribut. Hasil inilah yang kemudian di urutkan dari yang terkecil ke yang terbesar.



Gambar 3. 7 Sub Program menghitung gini indeks

Sedangkan untuk sub proses sorting gini indeks dari setiap batas ditunjukkan pada Gambar 3.8. nilai gini dari batas yang telah dihitung, kemudian di sorting ntuk mengetahui gini terkecil yang kemudian dijadikan split. Hasil akhirnya, akan diperoleh gini dengan urutan dari yeng terkecil, pada masing masing atribut.

Gini indeks terkecil yang dihasilkan dari perhitungan sebelumnya kemudian gini inilah yang menjadi acuan untuk menentukan split. Namun sebelumnya, data yang memiliki gini indeks terkecil harus di urutkan dari nilai yang kecil ke besar.

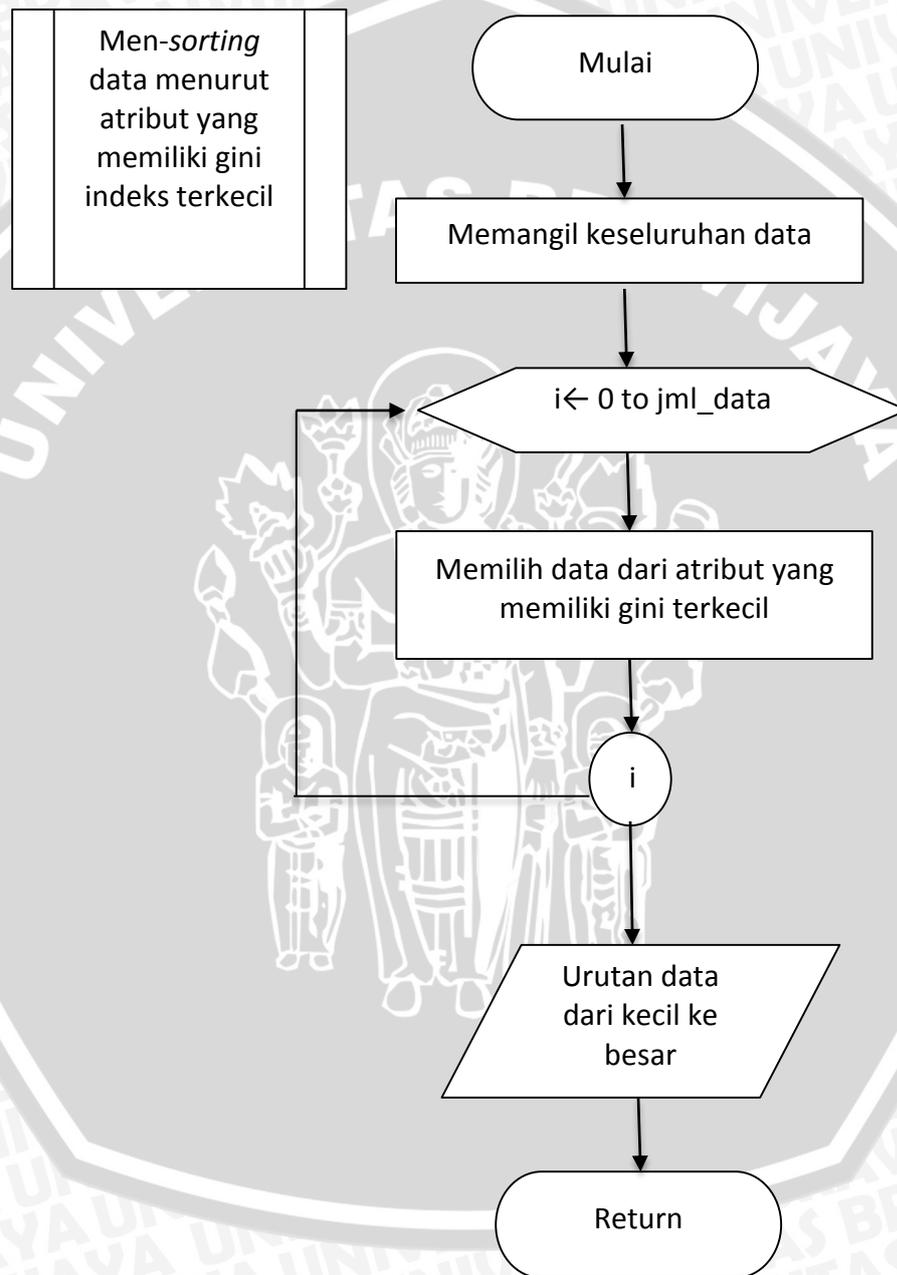


Gambar 3. 8 Sub Proses sorting gini indeks dari kecil ke besar

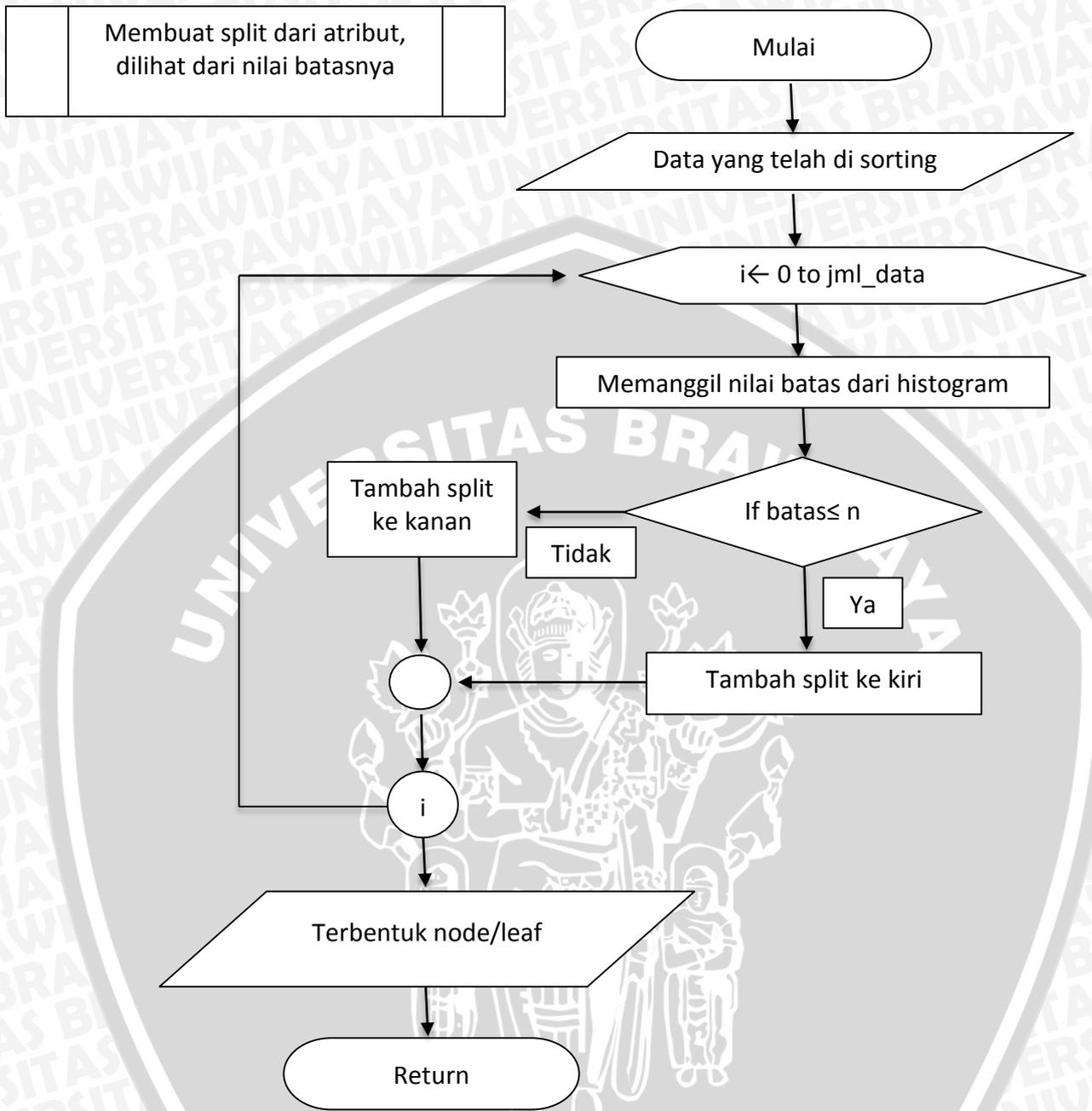
Hal tersebut dilakukan karena batas yang memiliki nilai gini terkecil yang di ambil untuk membelah data menjadi dua bagian. Hal inilah yang disebut sebagai pembentukan split. *Sorting* data menurut gini terkecil dapat dilihat pada diagram alir Gambar 3.9. diagram ini menunjukkan jalannya *sorting* data yang nantinya diimplementasikan pada program *sorting* data keseluruhan.

Sub proses selanjutnya adalah *split* dari atribut dilihat dari batasnya (lihat Gambar 3.10). Maksudnya adalah split diambil dari atribut yang memiliki nilai

gini terkecil, kemudian dilihat gini tersebut berada pada indeks batas ke berapa. Misal untuk nilai batas 9 pada atribut umur. Jika nilai batas ≤ 9 maka *split* akan berada di sebelah kiri, dan jika batas bernilai > 9 maka *split* berada di bagian kanan. Begitupun seterusnya untuk nilai batas pada atribut yang lain. Hal ini akan diulang selama data masih ada dan membentuk *node/leaf* baru, sehingga terbentuk menjadi satu kesatuan *tree*.

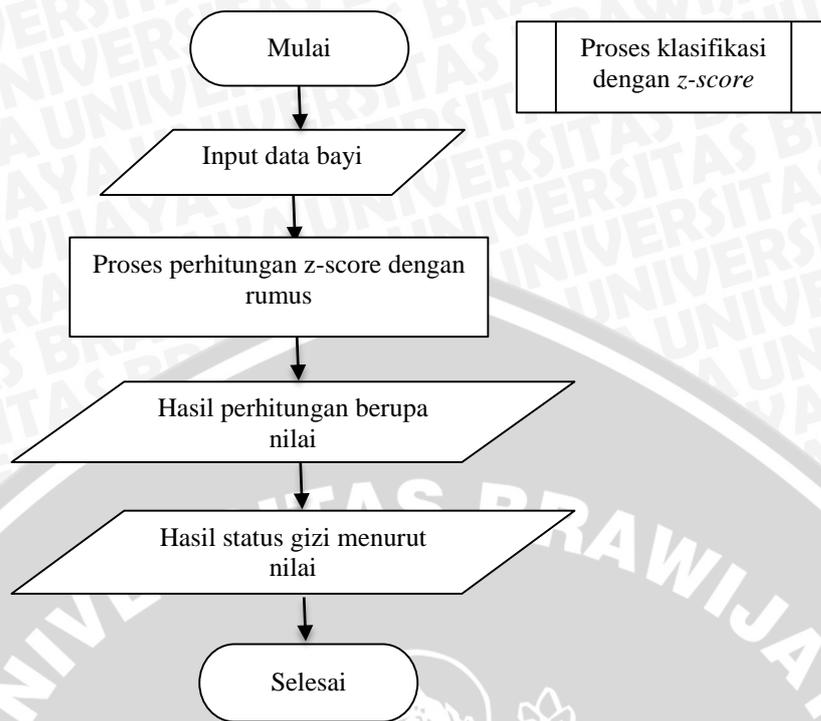


Gambar 3. 9 Sub Proses sorting data menurut gini terkecil



Gambar 3. 10 Sub Proses Spit

Selanjutnya penentuan status gizi dilihat dari perhitungan z-score. Tahapan untuk perhitungan z-score ditunjukkan pada gambar 3.11. Masukan data yang dibutuhkan untuk perhitungan ini sama dengan masukan untuk perhitungan menggunakan algoritma SLIQ. Rumus yang digunakan untuk perhitungan z-score telah dijelaskan pada Bab II. *Output* yang diperoleh dari diagram alir pada Gambar 3.11 adalah nilai z-score yang berupa angka dan status gizi balita.



Gambar 3. 11 Diagram alir perhitungan Z-Score

3.3. Perancangan Stuktur Tabel pada Basisdata

Subbab ini menjelaskan tentang struktur tabel yang digunakan dalam basisdata sebagai sumber penyimpanan data latih maupun data uji. Untuk *tool*/DBMS menggunakan phpMyadmin dengan DBMS MySQL, sedangkan untuk pengaksesan *database* menggunakan XAMPP.

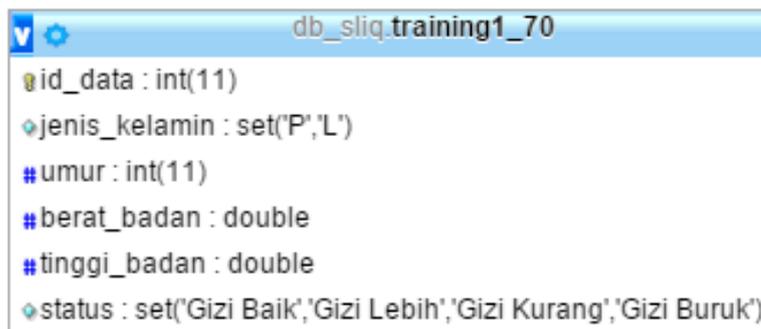
3.3.1.1. Tabel data_training

Tabel *data_balita* di sini merupakan tabel data latih / data latih yang digunakan untuk kepentingan sistem. Tabel ini menyimpan data balita-balita yang berupa jenis kelamin balita, umur, berat bada, tinggi badan dan status gizi balita. Struktur tabel ditunjukkan pada Tabel 3.3 dan Gambar 3.12.

Tabel 3. 2 Tabel data_balita

Field	Jenis	Keterangan
id_balita	Integer(11)	Kode data
jenis_kelamin	Set 'L' & 'P'	Jenis kelamin balita
Umur	Integer(11)	Umur balita
berat_badan	Double	Berat badan balita
tinggi_badan	Double	Tinggi badan balita
Status	Set (Gizi baik, gizi buruk, gizi kurang, gizi lebih)	Status gizi balita





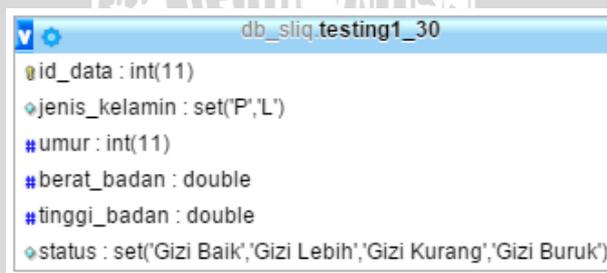
Gambar 3. 12 Gambar desain tabel data training pada database

3.3.1.2. Tabel data_testing

Tabel ini berfungsi untuk menyimpan keseluruhan data yang akan digunakan untuk pengujian. Atribut dari tabel ini ditunjukkan pada Tabel 3.4 dan Gambar 3.13.

Tabel 3. 3 Tabel data_testing

Field	Jenis	Keterangan
id	Integer(11)	Kode data
jk	Set 'L' & 'P'	Jenis kelamin balita
Umur	Integer(11)	Umur balita
bb	Double	Berat badan balita
tb	Double	Tinggi badan balita
status	Set (Gizi baik, gizi buruk, gizi kurang, gizi lebih)	Status gizi balita



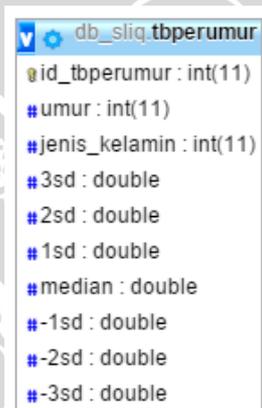
Gambar 3. 13 Gambar struktur data tabel data_testing

3.3.1.3. Tabel tbperumur

Tabel tbperumur menyimpan data-data standar tetapan rentang gizi untuk balita yang dilihat dari tinggi badan menurut umur. Tabel ini berfungsi untuk menghitung nilai z-score. Struktur tabel ditunjukkan pada Tabel 3.5 dan Gambar 3.14.

Tabel 3. 4 struktur tabel tbperumur di database

Field	Jenis	Keterangan
id_tbperumur	Integer(11)	Kode data
jenis_kelamin	Integer(11)	Jenis kelamin balita
Umur	Integer(11)	Umur balita
3sd	Double	Standar deviasi +3
2sd	Double	Standar deviasi +2
1sd	Double	Standar deviasi +1
Median	Double	Nilai tengah
-1sd	Double	Standar deviasi -3
-2sd	Double	Standar deviasi -2
-3sd	Double	Standar deviasi -1



Gambar 3. 14 Desain tabel tbperumur pada basisdata

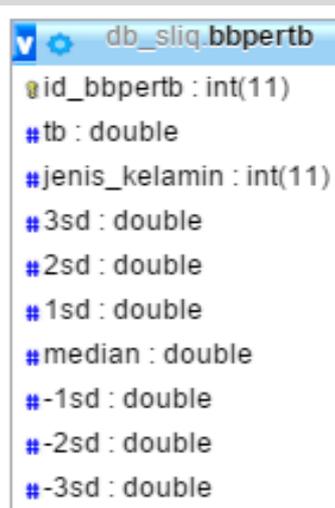
3.3.1.4. Tabel bbpertb

Sama hal nya dengan tabel tbperumur tabel bbpertb juga menyimpan data-data standar tetapan rentang gizi untuk balita yang dilihat dari berat badan menurut tinggi badan balita. Tabel ini juga berfungsi untuk menghitung nilai z-score. Struktur tabel ditunjukkan pada Tabel 3.6 dan Gambar 3.15.

Tabel 3. 5 Tabel bbpertb

Field	Jenis	Keterangan
id_bbpertb	Integer(11)	Kode data
tb	double	Tinggi badan
Umur	Integer(11)	Umur balita
3sd	Double	Standar deviasi +3

2sd	Double	Standar deviasi +2
1sd	Double	Standar deviasi +1
Median	Double	Nilai tengah
-1sd	Double	Standar deviasi -3
-2sd	Double	Standar deviasi -2
-3sd	Double	Standar deviasi -1



Gambar 3. 15 Desain tabel bbpertb pada basisdata

3.3.1.5. Tabel bbperumur

Tabel bbperumur menyimpan data-data standar tetapan rentang gizi untuk balita yang dilihat dari berat badan menurut umur. Tabel ini juga berfungsi untuk menghitung nilai z-score. Struktur tabel ditunjukkan pada Tabel 3.7 dan Gambar 3.16.

Tabel 3. 6 Tabel bbperumur

Field	Jenis	Keterangan
id_bbperumur	Integer(11)	Kode data
jenis_kelamin	Integer(11)	Jenis kelamin balita
Umur	Integer(11)	Umur balita
3sd	Double	Standar deviasi +3
2sd	Double	Standar deviasi +2
1sd	Double	Standar deviasi +1

Median	Double	Nilai tengah
-1sd	Double	Standar deviasi -3
-2sd	Double	Standar deviasi -2
-3sd	Double	Standar deviasi -1

```

db_sliq.bbperumur
#id_bbperumur : int(11)
#umur : int(11)
#jenis_kelamin : int(11)
#3sd : double
#2sd : double
#1sd : double
#median : double
#-1sd : double
#-2sd : double
#-3sd : double
    
```

Gambar 3. 16 Gambar desain tabel bbperumur pada database

3.4. Perancangan Antarmuka

Perancangan antarmuka pada klasifikasi status gizi balita ini sangat dibutuhkan untuk mempermudah pengguna dalam menjalankan aplikasi ini. Di bawah ini merupakan rancangan antarmuka dari aplikasi pengklasifikasian status gizi pada balita :

3.4.1. Modul Data Latih

KLASIFIKASI STATUS GIZI BALITA
MENGUNAKAN ALGORITMA *SUPERVISED LEARNING IN QUEST*

DATA TRAINING

Decision Rule

pilih tabel latih

Gambar 3. 17 Modul data latih



Antarmuka untuk modul data latih berupa table data latih dan kolom *decision rule*. Rule dari algoritma SLIQ ini diproses untuk menghasilkan data latih. Untuk rancangan antarmuka modul data latih ditunjukkan pada Gambar 3.17.

3.4.2. Modul Data Uji

Untuk melakukan pengujian dilakukan pada modul data uji disini. Pengguna nantinya akan memilih tabel data bayi yang akan di uji untuk menentukan klasifikasi dan akurasi. Pada Gambar 3.18 merupakan perancangan antarmuka modul data uji:

KLASIFIKASI STATUS GIZI BALITA
MENGUNAKAN ALGORITMA *SUPERVISED LEARNING IN QUEST*

pilih tabel uji

parameter uji	hasil
jumlah data uji	<input type="text"/>
jumlah data benar	<input type="text"/>
akurasi	<input type="text"/>

Gambar 3. 18 Antarmuka modul data uji

3.4.3. Modul klasifikasi

Kedua modul di atas merupakan modul dengan masukan data yang banyak, sedangkan modul klasifikasi yang ini akan memproses data per individu, jadi masukan dari modul ini cukup untuk satu individu saja. Data masukan yang diperlukan untuk perhitungan klasifikasi adalah nama balita, jenis kelamin, umur, berat badan dan tinggi badan balita. Hasil dari masukan tersebut berupa klasifikasi status gizi balita menggunakan SLIQ dan z-score. Modul klasifikasi ditunjukkan oleh Gambar 3.19 di bawah ini :

KLASIFIKASI STATUS GIZI BALITA
MENGUNAKAN ALGORITMA *SUPERVISED LEARNING IN QUEST*

DATA MASUKAN :

NAMA :

JENIS KELAMIN :

UMUR :

BERAT BADAN :

TINGGI BADAN :

HASIL KLASIFIKASI

ALGORITMA SLIQ :

INDEX ANTROPOMETRI :

NILAI Z-SCORE	STATUS Z-SCORE
BB/U <input type="text"/>	<input type="text"/>
TB/U <input type="text"/>	<input type="text"/>
BB/TB <input type="text"/>	<input type="text"/>

Gambar 3. 19 Modul Klasifikasi

3.5. Perhitungan Manual

Penerapan algoritma SLIQ akan dijelaskan dalam perhitungan manual dengan data latih pada Tabel 3.8 di bawah ini :

Tabel 3. 7 Tabel sampel data untuk perhitung manual

No	Nama Balita	Sex	Umur	Bb	Tb	Nilai Z_Score			Status Gizi			Status
		L=1,P=2	(Bln)	(Kg)	(Cm)	Bb / U	Tb / U	Bb / Tb	Bb / U	Tb / U	Bb / Tb	
1	Gusna Maratus	2	22	11.8	81	0.5	-1.16129	1.333333	Bb Normal	Normal	Normal	G. Baik
2	Sifaatul Khasanah	2	19	12	76	1.117647	-1.93333	2.5	Bb Normal	Normal	Gemuk	G. Baik
3	Flora Dwiyanti	2	17	6.7	73	-3.33333	-2.34483	-3.33333	Bb S.Krg	Pendek	S.Kurus	G.Buruk
4	M. Shohibul	1	16	9.1	74	-1.3	-2.4	-0.42857	Bb Normal	Pendek	Normal	G.Baik
5	Yuda Pratama	1	21	7.5	74	-3.7	-3.86207	-2.83333	Bb S.Krg	S.Pendek	Kurus	G.Buruk
6	Yeremia Yulius S K	1	19	7.7	76	-3.33333	-2.62963	-3	Bb S.Krg	Pendek	Kurus	G. Buruk
7	Ibrahim Islami Pasei	1	18	7.8	76	-3	-2.33333	-2.85714	Bb Krg	Pendek	Kurus	G. Kurang
8	M. Okta Fadhil	1	15	7.5	72.5	-2.88889	-2.64	-2.5	Bb Krg	Pendek	Kurus	G.Kurang
9	Aura Nabila	2	20	13.8	79	2.05	-1.23333	3.416667	Bb Lbh	Normal	Gemuk	G. Lebih
10	Kaila Isfiona	2	19	7	75	-3.33333	-2.26667	-3.33333	Bb S.Krg	Pendek	S.Kurus	G. Buruk
11	Eka Nurcyntaningrum	2	48	24	95	2.675676	-1.7907	6.722222	Bb Lbh	Normal	Gemuk	G. Lebih
12	M.A. Satria Natanegara	1	16	6.8	71	-3.77778	-3.6	-3.16667	Bb S.Krg	S.Pendek	S.Kurus	G. Buruk
13	Nicolas A. Y.	1	8	6.5	60	-2.57143	-4.81818	8.857143	Bb Krg	S.Pendek	Gemuk	G. Kurang
14	Gilang Putra R.	1	13	9	76	-0.81818	-0.375	3.818182	Bb Normal	Normal	Gemuk	G. Baik
15	Fajar Ismu K.	1	30	13	90	-0.2	-0.55882	4.714286	Bb Normal	Normal	Gemuk	G. Baik
16	Nadila Syafa	2	59	11.6	96	-3.25	-2.72917	-2.3	Bb S.Krg	Pendek	Kurus	G. Buruk
17	Alea	2	57	29	112	3.113636	0.893617	3.884615	Bb Lbh	Normal	Gemuk	G. Lebih
18	Yuda Pratama	1	21	19	88	4.647059	1	5.307692	Bb Lbh	Normal	Gemuk	G. Lebih
29	Siti Masruroh	2	19	7.9	80	-2.33333	-0.58621	-3	Bb Krg	Normal	Kurus	G. Kurang
20	Susi Saridewi	2	19	14	88	2.263158	2.133333	1.538462	Bb Lbh	Jangkung	Normal	G. Lebih
21	Dimas Adrianto	1	19	8	75.5	-3	-2.81481	-2.5	Bb Krg	Pendek	Kurus	G. Kurang
22	Fitriana	2	26	9	76.5	-2.4	-3.30303	-0.66667	Bb Krg	S.Pendek	Normal	G. Kurang
23	Amalia Anisah	2	21	7.1	75	-3.5	-2.83333	-3.16667	Bb S.Krg	Pendek	S.Kurus	G. Buruk

24	Mahardika Putra R.	1	18	7.6	73.5	-3.2	-3.25926	-2.6	Bb S.Krg	S.Pendek	Kurus	G. Buruk
25	Moh. Nafal A.	1	9	9	71	0.1	-0.43478	3.9	Bb Normal	Normal	Gemuk	G. Baik
26	Dewa Putra N.	1	26	19	96	3.6	2.258065	2.6	Bb Lbh	Jangkung	Gemuk	G. Lebih
27	Nurfarhan	1	21	12.2	83	0.5	-0.75	4.416667	Bb Normal	Normal	Gemuk	G. Baik
28	Imeida Reva P.	2	31	10.9	86	-1.38462	-1.52778	5.785714	Bb Normal	Normal	Gemuk	G. Baik
29	Arsa Aditya P.	2	23	12	83	0.466667	-0.78125	4.571429	Bb Normal	Normal	Gemuk	G. Baik
30	Aizin Azna	1	23	17	95	3.111111	2.7	2.133333	Bb Lbh	Jangkung	Gemuk	G. Lebih
31	Vanesa Meiga W.	2	15	15	83.5	3.529412	2.185185	2.615385	Bb Lbh	Jangkung	Gemuk	G. Lebih

Data di atas merupakan data latih yang akan digunakan untuk perhitungan manual algoritma SLIQ. Yang digunakan sebagai atribut adalah jenis kelamin, umur, bb (berat badan), dan tb (tinggi badan) dan untuk *class list* nya menggunakan kolom status. Sedangkan nilai *z-score* dan status gizi digunakan untuk nilai perbandingan. Yang dimaksud nilai perbandingan disini adalah membandingkan nilai akurasi yang dihasilkan oleh algoritma SLIQ dengan *z-score* dan status gizi (diperoleh dari perhitungan antropometri status gizi menurut teori dan ketetapan menteri kesehatan).

Tabel 3. 8 Tabel data balita

NO	NAMA BALITA	Sex	Umur	BB	TB	Status
		L=1,P=2	(Bln)	(Kg)	(Cm)	
1	Gusna Maratus	2	22	11.8	81	G. Baik
2	Sifaatul Khasanah	2	19	12	76	G. Baik
3	Flora Dwiyanti	2	17	6.7	73	G. Buruk
4	M. Shohibul	1	16	9.1	74	G. Baik
5	Yuda Pratama	1	21	7.5	74	G. Buruk
6	Yeremia Yulius S K	1	19	7.7	76	G. Buruk
7	Ibrahim Islami Pasei	1	18	7.8	76	G. Kurang
8	M. Okta Fadhil	1	15	7.5	72.5	G. Kurang
9	Aura Nabila	2	20	13.8	79	G. Lebih
10	Kaila Isfiona	2	19	7	75	G. Buruk
11	Eka Nurcyntaningrum	2	48	24	95	G. Lebih
12	M.A. Satria Natanegara	1	16	6.8	71	G. Buruk
13	Nicolas A. Y.	1	8	6.5	60	G. Kurang
14	Gilang Putra R.	1	13	9	76	G. Baik
15	Fajar Ismu K.	1	30	13	90	G. Baik
16	Nadila Syafa	2	59	11.6	96	G. Buruk
17	Alea	2	57	29	112	G. Lebih
18	Yuda Pratama	1	21	19	88	G. Lebih
29	Siti Masrurroh	2	19	7.9	80	G. Kurang
20	Susi Saridewi	2	19	14	88	G. Lebih
21	Dimas Adrianto	1	19	8	75.5	G. Kurang
22	Fitriana	2	26	9	76.5	G. Kurang
23	Amalia Anisah	2	21	7.1	75	G. Buruk
24	Mahardika Putra R.	1	18	7.6	73.5	G. Buruk
25	Moh. Nafal A.	1	9	9	71	G. Baik
26	Dewa Putra N.	1	26	19	96	G. Lebih
27	Nurfarhan	1	21	12.2	83	G. Baik
28	Imeida Reva P.	2	31	10.9	86	G. Baik
29	Arsa Aditya P.	2	23	12	83	G. Baik
30	Aizin Azna	1	23	17	95	G. Lebih
31	Vanesa Meiga W.	2	15	15	83.5	G. Lebih

Tabel 3.9 merupakan kumpulan data yang siap untuk diproses untuk perhitungan manual.

LANGKAH 1. MEMBUAT *ATRIBUT LIST* DAN *CLASS LIST*

Atribut list terdiri dari 2 *field* yaitu nilai atribut dan indek dari daftar kelas. *Attribut list* JK, *attribut list* UMUR, *attribut list* BB, *attribut list* TB ditunjukkan pada Tabel 3.10. Data tersebut di indeks kemudian di sorting dari nilai kecil ke besar. Data

yang diindeks adalah data yang berjenis numerikal saja. Dari data diatas maka terbentuk atribut list dan class list seperti di bawah ini :

Tabel 3. 9 Tabel Attibut list

Sex L=1,P=2	indeks	Umur (Bln)	indeks	BB (Kg)	indeks	TB (Cm)	indeks
2	1	8	13	6.5	13	60	13
2	2	9	25	6.7	3	71	12
2	3	13	14	6.8	12	71	25
1	4	15	8	7	10	72.5	8
1	5	15	31	7.1	23	73	3
1	6	16	4	7.5	5	73.5	24
1	7	16	12	7.5	8	74	4
1	8	17	3	7.6	24	74	5
2	9	18	7	7.7	6	75	10
2	10	18	24	7.8	7	75	23
2	11	19	2	7.9	19	75.5	21
1	12	19	6	8	21	76	2
1	13	19	10	9	14	76	6
1	14	19	19	9	22	76	7
1	15	19	20	9	25	76	14
2	16	19	21	9.1	4	76.5	22
2	17	20	9	10.9	28	79	9
1	18	21	5	11.6	16	80	19
2	19	21	18	11.8	1	81	1
2	20	21	23	12	2	83	27
1	21	21	27	12	29	83	29
2	22	22	1	12.2	27	83.5	31
2	23	23	29	13	15	86	28
1	24	23	30	13.8	9	88	18
1	25	26	22	14	20	88	20
1	26	26	26	15	31	90	15
1	27	30	15	17	30	95	11
2	28	31	28	19	18	95	30
2	29	48	11	19	26	96	16
1	30	57	17	24	11	96	26
2	31	59	16	29	17	112	17

Sedangkan untuk class list nya ada pada Tabel 3.11.

Tabel 3. 10 Tabel Class List

Status	leaf
G. Baik	N1
G. Baik	N1
G. Buruk	N1
G. Baik	N1
G. Buruk	N1
G. Buruk	N1
G. Kurang	N1
G. Kurang	N1
G. Lebih	N1
G. Buruk	N1
G. Lebih	N1
G. Buruk	N1
G. Kurang	N1
G. Baik	N1
G. Baik	N1
G. Buruk	N1
G. Lebih	N1
G. Lebih	N1
G. Kurang	N1
G. Lebih	N1
G. Kurang	N1
G. Kurang	N1
G. Buruk	N1
G. Buruk	N1
G. Baik	N1
G. Lebih	N1
G. Baik	N1
G. Baik	N1
G. Baik	N1
G. Lebih	N1
G. Lebih	N1

LANGKAH 2 . MEMBUAT HISTOGRAM UNTUK SETIAP ATRIBUT

Histogram adalah table yang menyimpan frekuensi dari kemunculan kelas dalam atribut tersebut. Contoh : histogram untuk atribut UMUR, dengan nilai batas 9,13,15,16,17,18,19,20,21,22,26,30,31,48,56,57. Nilai ini di pilih sendiri oleh peneliti. Caranya menenukan jumlah yang memenuhi pernyataan \leq nilai batas



[$L(\leq \text{nilai batas})$] dan $R(> \text{nilai batas})$] tersebut. begitu pula untuk histogram atribut yang lainnya, menentukan nilai batas nya terlebih dahulu. Jenis kelamin merupakan atribut kategorikal dan umur, bb dan tb merupakan atribut numerikal. Dibawah ini telah diilustrasikan histogram untuk setiap atribut. Ilustrasi histogram untuk Jenis Kelamin, umur, berat badan dan tinggi badan di tunjukkan pada Tabel 3.12, 3.13, 3.14, dan 3.15.

- Jenis kelamin

Tabel 3. 11 Tabel histogram jenis kelamin

J.K	G. Buruk	G. Kurang	G. Baik	G. Lebih
L	4	4	5	3
P	4	2	4	5

- Umur (nilai batas =9,13,15,16,17,18,19,20,21,22,26,30,31,48,56,57)

Tabel 3. 12 Tabel histogram umur

umur	G. Buruk	G. Kurang	G. Baik	G. Lebih
L(≤ 9)		1	1	
R(> 9)	8	5	8	8
L(≤ 13)		1	2	
R(> 13)	8	5	7	8
L(≤ 15)		2	2	1
R(> 15)	8	4	7	7
L(≤ 16)	1	2	3	1
R(> 16)	7	4	6	7
L(≤ 17)	2	2	3	1
R(> 17)	6	4	6	7
L(≤ 18)	3	3	3	1
R(> 18)	5	3	6	7
L(≤ 19)	5	5	4	2
R(> 19)	3	1	5	6
L(≤ 20)	5	5	4	3
R(> 20)	3	1	5	5
L(≤ 21)	7	5	5	4
R(> 21)	1	1	4	4
L(≤ 22)	7	5	6	4
R(> 22)	1	1	3	4
L(≤ 26)	7	6	5	6
R(> 26)	1		3	2
L(≤ 30)	7	6	8	6
R(> 30)	1		1	2
L(≤ 48)	7	6	9	7
R(> 48)	1			1

L(<=56)	7	6	9	7
R(>56)	1			1
L(<=57)	7	6	9	8
R(>57)	1			

- Berat Badan (nb :7;8;9;10;11;12;13;14;15;17;19;24)

Tabel 3. 13 Tabel histogram berat badan

bb	G.Buruk	G. Kurang	G. Baik	G.Lebih
L(<=7)	3	1		
R(>7)	5	5	9	8
L(<=8)	7	5		
R(>8)	1	1	9	8
L(<=9)	7	6	2	
R(>9)	1		7	8
L(<=10)	7	6	3	
R(>10)	1		6	8
L(<=11)	7	6	4	
R(>11)	1		5	8
L(<=12)	8	6	7	
R(>12)			2	8
L(<=13)	8	6	9	
R(>13)				8
L(<=14)	8	6	9	2
R(>14)				6
L(<=15)	8	6	9	3
R(>15)				5
L(<=17)	8	6	9	4
R(>17)				4
L(<=19)	8	6	9	6
R(>19)				2
L(<=24)	8	6	9	7
R(>24)				1

- Tinggi Badan (nilai batas:71;72;73;74;75;76;79;80;81;83;86;88;90;95;96; 105)

Tabel 3. 14 Tabel Histogram tinggi badan

TB	G.Buruk	G. Kurang	G. Baik	G.Lebih
L(<=71)	1	1	1	
R(>71)	7	5	8	8
L(<=72)	1	1	1	
R(>72)	7	5	8	8

L(<=73)	2	2	1	
R(>73)	6	4	8	8
L(<=74)	4	2	2	
R(>74)	4	4	7	8
L(<=75)	6	2	2	
R(>75)	2	4	7	8
L(<=76)	7	4	4	
R(>76)	1	2	5	8
L(<=79)	7	5	4	1
R(>79)	1	1	5	7
L(<=80)	7	6	4	1
R(>80)	1		5	7
L(<=81)	7	6	5	1
R(>81)	1		4	7
L(<=83)	7	6	7	1
R(>83)	1		2	7
L(<=86)	7	6	8	2
R(>86)	1		1	6
L(<=88)	7	6	8	4
R(>88)	1		1	4
L(<=90)	7	6	9	4
R(>90)	1			4
L(<=95)	7	6	9	6
R(>95)	1			2
L(<=96)	8	6	9	7
R(>96)				1
L(<=105)	8	6	9	7
R(>105)				1

LANGKAH 3 MENGHITUNG NILAI GINI INDEKS UNTUK MEMILIH ATRIBUT TERBAIK DAN SPLIT NODE

Untuk setiap atribut dihitung nilai gini indeksnya dengan menggunakan persamaan di bawah. Contoh perhitungan gini indeks untuk atribut UMUR dengan nilai batas 9 [L(<=9) & R (>9)] dijelaskan pada persamaan (3.1), (3.2), dan (3.3).

$$Gini(D_1) = 1 - \left(\left(\frac{0}{2}\right)^2 + \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 + \left(\frac{0}{2}\right)^2 \right) = 0.5 \quad (3.1)$$

$$Gini(D_2) = 1 - \left(\left(\frac{8}{29}\right)^2 + \left(\frac{5}{29}\right)^2 + \left(\frac{8}{29}\right)^2 + \left(\frac{8}{29}\right)^2 \right) = 0.742 \quad (3.2)$$

$$Gini_A(D) = \left(\frac{2}{31}\right) \cdot 0.5 + \left(\frac{30}{31}\right) \cdot 0.74 = 0.704 \quad (3.3)$$



Perhitungan gini indeks pada persamaan (3.1), (3.2), dan (3.3) didasarkan pada persamaan (2.2) dan (2.3) pada Bab 2. Perhitungan gini indeks juga dilakukan untuk semua nilai batas lainnya. Baik untuk atribut JK, BB, maupun TB. Kemudian dicari atribut dengan nilai gini terkecil. Pada Tabel 3.16 menunjukkan hasil perhitungan gini indeks untuk semua atribut.

Tabel 3. 15 Tabel hasil perhitungan gini indeks

ATRIBUT	GINI
JK	0.71276
UMUR	0.674479
BB	0.472826
TB	0.599107

Dari table di atas dapat dilihat bahwa nilai terkecil ada pada atribut BB dengan nilai gini 0.528382, sehingga atribut ini dipilih sebagai split node. Nilai gini dari atribut BB terkecil didapat dari nilai batas = 13. Selanjutnya adalah menentukan subset untuk $BB \leq 13$ dan $BB > 13$.

Table 3.17 dan Tabel 3.18 menunjukkan data *subset* pada atribut berat badan dengan nilai batas 13.

- Tabel data subset $BB \leq 13$

Tabel 3. 16 Tabel subset pada atribut Berat Badan ≤ 13

13	Nicolas A. Y.	1	8	6.5	60	G. Kurang
3	Flora Dwiyanti	2	17	6.7	73	G. Buruk
12	M.A. Satria Natanegara	1	16	6.8	71	G. Buruk
10	Kaila Isfiona	2	19	7	75	G. Buruk
23	Amalia Anisah	2	21	7.1	75	G. Buruk
5	Yuda Pratama	1	21	7.5	74	G. Buruk
8	M. Okta Fadhil	1	15	7.5	72.5	G. Kurang
24	Mahardika Putra Ramadani	1	18	7.6	73.5	G. Buruk
6	Yeremia Yulius S K	1	19	7.7	76	G. Buruk
7	Ibrahim Islami Pasei	1	18	7.8	76	G. Kurang
19	Siti Masrurroh	2	19	7.9	80	G. Kurang
21	Dimas Adrianto	1	19	8	75.5	G. Kurang
14	Gilang Putra R.	1	13	9	76	G. Baik
22	Fitriana	2	26	9	76.5	G. Kurang
25	Moh. Nafal A.	1	9	9	71	G. Baik
4	M. Shohibul	1	16	9.1	74	G. Baik
28	Imeida Reva P.	2	31	10.9	86	G. Baik
16	Nadila Syafa	2	59	11.6	96	G. Buruk
1	Gusna Maratus	2	22	11.8	81	G. Baik
2	Sifaatul Khasanah	2	19	12	76	G. Baik
29	Arsa Aditya P.	2	23	12	83	G. Baik

27	Nurfarhan	1	21	12.2	83	G. Baik
15	Fajar Ismu K.	1	30	13	90	G. Baik

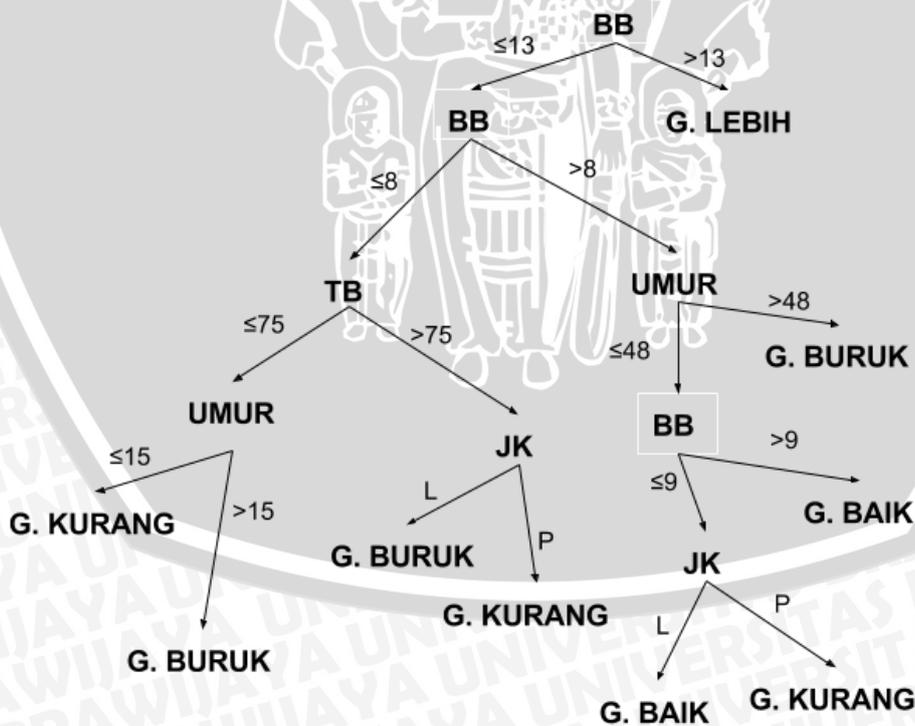
Sedangkan pada Tabel 3.18 di bawah ini merupakan subset untuk atribut berat badan yang lebih dari 13.

Tabel 3. 17 Tabel subset pada atribut Berat Badan > 13

9	Aura Nabila	2	20	13.8	79	G. Lebih
20	Susi Saridewi	2	19	14	88	G. Lebih
31	Vanesa Meiga W.	2	15	15	83.5	G. Lebih
30	Aizin Azna	1	23	17	95	G. Lebih
18	Yuda Pratama	1	21	19	88	G. Lebih
26	Dewa Putra N.	1	26	19	96	G. Lebih
11	Eka Nurcyntaningrum	2	48	24	95	G. Lebih
17	Alea	2	57	29	112	G. Lebih

LANGKAH 4. PEMBENTUKAN TREE

Dari table di atas menunjukkan bahwa subset belum berada dalam satu kelas, sehingga dilakukan lagi langkah 1-4 untuk menentukan *split*-nya. Setelah melakukan pengulangan langkah 1-4 didapatkan hasilnya ditunjukkan pada Gambar 3.20.



Gambar 3. 20 Hasil Decision Tree

LANGKAH 5. PEMBENTUKAN *DECISION RULE*

Decision rule terbentuk jika *decision tree* sudah ada. Untuk melihat aturan yang akan diterapkan mengacu pada *node*, *leaf* serta *root* dari tersebut. Sehingga terbentuklah *rule* seperti di bawah ini :

1. **IF BB >13 THEN GIZI LEBIH**
2. **IF BB <=13 AND BB >8 AND UMUR >48 THEN GIZI BURUK**
3. **IF BB <=13 AND BB >8 AND UMUR <=48 AND BB >9 THEN GIZI BAIK**
4. **IF BB <=13 AND BB >8 AND UMUR <=48 AND BB <=9 AND JK = L THEN GIZI BAIK**
5. **IF BB <=13 AND BB >8 AND UMUR <=48 AND BB <=9 AND JK = P THEN GIZI KURANG**
6. **IF BB <=13 AND BB <=8 AND TB >75 AND JK = L THEN GIZI BURUK**
7. **IF BB <=13 AND BB <=8 AND TB >75 AND JK =P THEN GIZI KURANG**
8. **IF BB <=13 AND BB <=8 AND TB <=75 AND UMUR >15 THEN GIZI BURUK**
9. **IF BB <=13 AND BB <=8 AND TB <=75 AND UMUR <= 15 THEN GIZI KURANG**



BAB 4 IMLEMENTASI

Bab ini berisikan tentang hasil implementasi metode *Supervised Learning In Quest* (SLIQ) ke dalam bentuk program serta penjelasan tiap fungsi halaman pada program.

4.1. Lingkungan Implementasi

Lingkungan implementasi yang digunakan untuk penelitian ini adalah lingkungan perangkat lunak dan lingkungan perangkat keras.

4.1.1. Lingkungan *Hardware*

Perangkat keras (*hardware*) yang digunakan untuk mplementasi penelitian ini adalah :

- Jenis : Notebook
- Prosesor : Intel Pentium @ 1.80 Ghz 64 bit
- *Memory* :4 GB
- *Hardisk* :500 GB

4.1.2. Lingkungan *Software*

Perangkat lunak yang digunakan untuk implementasi penelitian ini adalah :

- Sistem Operasi Windows 8
- Editor Java (Netbean 8.0.2)
- JDK 8 update 45 64bit
- XAMPP v3.2.1
- DBMS MySQL
- *Tool* DBMS phpmyAdmin 4.4.12

4.2. Persiapan Data

Data yang digunakan untuk penelitian ini berupa data latih dan data uji. Rincian isi data berupa umur, jenis kelamin, berat badan, tinggi badan, status gizi bayi di beberapa desa di Kecamatan Kertosono Kabupaten Nganjuk. Untuk data latih berupa data bayi berstatus gizi berjumlah 504 dengan rincian yang telah disebutkan di Bab 3. Data ini di gunakan sebagai data latih dan data uji untuk aplikasi ini.

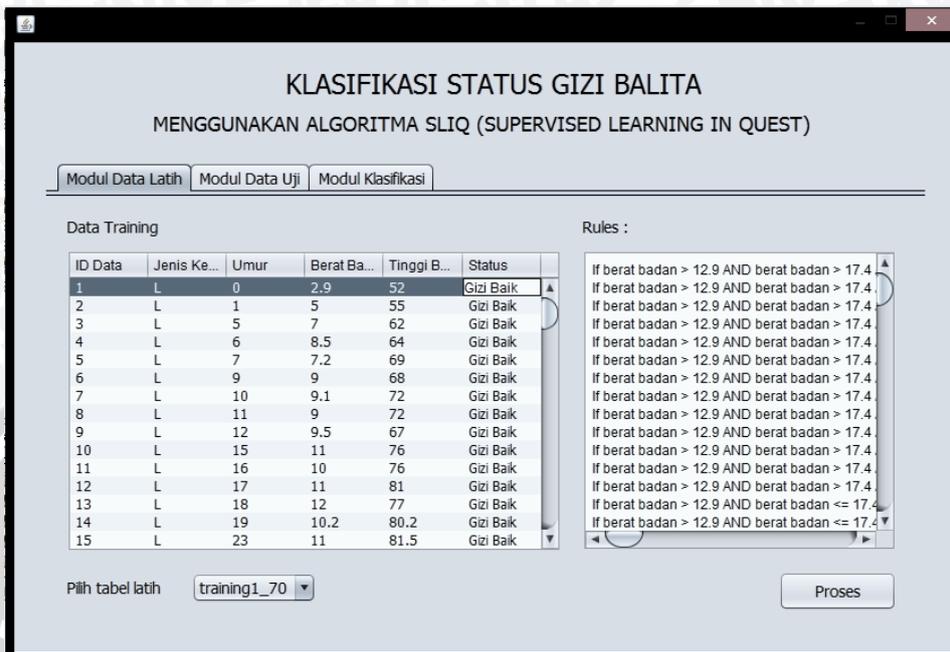
Metode penyimpana data dilakukan dengan mengumpulkan data bayi tersebut ke dalam sebuah basisdata yang di simpan dalam tabel-tabel. Masing-masing tabel berisi data bayi yang di simbolkan dengan id sesuai dengan atribut klasifikasinya yang telah disebutkan dirincian pada Bab 5.

4.3. Implementasi Antarmuka

4.3.1. Modul Data Latih

Pada Gambar 4.1 merupakan tampilan antarmuka untuk modul proses data latih. PAda modul ini pengguna diarahkan untuk memilih tabel mana yang akan dijadikan sebagai kumpulan data latih, kemudian, setelah

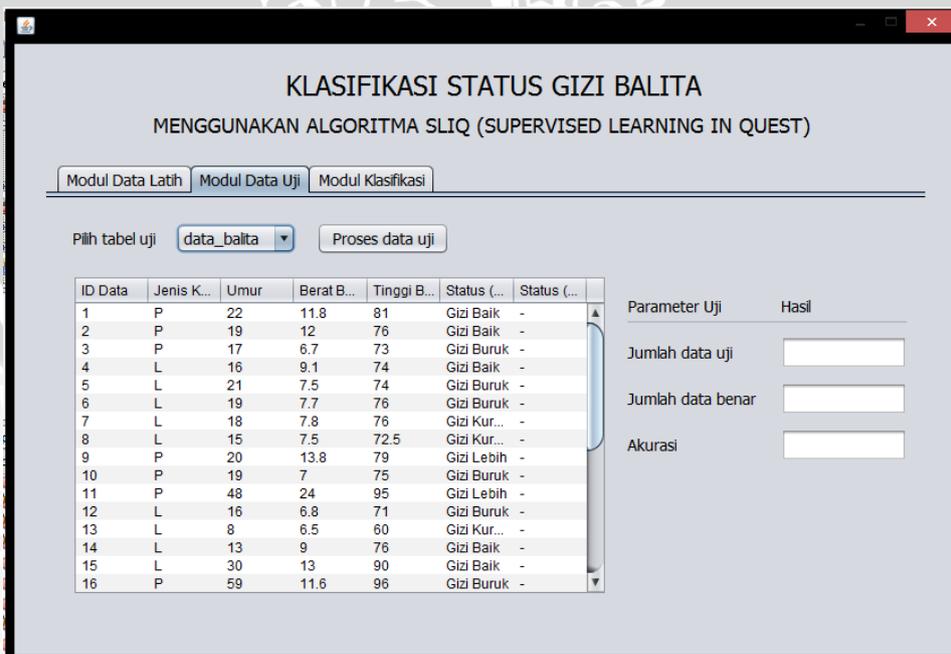
memilih tabel, selanjutnya dilakukan proses pembentukan *rule* dengan tombol proses.



Gambar 4. 1 Modul Data Latih

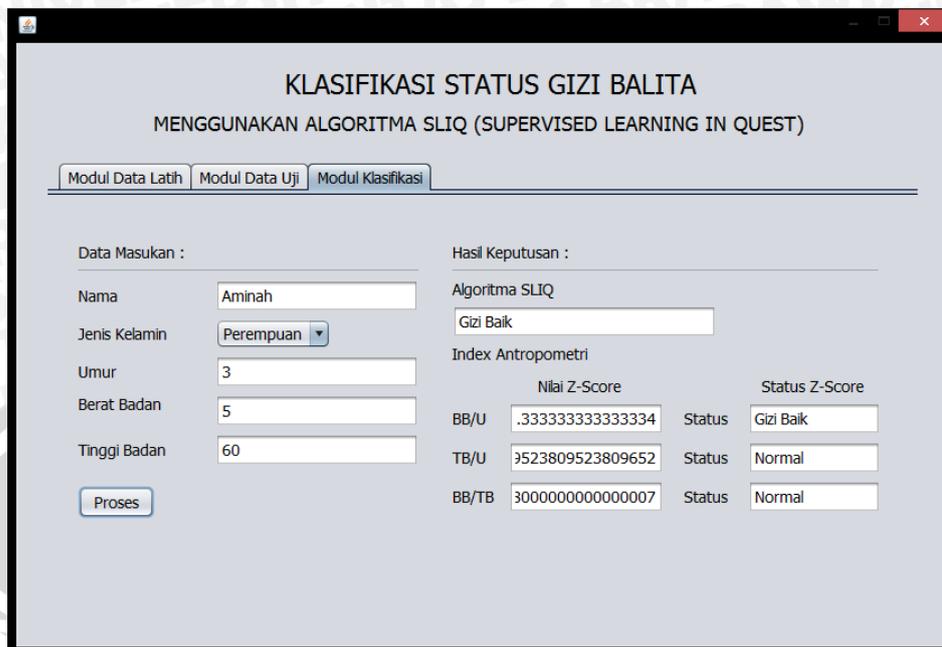
4.3.2. Modul Data Uji

Modul data uji disini berguna untuk menguji datayang telah dipilih, setelah dilakukannya *training* pada modul data latih. Untuk tampilan modul data latih dapat dilihat pada Gambar 4.2.



Gambar 4. 2 Modul Data Uji

4.3.3. Modul Klasifikasi



Gambar 4. 3 Modul Klasifikasi

Selanjutnya, modul klasifikasi ditunjukkan pada Gambar 4.3. Modul ini berguna untuk melihat klasifikasi data per individu, dengan memasukkan data yang diminta, yaitu nama, jenis kelamin, umur, berat badan dan tinggi badan. Kemudian pengguna memilih tombol proses untuk memproses hasil klasifikasi, hasil klasifikasinya dapat dilihat pada box di sebelah kanan pada Gambar 4.3. Pada aplikasi ini juga diengkapi hasil perhitungan z-score untuk TB/U dan BB/TB

4.4. Implementasi Program

4.4.1. Program Proses Algoritma SLIQ

Proses penerapan algoritma SLIQ secara rancangan dapat dilihat pada Gambar 3.4 diagram alir untuk algoritma SLIQ. Selanjutnya dijelaskan untuk penerapan nya di bagi kedalam beberapa fungsi pada pemrogramannya. Diantaranya adalah mengindeks data, penentuan nilai batas, pementukan histogram, menghitung gini indeks, *sorting* gini indeks, menentukan split, dan pembentukan *decision tree* dan *decision rule*.

- Mengindeks data

```

121 public double [][] sorting(double index[],double att[],int
    jml_data) {
122     double hasil [][] = new double [2][jml_data];
123     double temp_index, temp_att;
124     for(int i=1; i<=jml_data; i++){
125         for(int j=0; j<jml_data-i; j++){
126             if(att[j]>att[j+1]){
127                 temp_index = index[j];

```

```

128         temp_att = att[j];
129         index[j] = index[j+1];
130         att[j] = att[j+1];
131         index[j+1] = temp_index;
132         att[j+1] = temp_att;
133     }
134 }
135 }
136 for(int i=0; i<jml_data; i++){
137     hasil[0][i]=index[i];
138     hasil[1][i]=att[i];
139 }
140 return hasil;
141 }

```

Kode Sumber 4. 1 Fungsi Sorting untuki Indeks data

Langkah pertama dalam algoritma ini adalah memberi indeks pada setiap data yang akan di olah pada masing-masing atribut. Pada Kode Sumber 4.1 menunjukkan bagaimana implementasi indeks data dalam sistem yang disimpan melalui beberapa *array*.

- **Penentuan nilai batas**

Selanjutnya adalah menentukan nilai batas untuk setiap attribut. Atribut yang digunakan ada empat yang diimplementasikan berdasarkan urutan nomor dengan status nilai *double*. Nilai batas ini dapat dilihat pada Kode Sumber 4.2.

```

144 public double[] proses(int i, double index[], double att[],
145 String status[],int jml_data){
146     double hasil [] = new double [3];
147     double hasil_sorting[][] = new double [2][jml_data];
148     double mingni [] = new double [2];
149     if(i==0){
150         hasil[0]
151         hitung_gni(1.5,index,att,status,jml_data);
152         hasil[1] = 1.5;
153         hasil[2] = 0;
154         // System.out.println("jk "+hasil[0]);
155     }
156     else if(i==1){
157         double batas[]
158         {1,3,6,10,12,15,18,21,24,30,36,42,48,54,58};
159         double hasil_gni [] = new double [batas.length];

```

```
158         for(int j=0; j<batas.length; j++){
159     hasil_gni[j]=hitung_gni(batas[j],index,att,status,jml_data);
160     //         System.out.println("umur "+hasil_gni[j]+"
161     "+batas[j]);
162     }
163     mingni = min_gni(batas.length, hasil_gni, batas);
164     hasil[0] = mingni[0];
165     hasil[1] = mingni[1];
166     hasil[2] = i;
167     }
168     else if(i==2){
169         double         batas[] =
170         {2.6,2.9,3.7,3.9,4,4.3,4.55,5,5.9,6.4,
171         7,7.4,7.6,8.35,9.2,9.7,9.9,10.3,10.5,10.7,11.7,13,14.1,
172         15.1,16.2,17.2,17.9};
173         double hasil_gni [] = new double [batas.length];
174         for(int j=0; j<batas.length; j++){
175     hasil_gni[j]=hitung_gni(batas[j],index,att,status,jml_data);
176     //         System.out.println("bb "+hasil_gni[j]+"
177     "+batas[j]);
178     }
179     mingni = min_gni(batas.length, hasil_gni, batas);
180     hasil[0] = mingni[0];
181     hasil[1] = mingni[1];
182     hasil[2] = i;
183     }
184     else{
185         double         batas[] =
186         {54.2,57.5,59,66.6,67.5,70,74.8,78,80,
187         81.1,81.5,86.4,90.1,91.2,95.8,99.4,103,106.4,108.6};
188         double hasil_gni [] = new double [batas.length];
189         for(int j=0; j<batas.length; j++){
190     hasil_gni[j]=hitung_gni(batas[j],index,att,status,jml_data);
191     //         System.out.println("tb "+hasil_gni[j]+"
192     "+batas[j]);
193     }
194     mingni = min_gni(batas.length, hasil_gni, batas);
195     hasil[0] = mingni[0];
```

```

190         hasil[1] = mingni[1];
191         hasil[2] = i;
192     }
193     return hasil;
194 }

```

Kode Sumber 4. 2 Menentukan Nilai Batas

- **Pembentukan histogram dan gini indeks**

Perhitungan gini indeks nya ditunjukkan pada baris 255, 256 Kode Sumber 4.3. Fungsi hitung_gni() ini merupakan inisialisasi untuk pembentukan histogram dan perhitungan gini indeks.

```

201 public double hitung_gni(double batas, double index[], double
att[], String status[], int jml_data){
202     double hasil;
203     double g_buruk1=0,g_kurang1=0,g_baik1=0,g_lebih1=0;
204     double g_buruk2=0,g_kurang2=0,g_baik2=0,g_lebih2=0;
205     double sum1=0, sum2=0;
206     double n1, n2;
207     for(int i=0; i<jml_data; i++){
208         if(att[i]<=batas){
209             if(status[(int)index[i]-
1].equalsIgnoreCase("Gizi Buruk")) g_buruk1++;
210             else if(status[(int)index[i]-
1].equalsIgnoreCase
("Gizi Kurang")) g_kurang1++;
211             else if(status[(int)index[i]-
1].equalsIgnoreCase
("Gizi Baik")) g_baik1++;
212             else g_lebih1++;
213             sum1++;
214         }
215         else{
216             if(status[(int)index[i]-
1].equalsIgnoreCase("Gizi Buruk")) g_buruk2++;
217             else if(status[(int)index[i]-
1].equalsIgnoreCase(
"Gizi Kurang")) g_kurang2++;
218             else if(status[(int)index[i]-
1].equalsIgnoreCase
("Gizi Baik")) g_baik2++;
219             else g_lebih2++;
220             sum2++;
221         }
222     }
223     if(sum1==0||sum2==0) return 1;
224     else{

```

```

                n1          =          1          -
(Math.pow(g_buruk1/sum1,2)+Math.pow(g_kurang1/sum1,2)
225 +Math.pow(g_baik1/sum1,2)+Math.pow(g_lebih1/sum1,2));
                n2          =          1          -
(Math.pow(g_buruk2/sum2,2)+Math.pow(g_kurang2/sum2,2)
226 +Math.pow(g_baik2/sum2,2)+Math.pow(g_lebih2/sum2,2));
                hasil
227 (sum1/(sum1+sum2))*n1+(sum2/(sum1+sum2))*n2;
228         return hasil;
229     }
230 }

```

Kode Sumber 4. 3 Pembentukan histogram dan gini indeks

- **Sorting gini indeks**

Setelah nilai gini indeks nya diketahui barulah nilai tersebut di urutkan dan dipilih nilai terkecil. Penerapannya ditunjukkan pada program Kode Sumber 4.4

```

234 public double [] min_gni(int jumlah_batas, double gni [],
double batas[]){
235     double hasil [] = new double [2];
236     double temp1, temp2;
237     for(int i=1; i<=jumlah_batas; i++){
238         for(int j=0; j<jumlah_batas-i; j++){
239             if(gni [j]>gni [j+1]){
240                 temp1 = batas[j];
241                 temp2 = gni[j];
242                 batas[j] = batas[j+1];
243                 gni[j] = gni[j+1];
244                 batas[j+1] = temp1;
245                 gni[j+1] = temp2;
246             }
247         }
248     }
249     hasil[0]=gni[0];
250     hasil[1]=batas[0];
251     return hasil;
252 }

```

Kode Sumber 4. 4 Mengurutkan hasil gini indeks

- **Split**

Fungsi `sliq()` berguna untuk menentukan nilai gin iindeks terbaik dari histogram, sehingga menghasilkan *split* terbaik.

```

255 public int sliq(String data[][],int jml_data, int batas,
int atribut, String r, int node){

```

```
256     int split;
257     String    nama_atribut    []    =    {"jenis
kelamin","umur","berat badan",
258     "tinggi badan"};
259     double hasil_gni[]=new double [3];
260     double hasil_sorting[][]=new double [2][jml_data];
261     double gni [][] = new double [4][3];
262     double index[][] = new double [4][jml_data];
263     double att[][] = new double [4][jml_data];
264     String status[]= new String [jml_data];
265     for(int i=0; i<jml_data; i++){
266         index[0][i] = Double.parseDouble(data[0][i]);
267         index[1][i] = Double.parseDouble(data[0][i]);
268         index[2][i] = Double.parseDouble(data[0][i]);
269         index[3][i] = Double.parseDouble(data[0][i]);
270         att[0][i] = Double.parseDouble(data[1][i]);
271         att[1][i] = Double.parseDouble(data[2][i]);
272         att[2][i] = Double.parseDouble(data[3][i]);
273         att[3][i] = Double.parseDouble(data[4][i]);
274         status [i] = data[5][i];
275         System.out.println(data[0][i]+" "+data[1][i]+"
"+data[2][i]+"
276         "+data[3][i]+" "+data[4][i]+" "+data[5][i]);
277     }
278     for(int i=0;i<jumlah_atribut;i++){
279         //if(i!=0){
280             hasil_sorting
281             sorting(index[i],att[i],jml_data);
282             for(int j=0; j<jml_data; j++){
283                 index[i][j]=hasil_sorting[0][j];
284                 att[i][j]=hasil_sorting[1][j];
285             }
286         //}
287     }
288     hasil_gni=proses(i,index[i],att[i],status,jml_data);
289     gni[i][0]=hasil_gni[0];
290     gni[i][1]=hasil_gni[1];
291     gni[i][2]=hasil_gni[2];
292     }
293     for(int i=0; i<4; i++){
294         System.out.println(gni[i][0]+" "+gni[i][1]);
295     }
296     double temp_batas,temp_gni,temp_att;
297     for(int i=1; i<=4; i++){
298         for(int j=0; j<4-i; j++){
299             if(gni[j][0]>gni[j+1][0]){
```

```
296         temp_batas = gni[j][1];
297         temp_gni = gni[j][0];
298         temp_att = gni[j][2];
299         gni[j][1] = gni[j+1][1];
300         gni[j][0] = gni[j+1][0];
301         gni[j][2] = gni[j+1][2];
302         gni[j+1][1] = temp_batas;
303         gni[j+1][0] = temp_gni;
304         gni[j+1][2] = temp_att;
305     }
306 }
307 }
308 System.out.println(gni[0][0]+"          "+gni[0][1]+"
309 "+gni[0][2]);
310 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
311 //
312 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
313 int jmlh=0;
314 for(int i=0; i<jml_data; i++){
315     if(att[(int)gni[0][2]][i]>gni[0][1]){
316         jmlh++;
317     }
318 }
319 split=(int)gni[0][1];
320 parent.add((int)gni[0][2]);
321 parent.add(split);
322 if(node==1){
323     kiri.add((int)gni[0][2]);
324     kiri.add(split);
325 }
326 if(node==2){
327     parent1.add(atribut);
328     parent1.add(batas);
329     kanan.add((int)gni[0][2]);
330     kanan.add((int)gni[0][1]);
331 }
423 return split;
424 }
```

Kode Sumber 4. 5 Pembentukan Split

- **Pembentukan *tree***

Pembentukan *tree* diimplementasikan pada Kode Sumber 4.6. Inisialisasi ini ditunjukkan untuk data latihnya. Sehingga terbentuklah *tree* yang menjadi *rule* untuk pengujian data pada tahap selanjutnya.

```
428 public int[][] tree(){
429     //inisialisasi untuk membuat pohon keputusan
430     int aturan [][] =new int[parent.size()][6]; //
431     int p[][]=new int [parent.size()/2][2];
432     int c[][]=new int [parent.size()/2][2];
433     int x=0,y=0;
434     for (Object o: parent){
435         if(x==0){
436             aturan[y][0]=Integer.parseInt(o.toString());
437                 x++;
438         }
439         else{
440             aturan[y][1]=Integer.parseInt(o.toString());
441                 y++;
442                 x=0;
443         }
444     }
445     x=0;
446     y=0;
447     for (Object o: kiri){
448         if(x==0){
449             aturan[y][2]=Integer.parseInt(o.toString());
450                 x++;
451         }
452         else{
453             aturan[y][3]=Integer.parseInt(o.toString());
454                 y++;
455                 x=0;
456         }
457     }
458     x=0;
459     y=0;
460     for (Object o: parent1){
461         if(x==0){
462             p[y][x]=Integer.parseInt(o.toString());
463                 x++;
464         }
```

```

465         else{
466             p[y][x]=Integer.parseInt(o.toString());
467             y++;
468             x=0;
469         }
470     }
471     x=0;
472     y=0;
473     for (Object o: kanan){
474         if(x==0){
475             c[y][x]=Integer.parseInt(o.toString());
476             x++;
477         }
478         else{
479             c[y][x]=Integer.parseInt(o.toString());
480             y++;
481             x=0;
482         }
483     }
484     for(int i=0; i<(parent.size()/2); i++){
485         for(int j=0; j<(parent.size()/2); j++){
486             if(aturan[i][1]==p[j][1]){
487                 aturan[i][4]=c[j][0];
488                 aturan[i][5]=c[j][1];
489             }
490         }
491     }
492     return aturan;
493     //end
494 }

```

Kode Sumber 4. 6 Tree

- **Pembentukan rule**

Penerapan pembentukan rule-nya sendiri ditunjukkan pada baris 534 s/d 540 Kode Sumber 4.7. Namun sebelumnya dilakukan inisialisasi untuk data latih yang akan di tentukan rule nya, proses ini diperlukan untuk mengetahui atribut mana saja yang digunakan. Inisialisasi data latih diimplementasikan pada baris ke- 501 s/d 530. Keluaran *rule* ini ditampilkan pada Modul Data Training (lihat gambar 4.2).

```

498     private void
499     jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
500         // TODO add your handling code here:
501         //inisialisasi data training
502         String data [][] = new String [6][jumlah_data];
503         int z=0;

```

```
503     for (Object o: array_index){
504         data[0][z] = o.toString();
505         z++;
506     }
507     z=0;
508     for (Object o: array_jk){
509         data[1][z] = o.toString();
510         z++;
511     }
512     z=0;
513     for (Object o: array_umur){
514         data[2][z] = o.toString();
515         z++;
516     }
517     z=0;
518     for (Object o: array_bb){
519         data[3][z] = o.toString();
520         z++;
521     }
522     z=0;
523     for (Object o: array_tb){
524         data[4][z] = o.toString();
525         z++;
526     }
527     z=0;
528     for (Object o: array_status){
529         data[5][z] = o.toString();
530         z++;
531     }
532     //end
533     //proses pembuatan rule
534     String r="";
535     sliq(data,jumlah_data,4,4,r,0);
536     String penjelasan="";
537     for (Object o: rule){
538         penjelasan+="If "+o.toString()+"\n";
539     }
540     output_rule.setText(penjelasan);
541     //end
542 }
```

Kode Sumber 4. 7 Pembentukan rule

4.4.2. Program perhitungan z-score pada modul klasifikasi

Program *z-score* ini di buat untuk membandingkan klasifikasi yang dihasilkan oleh algoritma SLIQ. Klasifikasi yang dibandingkan adalah hasil klasifikasi SLIQ dengan BB/U pada kolom algoritma SLIQ dan kolom BB/U (lihat Gambar 4.3 Modul klasifikasi). Perhitungan *z-score* ini membutuhkan nilai simpang baku menurut rumusnya. Tabel simpang baku ini disimpan pada basisdata dan dipanggil melalui program baris 566 pada Kode Sumber 4.8. rumus *z-score* diimplementasikan pada program baris 592 san 594. Rumus ini sama untuk ketiga keluaran yaitu BB/U, TB/U, dan BB/TB. Namun nilai simpang baku yang diambil,serta masukan yang diambil berbeda.

Rumus ini menghasilkan keluaran nilai pada kolom keluaran BB/U, TB/U, dan BB/TB. Kemudian setelah menghasilkan nilai tersebut, barulah nilai tersebut dicek kembali untuk menentukan status gizinya (lihat baris 598 s/d 615 Kode Sumber 4.8).

```

547 public String [] hitung_zscore(double id1,int id2, double
548 id3, int i){
549     //proses perhitungan z_score
550     String hasil[]=new String [2];
551     double z_score;
552     String tabel, kolom;
553     if(i==1){
554         tabel="bbperumur";
555         kolom="umur";
556     }
557     else if(i==2){
558         tabel="tbperumur";
559         kolom="umur";
560     }
561     else{
562         tabel="bbpertb";
563         kolom="tb";
564     }
565     double simpang_baku[] = new double [7];
566     try{
567         Class.forName("com.mysql.jdbc.Driver");
568         Connection con = DriverManager.getConnection
569         ("jdbc:mysql://localhost/db_sliq","root","");
570         Statement stmt = con.createStatement();
571         ResultSet set = stmt.executeQuery("select *
572         from "+tabel+" where "+kolom+" = "+id1+" and jenis_kelamin
573         = "+id2);
574         while(set.next()){
575             Object [] a = new Object [7];
576             a[0] = set.getString("-3sd");

```

```
573         a[1] = set.getString("-2sd");
574         a[2] = set.getString("-1sd");
575         a[3] = set.getString("median");
576         a[4] = set.getString("1sd");
577         a[5] = set.getString("2sd");
578         a[6] = set.getString("3sd");

579     simpang_baku[0]=Double.parseDouble((String) a[0]);
580     simpang_baku[1]=Double.parseDouble((String) a[1]);
581     simpang_baku[2]=Double.parseDouble((String) a[2]);
582     simpang_baku[3]=Double.parseDouble((String) a[3]);
583     simpang_baku[4]=Double.parseDouble((String) a[4]);
584     simpang_baku[5]=Double.parseDouble((String) a[5]);
585     simpang_baku[6]=Double.parseDouble((String) a[6]);
586     }
587     set.close();
588 }catch(Exception a){
589     System.err.println(a);
590 }
591 if(id3>simpang_baku[3]){
592     z_score=(id3-
simpang_baku[3])/(simpang_baku[4]-simpang_baku[3]);
593 }
594     else z_score=(id3-
simpang_baku[3])/(simpang_baku[3]-simpang_baku[2]);
595     hasil[0]=Double.toString(z_score);
596     //end
597     //cek status
598     if(i==1){
599         if(z_score<-3) hasil[1]="Gizi Buruk";
600         else if(-3<z_score && z_score<-2)
hasil[1]="Gizi Kurang";
601         else if(-2<z_score && z_score<2) hasil[1]="Gizi
Baik";
602         else hasil[1]="Gizi Lebih";
603     }
604     else if(i==2){
605         if(z_score<-3) hasil[1]="Sangat Pendek";
```

```

606     else     if(-3<z_score     &&     z_score<-2)
        hasil[1]="Pendek";
        else     if(-2<z_score     &&     z_score<2)
607     hasil[1]="Normal";
608     else hasil[1]="Tinggi";
609     }
610     else{
611         if(z_score<-3) hasil[1]="Sangat Kurus";
        else     if(-3<z_score     &&     z_score<-2)
612     hasil[1]="Kurus";
        else     if(-2<z_score     &&     z_score<2)
613     hasil[1]="Normal";
614         else hasil[1]="Gemuk";
615     }
616     return hasil;
617 }

```

Kode Sumber 4. 8 Z-Score

4.4.3. Program *Testing* Pada Modul Klasifikasi

Untuk menentukan sebuah data termasuk pada klasifikasi mana, diperlukan aturan-aturan untuk menyelesaikan kasus tersebut. Untuk mengklasifikasikan data, penulis menetapkan aturan yang di tunjukkan pada Kode Sumber 4.9. Pada baris 621 menunjukkan perulangan untuk beberapa aturan yang dibutuhkan. Baris ke-625 sampai dengan 644 fungsi untuk menentukan hasil klasifikasi dari data.

Selanjutnya untuk pengujian data individu dijabarkan melalui program yang ditunjukkan Kode Sumber 4.10. baris ke-650 s/d 661 menunjukkan pengambilan data dari kolom-kolom masukan, kemudian data tersebut disimpan pada bentuk *array*. Baris 662 menunjukkan bahwa klasifikasi ini diambil dari *tree* yang sudah dihasilkan oleh data latih dengan memanggil fungsi *tree()*. Setelah menerapkan *tree* yang telah terbentuk dari data masukan, barulah proses penentuan klasifikasi dilakukan. Proses ini ditunjukkan pada program baris ke-665 sampai selesai.

```

620     public void testing(int aturan[][], int p){
621         int i = 0;
622         for(int j=0; j<(parent.size()/2); j++){
623             if(p==aturan[j][1]) i=j;
624         }
625         if(data[(int)aturan[i][0]]>aturan[i][1]){
626             if(aturan[i][3]==0){
627                 if(aturan[i][2]==1)     hasil_testing="Gizi
        Buruk";
        else     if(aturan[i][2]==2)
628     hasil_testing="Gizi Kurang";

```

```

629         else if (aturan[i][2]==3)
        hasil_testing="Gizi Baik";
630         else hasil_testing="Gizi Lebih";
631     }
632     else{
633         testing(aturan,aturan[i][3]);
634     }
635 }
636 else{
637     if (aturan[i][5]==0){
638         if (aturan[i][4]==1) hasil_testing="Gizi
        Buruk";
639         else if (aturan[i][4]==2)
        hasil_testing="Gizi Kurang";
640         else if (aturan[i][4]==3)
        hasil_testing="Gizi Baik";
641         else hasil_testing="Gizi Lebih";
642     }
643     else{
644         testing(aturan,aturan[i][5]);
645     }
646 }
647 }

```

Kode Sumber 4. 9 Aturan data uji

```

650 String nama = input_nama.getText();
651     int jenis_kelamin;
652     if (input_jk.getSelectedItem()== "Laki - Laki")
        jenis_kelamin = 1;
653     else jenis_kelamin = 2;
654     double umur =
        Double.parseDouble(input_umur.getText());
655     double bb =
        Double.parseDouble(input_bb.getText());
656     double tb =
        Double.parseDouble(input_tb.getText());
657     System.out.println(nama+"
        "+jenis_kelamin+"
        "+umur+" "+bb+" "+tb);
658     data[0]=jenis_kelamin;
659     data[1]=umur;
660     data[2]=bb;
661     data[3]=tb;
662     //end
663     int aturan [][] =new int[parent.size()][6];
664     aturan = tree();
665     //proses membuat keputusan dari data input
666     String bb_umur[]=new String [2];

```

```
667     String tb_umur[]=new String [2];
668     String bb_tb[]=new String [2];
669     testing(aturan, aturan[0][1]);
670     bb_umur=hitung_zscore(data[1], (int)data[0],data[2],1);
671     tb_umur=hitung_zscore(data[1], (int)data[0],data[3],2);
672     bb_tb=hitung_zscore(data[3], (int)data[0],data[2],3);
673     sliq.setText(hasil_testing);
674     bbumur.setText(bb_umur[0]);
675     tbumur.setText(tb_umur[0]);
676     bbtb.setText(bb_tb[0]);
677     status_bbumur.setText(bb_umur[1]);
678     status_tbumur.setText(tb_umur[1]);
679     status_bbtb.setText(bb_tb[1]);
680     //end
681 }
```

Kode Sumber 4. 10 Proses penentuan keputusan data uji



BAB 5 PENGUJIAN DAN PEMBAHASAN

Bab 5 ini membahas tahapan pengujian dan analisis sistem penentuan klasifikasi status gizi balita yang telah diimplementasikan pada bab sebelumnya.

5.1. Sisitematika Pengujian

Pada laporan ini, terdapat tiga macam pengujian yang dilakukan. Pengujian yang pertama dilakukan dengan pengujian data latih dan data uji sama. Data yang di latih sama dengan data yang diujikan. Pengujian yang kedua adalah pengujian pengaruh perubahan data latih dan data uji untuk mengetahui tingkat akurasi yang dihasilkan oleh sistem. Pengujian ketiga dilakukan dengan perubahan nilai batas pada metode SLIQ. Nilai batas yang ditentukan pada perhitungan menggunakan metode SLIQ berpengaruh besar atau tidak dilakukanlah pengujian ini. Dari ketiga pengujian tersebut yang menjadi acuan bahwa data tersebut valid atau tidak diambil dari data dengan perhitungan z-score yang telah didapatkan dari instansi terkait.

5.1.1. Sistematika Pengujian Dengan Data Latih dan Uji Sama

Pengujian ini menggunakan data pada modul data latih dan modul data uji yang sama. Pengujian ini dilakukan untuk mengetahui tingkat kesamaan antara latih dan uji akan menghasilkan prosentase akurasi yang tinggi atau tidak. Sistematika pengujiannya adalah sebagai berikut :

- Data 1 : 350 data latih dan uji
- Data 2 : 120 data latih dan uji
- Data 3 : 31 data latih dan uji

Data yang dijadikan data latih dan uji sama, diambil dari data yang berjumlah 504 data balita dengan status gizi sebagai perbandingan benar tidaknya klasifikasi. Dari data 504 tersebut dibagi menjadi tiga bagian. Dari ketiga bagian tersebut tidak dibagi secara konstan, namun diambil secara acak dan jumlah data yang berbeda dikarenakan untuk mengetahui bagaimana hasil akurasi jika data yang dipakai tersebut sangat sedikit dan banyak. Seperti ditunjukkan pada Perbandingan 1 dan Perbandingan 3, jumlah data yang digunakan berbanding terbalik.

5.1.2. Sistematika Pengujian Pegaruh Perubahan Data Latih dan Data Uji

5.1.2.1. Data Latih dan Data Uji Beda

Pengujian yang kedua adalah menguji pengaruh terhadap perubahan data latih dan data uji untuk mengetahui akurasi dari metode SLIQ. Pengujian perubahan data latih dan data uji dilakukan sebanyak 3 kali perubahan jumlah data *training* dan data *testing* dari 504 data, yaitu :

- Perbandingan 1 : 70% data latih dan 30% data uji
- Perbandingan 2 : 80% data latih dan 20% data uji
- Perbandingan 3 : 60% data latih dan 40% data uji

Data yang dijadikan sebagai data uji diambil secara acak. Jumlah keseluruhan data sebanyak 504 data balita. Pada perbandingan 60% - 40% data

latih yang digunakan sebanyak 303 data dan data uji yang digunakan sebanyak 201 data. Pada perbandingan 80% - 20% data latih yang digunakan sebanyak 403 data dan data uji yang digunakan sebanyak 101 data. Dan pada perbandingan 70% - 30% data latih yang digunakan sebanyak 353 data dan data uji yang digunakan sebanyak 151 data.

5.1.2.2. Data Latih Beda dengan Data Uji Sama

Pengujian ini dilakukan dengan menggunakan data latih yang berbeda untuk 3 pengujian, namun menggunakan data uji yang sama untuk ketiga pengujian. Pengujian yang dilakukan adalah :

- Perbandingan 1 : 70% data latih dan 151 data uji
- Perbandingan 2 : 80% data latih dan 151 data uji
- Perbandingan 3 : 60% data latih dan 151 data uji

5.1.3. Sistematika Pengujian Perubahan Nilai Batas

Nilai batas yang digunakan untuk pengujian ada 3 yang dapat dilihat pada Tabel 5.1. Perubahan nilai batas ini dilakukan untuk mengetahui tingkat akurasi sistem. Jadi setelah merubah nilai batas untuk pengujian, kemudian juga dilakukan pengujian terhadap perubahan data. Data yang diuji dibagi menjadi 6 data dengan jumlah yang berbeda, sedangkan data latih sebagai pembentuk rule sengaja ditentukan agar hasil klasifikasinya tepat. Data latihnya ditunjukkan pada Tabel 5.2. Namun di sini penulis memberikan pengecualian untuk nilai batas ke-4. Nilai batas ke empat menggunakan data latih yang di tunjukkan pada Lampiran B. Penentuan nilai batas dan data latih ini dimungkinkan akan memiliki akurasi yang lebih baik dibandingkan dengan data atih yang lain.

Pengambilan nilai batas 1 nilai batasnya diambil berdasarkan nilai normal berat badan balita dilihat dari kombinasi umur, berat badan dan tinggi badan dari ketetapan pemerintah. Sedangkan untuk nilai batas 2 didasarkan pada nilai tengah simpang baku untuk perhitungan *z-score*. Dan untuk nilai batas 3 didasarkan pada nilai yang sering muncul pada data latih dan data uji. NB pada tabel adalah Nilai Batas.

Tabel 5. 1 Tabel nilai batas untuk pengujian

NB1	JK	L,P
	Umur	2,4,6,8,10,12,15,18,21,24,27,30,33,36,39,42,45,48,54,57,59
	BB	4.3,5.7,6.9,8,8.9,9.6,9.9,10.6,11.3,11.9,12.4,12.9,13.5,14,14.5, 15,15.5,16,16.5,17.4
	TB	55,60,64.5,67.5,70.5,73.5,74.5,78,81.5,84.5,87,89.5, 92,94,96,98,99.5,101.5,103.5,107
NB2	JK	L,P
	Umur	1,3,6,10,12,15,18,21,24,30,36,42,48,54,58
	BB	2.6,2.9,3.7,3.9,4,4.3,4.5,5,5.9,6,4,7,7.4,7.6,8.35,9.2,9.7,9.9,10.3, 10.5,10.7, 11.7,13,14.1,15.1,16.2,17.2,17.9
	TB	54.2,57.5,59,66.6,67.5,70,74.8,78,80,81.1,81.5,86.4, 90.1,91.2,95.8,99.4,103,106.4,108.6
NB3	JK	L,P

	UMUR	3,6,10,12,15,18,21,24,30,36,42,48,54,58
	BB	5,6,7,8,9,10,11,12,13,15,17,19,22,24,25
	TB	58,64,69,73,74,75,78,81,83,86,88,90,92,95,97,105,114
NB4	JK	L,P
	UMUR	1,3,6,10,12,15,18,21,24,27,30,35,36,38,42,46,48,54,56,58
	BB	4.55,5,5.8,6,6.3,6.8,7,7.4,7.6,8,9,9.2,9.6,10,11,12,13,15,17,19,22,24,25
	TB	47.5,50,52,53.5,54.5,57.5,58,64,65.8,69,72,73,74,75,78,80,81,83,84,85,86,88,89,90,92,93.5,94.5,97,105,114

Tabel 5. 2 Data latihan pembentuk rule

Indeks	JK	Umur	BB	TB	Status
1	P	22	11.8	81	Gizi Baik
2	P	19	12	76	Gizi Baik
3	P	17	6.7	73	Gizi Buruk
4	L	16	9.1	74	Gizi Baik
5	L	21	7.5	74	Gizi Buruk
6	L	19	7.7	76	Gizi Buruk
7	L	18	7.8	76	Gizi Kurang
8	L	15	7.5	72.5	Gizi Kurang
9	P	20	13.8	79	Gizi Lebih
10	P	19	7	75	Gizi Buruk
11	P	48	24	95	Gizi Lebih
12	L	16	6.8	71	Gizi Buruk
13	L	8	6.5	60	Gizi Kurang
14	L	13	9	76	Gizi Baik
15	L	30	13	90	Gizi Baik
16	P	59	11.6	96	Gizi Buruk
17	P	57	29	112	Gizi Lebih
18	L	21	19	88	Gizi Lebih
19	P	19	7.9	80	Gizi Kurang
20	P	19	14	88	Gizi Lebih
21	L	19	8	75.5	Gizi Buruk
22	P	26	9	76.5	Gizi Kurang
23	P	21	7.1	75	Gizi Buruk

24	L	18	7.6	73.5	Gizi Buruk
25	L	9	9	71	Gizi Baik
26	L	26	19	96	Gizi Lebih
27	L	21	12.2	83	Gizi Baik
28	P	31	10.9	86	Gizi Baik
29	P	23	12	83	Gizi Baik
30	L	23	17	95	Gizi Lebih
31	P	15	15	83.5	Gizi Lebih

5.2. Hasil Pengujian

5.2.1. Hasil Pengujian Dengan Data Latih dan Data Uji Sama

Berdasarkan pengujian dengan data latih dan data uji yang sama, didapatkan hasil berupa tingkat akurasi sistem yang ditunjukkan pada Tabel 5.3.

Tabel 5. 3 Hasil uji dengan data latih dan uji sama

Jenis pengujian	Akurasi Data Uji (%)
Perbandingan 1 (350 data uji dan latih)	83.14 %
Perbandingan 2 (120 data uji dan latih)	100 %
Perbandingan 3 (31 data uji dan latih)	100 %
Rata-rata	94.38 %

5.2.2. Hasil Uji Pengaruh Perubahan Data Latih dan Data Uji

5.2.2.1. Hasil Pengujian Data Latih dan Data Uji Beda

Pada pengujian pengaruh perubahan data latih dan data uji digunakan perbandingan 90% data latih dan 10% data uji, 80% data latih dan 20% data uji dan 70% data latih dan 30% data uji. Hasil pengujian dapat dilihat pada Tabel 5.4.

Tabel 5. 4 Hasil pengujian akurasi perubahan jumlah data

Jenis pengujian	Akurasi Data Uji (%)
Perbandingan 1 (70% dan 30%)	66.88741 %
Perbandingan 2 (80% dan 20%)	68.31682 %
Perbandingan 3 (60% dan 40%)	72.63682 %
Rata-rata	69.28 %



5.2.2.2. Hasil Pengujian Data Latih Beda dan Data Uji Sama

Pengujian ini dilakukan untuk mengetahui hasil akurasi sistem dilihat dari data latih yang digunakan sebagai pembentuk *rule* diujikan dengan data uji yang sama. Tabel 5.5 menunjukkan hasil dari pengujian ini.

Tabel 5. 5 Hasil pengujian data uji sama

Jenis pengujian	Akurasi Data Uji (%)
Pengujian 1 (70% dan 151 data)	75.62189 %
Pengujian 2 (80% dan 151 data)	73.6318 %
Pengujian 3 (60% dan 151 data)	72.63682 %
Rata-rata	73.97 %

5.2.3. Hasil Uji Perubahan Nilai Batas

Pengujian pengaruh perubahan nilai batas hasilnya ditunjukkan pada subbab ini. Nilai batas yang digunakan dapat dilihat pada Tabel 5.1. Sedangkan untuk hasil pengujiannya dapat dilihat pada Tabel 5.6, 5.7, 5.8 dan 5.9. Data yang digunakan sebagai data latih untuk pembentukan rule adalah 31 data balita yang ditunjukkan pada Tabel 5.2. Penggunaan data ini sebagai pembentuk rule dikarenakan, data ini memiliki kecocokan antara ketiga nilai batas.

Tabel 5. 6 Hasil pengujian nilai batas 1

Nilai Batas	Jenis pengujian (data)	Akurasi Data Uji (%)
NB1	Pengujian 1 (151)	52.317883 %
	Pengujian 2 (101)	58.41584 %
	Pengujian 3 (201)	56.218903 %
	Pengujian 4 (353)	51.274788 %
	Pengujian 5 (403)	51.861042 %
	Pengujian 6 (303)	52.145214 %
Rata-rata		53.71%

Tabel 5. 7 Hasil pengujian nilai batas 2

Nilai Batas	Jenis pengujian	Akurasi Data Uji (%)
NB2	Pengujian 1 (151)	54.966885 %
	Pengujian 2 (101)	60.396038 %

	Pengujian 3 (201)	55.22388 %
	Pengujian 4 (353)	53.257793 %
	Pengujian 5 (403)	54.34243 %
	Pengujian 6 (303)	55.11551 %
	Rata-rata	55.55%

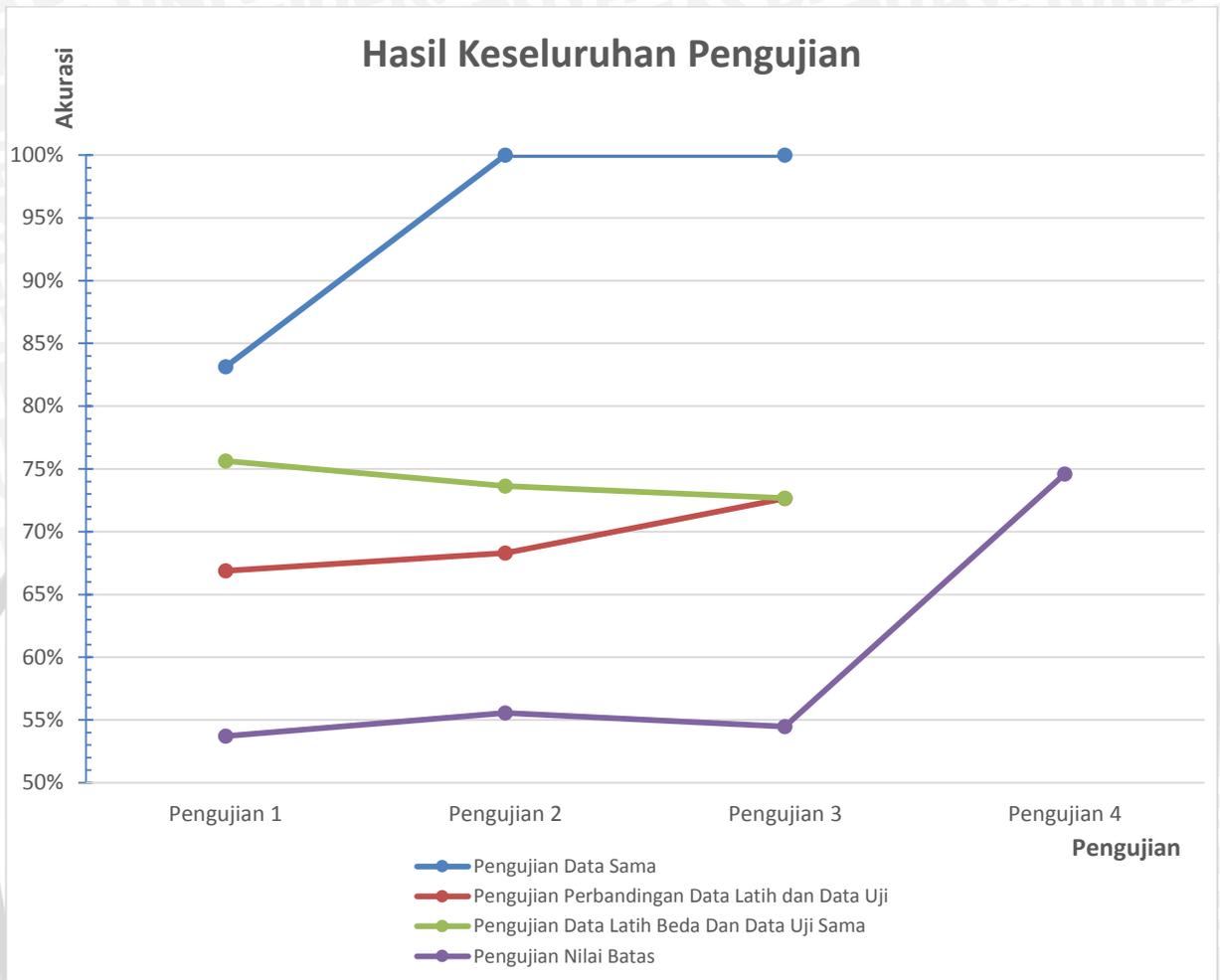
Tabel 5. 8 Hasil pengujian niai batas 3

Nilai Batas	Jenis pengujian	Akurasi Data Uji (%)
NB3	Pengujian 1 (151)	54.304634 %
	Pengujian 2 (101)	60.396038 %
	Pengujian 3 (201)	54.726368 %
	Pengujian 4 (353)	51.55807 %
	Pengujian 5 (403)	52.357323 %
	Pengujian 6 (303)	53.465347 %
	Rata-rata	54.47%

Tabel 5. 9 Hasil pengujian nilai batas 4

Nilai Batas	Jenis pengujian	Akurasi Data Uji (%)
NB4	Pengujian 1 (151)	70.86093 %
	Pengujian 2 (101)	74.25743 %
	Pengujian 3 (201)	67.16418 %
	Pengujian 4 (353)	72.804535 %
	Pengujian 5 (403)	73.697266 %
	Pengujian 6 (303)	88.77888 %
	Rata-rata	74.59 %

5.3. Analisa Hasil



Berdasarkan hasil uji coba yang telah dilakukan dari pengujian perubahan jumlah data latih dan uji, serta pengujian perubahan nilai batas yang telah ditampilkan pada Tabel 5.5 – 5.7. Pengujian akurasi yang diterapkan yaitu berupa 70% data latih dan 30% data uji, 80% data latih dan 20% data uji, serta 60% data latih dan 40% data uji. Untuk perubahan nilai batas dapat dilihat rinciannya pada Tabel 5.1.

Tabel 5.3 yang menunjukkan hasil perhitungan akurasi memperlihatkan rata-rata dari ketiga pengujian sebesar 94.38%. Dari akurasi ini dapat dilihat bahwa kesamaan klasifikasi sudah hampir mencapai nilai sempurna yaitu 100%. Nilai ini diperoleh dari perbandingan data latih dan data uji yang sama. Hasil ini membuktikan bahwa sistem sudah berjalan dengan baik untuk pengujian data sama.

Sedangkan untuk pengujian 2 yang ditunjukkan pada Tabel 5.4 menunjukkan nilai rata-rata dari ketiga pengujian adalah 69.28 %. Data yang dijadikan data latih dan data uji diambil dari data berjumlah 504. Nilai tertinggi didapat dari percobaan ketiga dengan data latih 60% dari data 504 yaitu 72.63682 %. Hal ini

dikarenakan data latih dan nilai batas dipilih yang terbaik untuk menghasilkan rule yang bagus, sehingga dapat membentuk klasifikasi yang tepat.

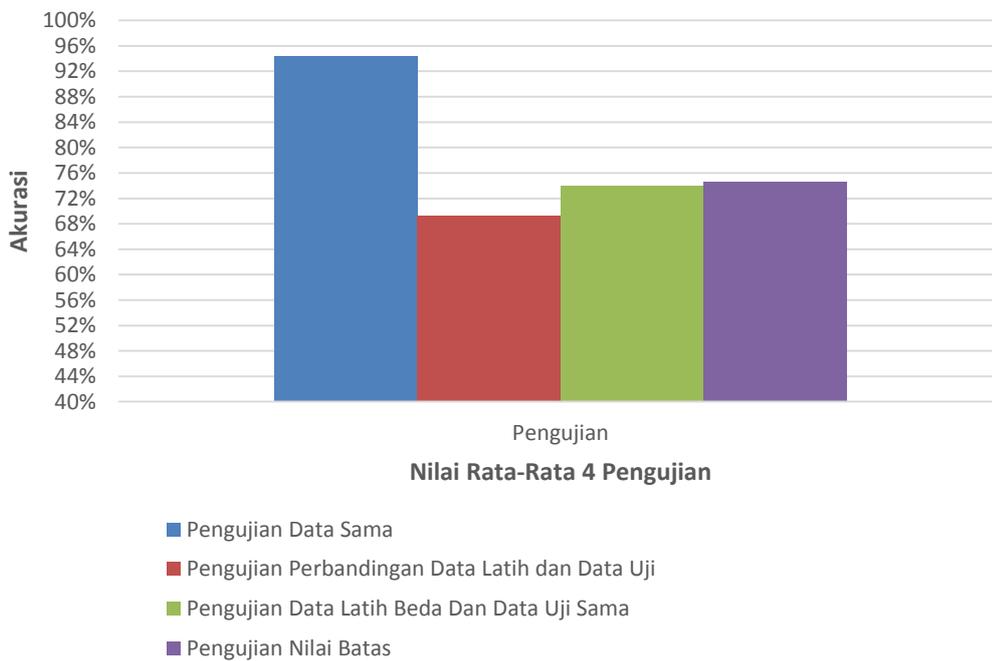
Tabel 5.5 – Tabel 5.8 menunjukkan nilai akurasi hasil pengujian perubahan nilai batas. Tabel 5.5 dengan Nilai Batas 1 mempunyai akurasi sebesar 53.71 %, Tabel 5.6 dengan nilai batas 2 rata-rata akurasi sebesar 55.55%, sedangkan untuk pengujian nilai batas 3 pada Tabel 5.7 memiliki akurasi sebesar 54.47%. Dari ketiga pengujian tersebut dapat dilihat nilai yang terbesar ada pada pengujian nilai batas 2, sedangkan yang terendah pada pengujian nilai batas 1. Perbedaan hasil akurasi tersebut disebabkan nilai batas yang ditentukan. Untuk percobaan 1 yang menggunakan nilai batas kombinasi antara umur, BB dan TB yang menjadi tetapan pemerintah malah menghasilkan nilai terendah, karena karakteristik data tidak sama dengan nilai batasnya. Sedangkan pada percobaan 2 yang menghasilkan nilai tertinggi, disebabkan nilai batas yang memiliki kesamaan karakteristik dengan data. Dimana nilai batas yang ditentukan menurut median simpang baku *z-score*. Sedangkan Tabel 5.8 menunjukkan nilai kesamaan yang paling tinggi yaitu 74.59%. hal ini disebabkan antara penentuan nilai batas dan data yang cocok serta baik dapat meningkatkan nilai akurasi dari sistem. Sehingga dapat disimpulkan bahwa nilai batas yang tepat serta data latih sebagai pembentuk rule sangat berpengaruh pada hasil akhir klasifikasi dari algoritma SLIQ.

Pengujian 1, pengujian 2, dan pengujian 3 menunjukkan nilai yang berbeda. Nilai paling tinggi ditunjukkan pada pengujian 1, sedangkan nilai akurasi terendah ditunjukkan pada pengujian 3 dengan perubahan nilai batas. Hal ini disebabkan oleh data training yang berbeda dari setiap pengujian, dilihat dari jumlah data yang dilatih, di mana data latih ini yang membentuk *rule* untuk data uji. Semakin banyak jumlah data latih, maka semakin banyak pula *rule* yang terbentuk pada data latih, sehingga untuk menentukan klasifikasi pada data uji, sistem memerlukan pertimbangan yang hampir sama untuk menentukan klasifikasinya dilihat dari rule satu dan rule lainnya.

Berdasarkan keseluruhan pengujian yang telah dilakukan, dapat disimpulkan bahwa data latih dan nilai batas sebagai pembentuk rule sangat berpengaruh terhadap hasil akhir klasifikasi dari algoritma SLIQ. Namun, pada pengujian ketiga, dengan menentukan data latih dan nilai batas sebagai pembentuk rule, dapat menghasilkan kesamaan klasifikasi hampir 75%. Hal ini disebabkan pemilihan data latih dan nilai batas yang baik untuk klasifikasi data uji. Data latih yang digunakan telah ditentukan, sehingga rule yang dihasilkan dapat mengklasifikasikan data dengan tepat.

Untuk hasil rata-rata akurasi dari keempat pengujian, ditunjukkan pada grafik Rata-rata Keempat Pengujian. Hasil tertinggi untuk pengujian dengan data yang berbeda ditunjukkan pada Pengujian 4 yaitu Pengujian nilai batas.

Rata-Rata Keempat Pengujian



BAB 6 PENUTUP

Bagian ini memuat kesimpulan dan saran terhadap skripsi ini. Kesimpulan dan saran disajikan secara terpisah, dengan penjelasan sebagai berikut:

6.1 Kesimpulan

Kesimpulan yang diperoleh dengan mengacu pada rumusan masalah, tujuan serta hasil analisis yang telah dilakukan, dapat disimpulkan bahwa :

1. Penerapan algoritma *Supervised Learning in Quest* (SLIQ) untuk pengklasifikasian status balita diterapkan dengan menggunakan empat kelas (Gizi buruk, gizi kurang, gizi baik dan gizi lebih) dan empat atribut (umur, jenis kelamin, berat badan dan tinggi badan) untuk menghasilkan klasifikasi data dari salah satu kelas (Gizi Buruk, Gizi Kurang, Gizi Baik, Gizi Lebih) diolah menggunakan algoritma SLIQ dengan proses :

- Mengindeks data
- Menentukan batas
- Membuat histogram
- Menghitung gini indeks
- Membuat split hingga terbentuk *tree*
- Merubah *decision tree* menjadi *decision rule*

2. Hasil yang diperoleh dari beberapa pengujian, akurasi tertinggi ditunjukkan pada pengujian nilai batas dengan prosentase akurasi sebesar 74.59%. Nilai ini didapat dari data latih dan nilai batas terbaik yang dipilih untuk menghasilkan *rule* yang baik, sehingga klasifikasi yang dihasilkan benar.

Penentuan data latih dan nilai batas sebagai pembentuk rule yang bagus sangat berpengaruh terhadap hasil klasifikasi dari algoritma yang diterapkan pada data uji. Oleh karena itu, harus dicermati dengan baik untuk menentukan data latih dan nilai batasnya.

6.2 Saran

Saran yang dapat diterapkan untuk penelitian sejenis selanjutnya adalah :

1. Penentuan nilai batas terhadap data latih dan data uji berpengaruh dalam pembentukan *rule*-nya, sehingga penentuan secara dinamis terhadap nilai batas dan data latih yang diambil sangat disarankan untuk penelitian selanjutnya.
2. Memperbandingkan performa algoritma SLIQ dengan algoritma klasifikasi yang lain untuk kasus seperti ini.

DAFTAR PUSTAKA

- Almatsier, S. 2005. Prinsip Dasar Ilmu Gizi. Pengenalan Ilmu Gizi. Penerbit PT Gramedia Pustaka Utama. Jakarta.
- Arisman, . 2002. Gizi Dalam Daur Kehidupan. (online) Tersedia di : <<http://klinikgizi.com/2013/12/13/cara-mencegah-kegemukan-pada-anak/>> [Diakses 10 September 2015].
- Bird, S., Klein, E., Loper, E., 2009. Natural Language Processing with Python. O'Reilly Media Inc, Sebastopol, CA 95472, United States of America.
- Departemen Kesehatan RI. 1999. Status Gizi dan Imunisasi Ibu dan Anak di Indonesia, Jakarta.
- Gibson, Rosalind S. 2005. *Principles of Nutritional Assessment*. Oxford University Press. New York.
- Han, J., Kamber, M., 2006. Data Mining : Concepts and Techniques, Second Edition. Morgan Kaufmann Publishers. San Francisco, CA 94111, United States of America.
- Jananto, A., 2010. Penggunaan Algoritma *SLIQ* untuk Pengklasifikasian Kinerja Akademik Mahasiswa. [pdf]. Universitas Stikubank. Semarang. Tersedia di : <<http://www.unisbank.ac.id/ojs/index.php/fti1/article/view/113>> [Diakses 13 Juni 2015]
- Keputusan Menteri Kesehatan RI nomor : 1995/MENKES/SK/XII/2010 tentang Standart Antropometri Penilaian Status Gizi Anak. Jakarta : Kementerian Kesehatan RI, Direktorat Jenderal Bina Gizi dan Kesehatan Ibu dan Anak.
- Kotsiantis, S, B. 2007. Supervised Machine Learning : A Review of Classification Techniques. Departement of Computer Science and Technology, University of Peloppnese. Greece.
- Kuncoro, H., 2012. Penerapan Algoritma *Sliq* Untuk Pengklasifikasian Jurusan Pada Smk Bagi Siswa Baru (Studi Kasus : Data Pendaftar Pada Psb Online

Jenang Smk Provinsi Dki Jakarta Tahun 2008-2010). Universitas Brawijaya.

Kusrini. L., Taufiq, E., 2009. Algoritma Data Mining. Penerbit C.V Andi Offset. Yogyakarta.

Larose, D., T., 2005. Discovering Knowledge in Data. New Jersey : John Willey & Sons, Inc.

Manampiring, A., E., 2008. Profil Status Gizi Balita Yang Berdomisili Di Sekitar TPA Bitung Karang Ria Kota Manado. Universitas Sam Ratulangi. Manado.

Mehta, M., Agrawal, R., dan Rissanen, J., 1996. SLIQ : A Fast Scalable Classifier for Data Mining. [pdf]. IBM Almaden Research Center. San Jose. California. Tersedia di : <www.math.unipd.it/~dulli/corso04/mehta96sliq.pdf> [Diakses 15 Mei 2015].

Muaris, Indah. 2006. Sarapan Sehat untuk Anak Balita. Penerbit PT Gramedia Pustaka Utama. Jakarta.

Nency Y., A., 2005. Gizi Buruk Ancaman Generasi yang Hilang. (online) Tersedia di <<http://io.ppijepang.org/old/article.php?id=113>> [diakses 10 Juli 2015]

Novitasari, D., 2012. Faktor- Faktor Risiko Kejadian Gizi Buruk pada Balita yang Dirawat di RSUP Dr. Kariadi Semarang.

Paath, F.E., dkk. 2004. Gizi Kesehatan Reproduksi. Penerbit Buku Kedokteran EGC. Jakarta.

Supariasa, dkk. 2002. Penilaian Status Gizi. (pdf) Jakarta : Penerbit Kedokteran EGC. Tersedia di <<http://digilib.unimus.ac.id/download.php?id=9316>> [Diakses 9 Juni 2015].

Sutabri, Tata. 2004. Pemrograman Terstruktur. Penerbit Andi. Yogyakarta

Sutjiadi, R., 2009. Supervised Learning In Quest (SLIQ) Algorithm. [online]. <<https://raymondsutjiadi.wordpress.com/2009/05/06/16/>> [Diakses tanggal 10 Juni 2015]

WHO. 2002. Program Kesehatan Anak dan Remaja 2. [online]. Tersedia di :
<<http://www.who.or.id>> [Diakses 6 Juni 2015].

Wikipedia. 2015. Nutrisi. (online) Tersedia di :
<<https://id.wikipedia.org/wiki/Nutrisi>> [Diakses 20 Juli 2015].



LAMPIRAN A DATA BALITA

INDEKS	JK	UMUR	BB	TB	STATUS GIZI
1	L	0	2.3	45	Gizi Kurang
2	L	1	3	49.2	Gizi Kurang
3	L	2	4	52.6	Gizi Kurang
4	L	3	4.6	55.8	Gizi Kurang
5	L	4	5	58	Gizi Kurang
6	L	5	5.5	60	Gizi Kurang
7	L	9	6.5	66	Gizi Kurang
8	L	10	6.8	66.8	Gizi Kurang
9	L	11	7	67	Gizi Kurang
10	L	12	7.1	69	Gizi Kurang
11	L	13	7.4	68	Gizi Kurang
12	L	15	7.5	72.2	Gizi Kurang
13	L	16	7.8	73	Gizi Kurang
14	L	17	7.9	73.5	Gizi Kurang
15	L	18	8.2	74.5	Gizi Kurang
16	L	19	8.3	75.5	Gizi Kurang
17	L	23	8.8	79	Gizi Kurang
18	L	26	9.3	80	Gizi Kurang
19	L	27	9.4	80.8	Gizi Kurang
20	L	28	9.4	82	Gizi Kurang
21	L	29	9.5	81.3	Gizi Kurang
22	L	30	9.6	82	Gizi Kurang
23	L	31	9.8	83	Gizi Kurang
24	L	32	10	83	Gizi Kurang
25	L	33	10.2	84.5	Gizi Kurang
26	L	37	10.5	86	Gizi Kurang
27	L	38	10.4	73.3	Gizi Kurang
28	L	39	10.7	86.9	Gizi Kurang
29	L	40	10.9	88.5	Gizi Kurang
30	L	41	10.7	88.3	Gizi Kurang
31	L	42	10.8	89	Gizi Kurang
32	L	43	11	89.1	Gizi Kurang
33	L	44	11.3	90	Gizi Kurang
34	L	45	11	90	Gizi Kurang
35	L	48	11.7	92	Gizi Kurang
36	L	49	12	92	Gizi Kurang
37	L	52	12.1	94	Gizi Kurang
38	L	53	12	94	Gizi Kurang
39	L	55	12	95	Gizi Kurang

40	L	56	12.3	96	Gizi Kurang
41	L	57	12.7	95	Gizi Kurang
42	L	58	12.5	96	Gizi Kurang
43	L	59	12.5	96	Gizi Kurang
44	L	1	2.8	48	Gizi Kurang
45	L	2	3.6	51.5	Gizi Kurang
46	L	3	4.1	54	Gizi Kurang
47	L	4	4.6	56	Gizi Kurang
48	L	5	5	58	Gizi Kurang
49	L	9	6	63	Gizi Kurang
50	L	10	6	65	Gizi Kurang
51	L	11	6.3	66	Gizi Kurang
52	L	12	6.5	66.5	Gizi Kurang
53	L	13	6.5	68	Gizi Kurang
54	L	14	6.8	69	Gizi Kurang
55	L	15	6.9	69.5	Gizi Kurang
56	L	16	7.2	71	Gizi Kurang
57	L	19	7.5	73	Gizi Kurang
58	L	21	7.8	75	Gizi Kurang
59	L	22	8	76	Gizi Kurang
60	L	23	8.2	77.5	Gizi Kurang
61	L	24	8.2	77	Gizi Kurang
62	L	25	8.4	77	Gizi Kurang
63	L	26	8.5	78	Gizi Kurang
64	L	29	9	81	Gizi Kurang
65	L	30	9.2	81	Gizi Kurang
66	L	31	9.1	81.2	Gizi Kurang
67	L	32	9.3	82	Gizi Kurang
68	L	33	9.5	82.5	Gizi Kurang
69	L	34	9.7	83	Gizi Kurang
70	L	35	10	84	Gizi Kurang
71	L	36	10.1	84	Gizi Kurang
72	L	37	10	85	Gizi Kurang
73	L	38	10	85	Gizi Kurang
74	L	39	10.3	86	Gizi Kurang
75	L	40	10.4	86	Gizi Kurang
76	L	41	10.5	87	Gizi Kurang
77	L	42	10.8	87.2	Gizi Kurang
78	L	46	10.9	90.1	Gizi Kurang
79	L	47	11.2	90	Gizi Kurang
80	L	48	11.4	90.5	Gizi Kurang
81	L	49	11.3	91.3	Gizi Kurang
82	L	50	11.5	91.7	Gizi Kurang

83	L	52	11.8	92	Gizi Kurang
84	L	53	11.9	93	Gizi Kurang
85	L	56	12.2	94	Gizi Kurang
86	L	59	12.6	95	Gizi Kurang
87	L	6	6	62	Gizi Kurang
88	L	7	6.1	63	Gizi Kurang
89	L	8	6.6	64.5	Gizi Kurang
90	L	20	8.4	76	Gizi Kurang
91	L	21	8.5	77	Gizi Kurang
92	L	22	8.6	78.1	Gizi Kurang
93	L	50	11.6	93	Gizi Kurang
94	L	51	11.9	93	Gizi Kurang
95	L	6	5.3	59	Gizi Kurang
96	L	7	5.5	61	Gizi Kurang
97	L	8	5.8	62	Gizi Kurang
98	L	27	8.8	79	Gizi Kurang
99	L	28	9	80	Gizi Kurang
100	L	54	12	93.4	Gizi Kurang
101	L	55	12.1	94	Gizi Kurang
102	L	34	10	84.3	Gizi Kurang
103	L	35	10.1	85.4	Gizi Kurang
104	L	36	10.6	86	Gizi Kurang
105	L	60	12.7	97	Gizi Kurang
106	L	0	2.1	49	Gizi Kurang
107	L	17	7.2	72	Gizi Kurang
108	L	18	7.4	73	Gizi Kurang
109	L	43	11	88	Gizi Kurang
110	L	44	10.9	88.3	Gizi Kurang
111	L	45	11.2	89	Gizi Kurang
112	L	60	12.4	96	Gizi Kurang
113	L	46	11.6	91	Gizi Kurang
114	L	47	11.5	91.5	Gizi Kurang
115	L	57	12.4	94.6	Gizi Kurang
116	L	58	12.3	95	Gizi Kurang
117	L	20	7.7	74.2	Gizi Kurang
118	L	24	9	79.4	Gizi Kurang
119	L	25	9.1	79	Gizi Kurang
120	L	59	12.5	96	Gizi Kurang
121	L	1	2.8	48	Gizi Kurang
122	L	2	3.6	51.5	Gizi Kurang
123	L	13	6.5	68	Gizi Kurang
124	L	39	10.7	86.9	Gizi Kurang
125	L	14	7.5	71	Gizi Kurang

126	L	54	12.4	95.5	Gizi Kurang
127	L	51	11.6	92	Gizi Kurang
128	L	0	6	56	Gizi Lebih
129	L	1	6.4	59.9	Gizi Lebih
130	L	3	9.7	68	Gizi Lebih
131	L	9	13	78.2	Gizi Lebih
132	L	10	12.8	79.5	Gizi Lebih
133	L	11	13.8	81	Gizi Lebih
134	L	12	13.9	83.4	Gizi Lebih
135	L	13	14	85	Gizi Lebih
136	L	14	15	85.6	Gizi Lebih
137	L	15	14.8	85.9	Gizi Lebih
138	L	16	15.2	89.2	Gizi Lebih
139	L	17	15.9	88.6	Gizi Lebih
140	L	21	16.8	94	Gizi Lebih
141	L	22	17.5	95	Gizi Lebih
142	L	23	17.8	95.7	Gizi Lebih
143	L	24	18.2	97	Gizi Lebih
144	L	25	18	96.9	Gizi Lebih
145	L	26	18.5	97	Gizi Lebih
146	L	28	19.2	101	Gizi Lebih
147	L	31	20	103.9	Gizi Lebih
148	L	33	20.4	104	Gizi Lebih
149	L	34	20.6	105.4	Gizi Lebih
150	L	35	21	106	Gizi Lebih
151	L	36	21.3	106.9	Gizi Lebih
152	L	37	24	108	Gizi Lebih
153	L	38	23	109.4	Gizi Lebih
154	L	39	22	109	Gizi Lebih
155	L	40	23.3	111	Gizi Lebih
156	L	41	24	110.9	Gizi Lebih
157	L	42	24.5	113.2	Gizi Lebih
158	L	45	25	114.7	Gizi Lebih
159	L	47	25.7	115.9	Gizi Lebih
160	L	48	24	117	Gizi Lebih
161	L	49	25	116.9	Gizi Lebih
162	L	50	24.6	118	Gizi Lebih
163	L	53	27	120.3	Gizi Lebih
164	L	54	26.5	120	Gizi Lebih
165	L	55	27.3	120.4	Gizi Lebih
166	L	56	28	90	Gizi Lebih
167	L	57	27	119	Gizi Lebih
168	L	58	28.4	100	Gizi Lebih

169	L	59	27.9	112	Gizi Lebih
170	P	6	11.3	72	Gizi Lebih
171	P	7	11.7	74.5	Gizi Lebih
172	P	8	12.3	76	Gizi Lebih
173	P	9	13.4	77	Gizi Lebih
174	P	10	13	79	Gizi Lebih
175	P	11	13.5	80.5	Gizi Lebih
176	P	12	13.4	82.3	Gizi Lebih
177	P	13	14	84	Gizi Lebih
178	P	14	14.7	84.3	Gizi Lebih
179	P	15	15.2	86.7	Gizi Lebih
180	P	18	16	90	Gizi Lebih
181	P	19	16.2	90.2	Gizi Lebih
182	P	20	16.7	92	Gizi Lebih
183	P	21	17	93	Gizi Lebih
184	P	22	17.8	94.5	Gizi Lebih
185	P	23	16.9	95.7	Gizi Lebih
186	P	24	17.4	96	Gizi Lebih
187	P	25	17.8	96.5	Gizi Lebih
188	P	29	19.3	99.9	Gizi Lebih
189	P	30	20.3	101	Gizi Lebih
190	P	31	19.7	101.8	Gizi Lebih
191	P	32	20.4	104	Gizi Lebih
192	P	33	21	103.7	Gizi Lebih
193	P	34	20.7	105.4	Gizi Lebih
194	P	35	21.2	105	Gizi Lebih
195	P	36	21	107	Gizi Lebih
196	P	37	22.2	107.5	Gizi Lebih
197	P	39	22.5	108.7	Gizi Lebih
198	P	40	22	109.9	Gizi Lebih
199	P	42	24	112	Gizi Lebih
200	P	43	24	113	Gizi Lebih
201	P	47	24.9	115	Gizi Lebih
202	P	48	24	115.5	Gizi Lebih
203	P	49	25.6	116.9	Gizi Lebih
204	P	50	26	118	Gizi Lebih
205	P	51	27.3	118.4	Gizi Lebih
206	P	52	27	119	Gizi Lebih
207	P	53	28.3	120	Gizi Lebih
208	P	54	28	100	Gizi Lebih
209	P	55	28.6	95	Gizi Lebih
210	P	56	28.5	90	Gizi Lebih
211	P	57	29.4	110	Gizi Lebih

212	L	4	10	72	Gizi Lebih
213	L	5	10.9	72.1	Gizi Lebih
214	L	32	20.2	104.6	Gizi Lebih
215	L	18	16.2	91	Gizi Lebih
216	L	19	16	92.3	Gizi Lebih
217	L	20	17	92.4	Gizi Lebih
218	P	2	8.2	64.3	Gizi Lebih
219	P	3	9	66.4	Gizi Lebih
220	P	4	9.8	69.3	Gizi Lebih
221	P	5	10.5	70.5	Gizi Lebih
222	P	26	18.5	97	Gizi Lebih
223	P	27	18.7	98.9	Gizi Lebih
224	P	28	19	100	Gizi Lebih
225	P	58	29	100	Gizi Lebih
226	P	59	30	97	Gizi Lebih
227	P	60	30.4	110	Gizi Lebih
228	P	44	24.5	113.4	Gizi Lebih
229	P	45	24.3	114	Gizi Lebih
230	P	46	25	114.7	Gizi Lebih
231	P	16	15	87.9	Gizi Lebih
232	P	17	15.7	88	Gizi Lebih
233	L	43	23.5	112	Gizi Lebih
234	L	44	21	112.8	Gizi Lebih
235	L	6	11.6	74.6	Gizi Lebih
236	L	7	12	76	Gizi Lebih
237	L	8	12.9	78	Gizi Lebih
238	L	51	26	117.4	Gizi Lebih
239	L	52	26.3	118	Gizi Lebih
240	P	38	21.5	108	Gizi Lebih
241	L	60	27.8	110	Gizi Lebih
242	P	0	5	55	Gizi Lebih
243	P	1	6	59	Gizi Lebih
244	P	41	22.8	110	Gizi Lebih
245	L	29	19	101.6	Gizi Lebih
246	L	30	19.4	102	Gizi Lebih
247	L	27	18.2	100	Gizi Lebih
248	L	46	24	115	Gizi Lebih
249	L	0	2.9	52	Gizi Baik
250	L	1	5	55	Gizi Baik
251	L	5	7	62	Gizi Baik
252	L	6	8.5	64	Gizi Baik
253	L	7	7.2	69	Gizi Baik
254	L	9	9	68	Gizi Baik

255	L	10	9.1	72	Gizi Baik
256	L	11	9	72	Gizi Baik
257	L	12	9.5	67	Gizi Baik
258	L	15	11	76	Gizi Baik
259	L	16	10	76	Gizi Baik
260	L	17	11	81	Gizi Baik
261	L	18	12	77	Gizi Baik
262	L	19	10.2	80.2	Gizi Baik
263	L	23	11	81.5	Gizi Baik
264	L	24	12	85	Gizi Baik
265	L	25	13	74	Gizi Baik
266	L	26	11.7	83	Gizi Baik
267	L	27	13	84	Gizi Baik
268	L	28	12	81.8	Gizi Baik
269	L	29	13	88	Gizi Baik
270	L	30	14	90	Gizi Baik
271	L	31	14.3	86	Gizi Baik
272	L	32	11	87	Gizi Baik
273	L	33	14	87	Gizi Baik
274	L	37	14.5	89	Gizi Baik
275	L	38	14	95	Gizi Baik
276	L	39	15	91	Gizi Baik
277	L	40	13.2	90	Gizi Baik
278	L	41	14	92	Gizi Baik
279	L	42	15	92	Gizi Baik
280	L	43	16	92	Gizi Baik
281	L	44	15	93	Gizi Baik
282	L	45	14.2	94	Gizi Baik
283	L	46	18	93.4	Gizi Baik
284	L	47	14.2	96	Gizi Baik
285	L	50	16	108.5	Gizi Baik
286	L	51	15	96	Gizi Baik
287	L	52	15.3	97	Gizi Baik
288	L	53	17	97	Gizi Baik
289	L	54	15.6	98	Gizi Baik
290	L	55	18	98	Gizi Baik
291	L	56	16	99	Gizi Baik
292	L	57	18	97.4	Gizi Baik
293	L	58	16	99	Gizi Baik
294	L	59	20	102	Gizi Baik
295	L	60	16	99	Gizi Baik
296	P	0	3	47	Gizi Baik
297	P	1	4.8	53	Gizi Baik

298	P	2	4.5	55	Gizi Baik
299	P	6	6.4	62	Gizi Baik
300	P	7	8	63	Gizi Baik
301	P	8	9	65	Gizi Baik
302	P	9	8.8	68	Gizi Baik
303	P	10	8	66	Gizi Baik
304	P	11	9	68	Gizi Baik
305	P	14	9.8	72	Gizi Baik
306	P	15	10.2	73	Gizi Baik
307	P	16	11	76	Gizi Baik
308	P	17	9.6	77	Gizi Baik
309	P	18	11	75	Gizi Baik
310	P	19	12	82	Gizi Baik
311	P	20	9	86	Gizi Baik
312	P	21	12	86	Gizi Baik
313	P	27	13.2	83	Gizi Baik
314	P	28	14	86.5	Gizi Baik
315	P	29	13	86	Gizi Baik
316	P	30	12.1	86.5	Gizi Baik
317	P	31	11	84	Gizi Baik
318	P	32	14	87	Gizi Baik
319	P	34	12	87	Gizi Baik
320	P	35	12.6	87	Gizi Baik
321	P	40	14	92	Gizi Baik
322	P	41	15	92	Gizi Baik
323	P	42	15.1	91.6	Gizi Baik
324	P	43	17.2	91	Gizi Baik
325	P	44	16	93	Gizi Baik
326	P	45	13.4	93	Gizi Baik
327	P	46	18	100	Gizi Baik
328	P	47	16	93	Gizi Baik
329	P	50	14	98	Gizi Baik
330	P	51	15.3	96	Gizi Baik
331	P	52	20	97	Gizi Baik
332	P	53	16.9	97	Gizi Baik
333	P	54	17	98	Gizi Baik
334	P	55	20	108	Gizi Baik
335	P	56	17	99	Gizi Baik
336	P	57	21	98	Gizi Baik
337	P	58	15	110	Gizi Baik
338	P	59	16	102	Gizi Baik
339	P	60	18	102	Gizi Baik
340	L	2	8.7	65	Gizi Baik

341	L	2	5	58	Gizi Baik
342	L	3	5.6	60	Gizi Baik
343	L	4	6.5	51.5	Gizi Baik
344	L	20	12.1	86	Gizi Baik
345	L	21	11	80	Gizi Baik
346	L	22	12	83	Gizi Baik
347	L	48	17	92	Gizi Baik
348	L	49	13.3	96	Gizi Baik
349	P	3	5	60	Gizi Baik
350	P	4	6	59	Gizi Baik
351	P	5	7	63	Gizi Baik
352	P	22	10.7	83	Gizi Baik
353	P	23	12	80	Gizi Baik
354	P	48	16.4	92	Gizi Baik
355	P	49	13	96	Gizi Baik
356	L	13	10	74	Gizi Baik
357	L	14	10.3	72	Gizi Baik
358	L	34	11.8	92.5	Gizi Baik
359	L	35	13	88	Gizi Baik
360	L	36	15	88	Gizi Baik
361	P	12	8.2	86	Gizi Baik
362	P	13	8	72	Gizi Baik
363	P	38	15.8	86.4	Gizi Baik
364	P	39	14	90	Gizi Baik
365	P	36	14	85.4	Gizi Baik
366	P	37	15	90	Gizi Baik
367	P	24	11	79.4	Gizi Baik
368	P	25	13	80	Gizi Baik
369	P	33	13	86	Gizi Baik
370	L	29	13	88	Gizi Baik
371	L	30	14	90	Gizi Baik
372	P	19	12	82	Gizi Baik
373	P	20	9	86	Gizi Baik
374	P	60	18	102	Gizi Baik
375	L	8	8	66	Gizi Baik
376	P	26	10.1	83	Gizi Baik
377	L	2	3.5	51	Gizi Buruk
378	L	3	3.8	54	Gizi Buruk
379	L	4	4.5	56	Gizi Buruk
380	L	5	5	57.3	Gizi Buruk
381	L	6	4.9	60	Gizi Buruk
382	L	7	5.8	60.4	Gizi Buruk
383	L	8	6	63	Gizi Buruk

384	L	9	5.7	65	Gizi Buruk
385	L	10	6.4	65.6	Gizi Buruk
386	L	11	6	67	Gizi Buruk
387	L	12	6.7	68	Gizi Buruk
388	L	13	6.5	68.6	Gizi Buruk
389	L	14	7	70	Gizi Buruk
390	L	15	7.1	71	Gizi Buruk
391	L	16	7.4	71.4	Gizi Buruk
392	L	17	7.3	73	Gizi Buruk
393	L	18	7	73.5	Gizi Buruk
394	L	19	7.5	73	Gizi Buruk
395	L	20	7.2	75	Gizi Buruk
396	L	24	8.4	77.4	Gizi Buruk
397	L	26	8.7	78.1	Gizi Buruk
398	L	27	8.5	79	Gizi Buruk
399	L	29	9	80.2	Gizi Buruk
400	L	30	9.1	81.3	Gizi Buruk
401	L	31	9.4	81.2	Gizi Buruk
402	L	32	9.5	82	Gizi Buruk
403	L	33	8.8	82	Gizi Buruk
404	L	34	9	82.9	Gizi Buruk
405	L	35	9.6	84	Gizi Buruk
406	L	36	9.5	84.5	Gizi Buruk
407	L	39	10	86.2	Gizi Buruk
408	L	40	10.2	86	Gizi Buruk
409	L	43	10.1	88	Gizi Buruk
410	L	44	10.4	88.1	Gizi Buruk
411	L	45	10.5	88.9	Gizi Buruk
412	L	46	10.6	87	Gizi Buruk
413	L	47	10.7	90	Gizi Buruk
414	L	48	11	90.2	Gizi Buruk
415	L	49	10.8	91	Gizi Buruk
416	L	53	11	92.3	Gizi Buruk
417	L	54	11.5	93	Gizi Buruk
418	L	55	11.3	93.2	Gizi Buruk
419	L	56	11.7	94	Gizi Buruk
420	L	57	11	94.3	Gizi Buruk
421	L	59	11.9	95.2	Gizi Buruk
422	L	60	12	95.7	Gizi Buruk
423	P	4	4	53	Gizi Buruk
424	P	5	4.4	56	Gizi Buruk
425	P	6	4.5	55	Gizi Buruk
426	P	7	5	57	Gizi Buruk

427	P	8	5.2	57.6	Gizi Buruk
428	P	9	5	60	Gizi Buruk
429	P	10	5.6	62.3	Gizi Buruk
430	P	11	5.7	63.4	Gizi Buruk
431	P	14	6.1	67	Gizi Buruk
432	P	15	6.3	65	Gizi Buruk
433	P	16	6.4	68	Gizi Buruk
434	P	17	6.9	70	Gizi Buruk
435	P	18	7	71	Gizi Buruk
436	P	19	6.7	70	Gizi Buruk
437	P	20	7.3	72.2	Gizi Buruk
438	P	21	7.4	74.3	Gizi Buruk
439	P	22	7.6	74.5	Gizi Buruk
440	P	23	7.8	75	Gizi Buruk
441	P	28	8.3	77.5	Gizi Buruk
442	P	29	8.1	78.4	Gizi Buruk
443	P	30	8.5	79.1	Gizi Buruk
444	P	31	8.7	80	Gizi Buruk
445	P	32	7.7	80.8	Gizi Buruk
446	P	33	8.2	80	Gizi Buruk
447	P	34	9	81	Gizi Buruk
448	P	35	8.4	82	Gizi Buruk
449	P	36	9	82.4	Gizi Buruk
450	P	37	9	83	Gizi Buruk
451	P	38	9.4	83.2	Gizi Buruk
452	P	43	9.4	86.4	Gizi Buruk
453	P	44	10	87	Gizi Buruk
454	P	45	9.6	87.6	Gizi Buruk
455	P	46	10.2	88	Gizi Buruk
456	P	47	10	88.3	Gizi Buruk
457	P	48	10.6	88	Gizi Buruk
458	P	49	10.2	89	Gizi Buruk
459	P	50	10.7	90.2	Gizi Buruk
460	P	51	11	90.6	Gizi Buruk
461	P	52	11	91	Gizi Buruk
462	P	56	11.2	92.4	Gizi Buruk
463	P	57	11	93	Gizi Buruk
464	P	58	11.4	93.4	Gizi Buruk
465	P	59	11.7	94	Gizi Buruk
466	P	60	11.1	95	Gizi Buruk
467	P	3	3.8	53	Gizi Buruk
468	L	0	1.9	50	Gizi Buruk
469	L	1	2	47	Gizi Buruk

470	L	21	7.9	75.9	Gizi Buruk
471	L	22	8	76	Gizi Buruk
472	L	23	8.3	77	Gizi Buruk
473	L	37	10	85	Gizi Buruk
474	L	38	9.9	85.4	Gizi Buruk
475	L	50	10.9	91.2	Gizi Buruk
476	L	51	10.5	91.7	Gizi Buruk
477	L	52	11.5	92	Gizi Buruk
478	L	28	8.9	79.4	Gizi Buruk
479	P	0	1.5	45	Gizi Buruk
480	P	1	2.4	45.3	Gizi Buruk
481	P	2	3	50	Gizi Buruk
482	P	41	10	85.6	Gizi Buruk
483	P	42	9.7	86	Gizi Buruk
484	P	24	7.1	75.2	Gizi Buruk
485	P	25	7.9	76	Gizi Buruk
486	P	26	8	75.4	Gizi Buruk
487	P	27	8	77.3	Gizi Buruk
488	P	53	10.3	91.2	Gizi Buruk
489	P	54	11	91.6	Gizi Buruk
490	P	55	11.2	92	Gizi Buruk
491	P	39	9	84	Gizi Buruk
492	P	40	9.5	85	Gizi Buruk
493	L	41	10.2	87	Gizi Buruk
494	L	42	10.5	86	Gizi Buruk
495	L	58	12	94.8	Gizi Buruk
496	P	12	6	65	Gizi Buruk
497	P	13	5.8	65.3	Gizi Buruk
498	L	59	11.9	95.2	Gizi Buruk
499	L	60	12	95.7	Gizi Buruk
500	P	36	9	82.4	Gizi Buruk
501	P	37	9	83	Gizi Buruk
502	L	53	11	92.3	Gizi Buruk
503	L	25	8	76.4	Gizi Buruk
504	P	3	3.5	51	Gizi Buruk

LAMPIRAN B DATA LATIH UNTUK NILAI BATAS 4

INDEKS	JK	UMUR	BB	TB	STATUS
1	P	15	6.9	69.5	Gizi Kurang
2	P	16	7.2	71	Gizi Kurang
3	P	19	7.5	73	Gizi Kurang
4	P	21	7.8	75	Gizi Kurang
5	P	22	8	76	Gizi Kurang
6	P	23	8.2	77.5	Gizi Kurang
7	P	4	4	53	Gizi Buruk
8	P	5	4.4	56	Gizi Buruk
9	P	6	4.5	55	Gizi Buruk
10	P	7	5	57	Gizi Buruk
11	P	8	5.2	57.6	Gizi Buruk
12	P	11	13.5	80.5	Gizi Lebih
13	P	12	13.4	82.3	Gizi Lebih
14	P	13	14	84	Gizi Lebih
15	P	14	14.7	84.3	Gizi Lebih
16	L	15	7.5	72.2	Gizi Kurang
17	L	16	7.8	73	Gizi Kurang
18	L	17	7.9	73.5	Gizi Kurang
19	L	18	8.2	74.5	Gizi Kurang
20	L	19	8.3	75.5	Gizi Kurang
21	L	15	7.1	71	Gizi Buruk
22	L	30	9.1	81.3	Gizi Buruk
23	L	31	9.4	81.2	Gizi Buruk
24	L	32	9.5	82	Gizi Buruk
25	L	33	8.8	82	Gizi Buruk
26	L	34	9	82.9	Gizi Buruk
27	L	35	9.6	84	Gizi Buruk
28	L	25	18	96.9	Gizi Lebih
29	L	26	18.5	97	Gizi Lebih
30	L	28	19.2	101	Gizi Lebih
31	L	31	20	103.9	Gizi Lebih
32	L	33	20.4	104	Gizi Lebih
33	L	34	20.6	105.4	Gizi Lebih
34	L	35	21	106	Gizi Lebih
35	L	36	21.3	106.9	Gizi Lebih
36	L	37	24	108	Gizi Lebih
37	L	38	23	109.4	Gizi Lebih
38	L	39	22	109	Gizi Lebih
39	L	40	23.3	111	Gizi Lebih

40	L	59	12.5	96	Gizi Kurang
41	P	1	2.8	48	Gizi Kurang
42	P	32	9.3	82	Gizi Kurang
43	P	0	2.1	49	Gizi Kurang
44	L	46	11.6	91	Gizi Kurang
45	L	9	13	78.2	Gizi Lebih
46	L	34	20.6	105.4	Gizi Lebih
47	P	23	16.9	95.7	Gizi Lebih
48	P	49	25.6	116.9	Gizi Lebih
49	P	50	26	118	Gizi Lebih
50	P	59	30	97	Gizi Lebih
51	P	60	30.4	110	Gizi Lebih
52	L	43	16	92	Gizi Baik
53	L	44	15	93	Gizi Baik
54	P	14	9.8	72	Gizi Baik
55	P	15	10.2	73	Gizi Baik
56	L	49	13.3	96	Gizi Baik
57	P	3	5	60	Gizi Baik
58	L	9	5.7	65	Gizi Buruk
59	L	10	6.4	65.6	Gizi Buruk
60	L	60	12	95.7	Gizi Buruk
61	P	4	4	53	Gizi Buruk
62	P	5	4.4	56	Gizi Buruk
63	L	53	11	92.3	Gizi Buruk
64	L	25	8	76.4	Gizi Buruk
65	P	3	3.5	51	Gizi Buruk
66	L	28	8.9	79.4	Gizi Buruk
67	P	0	1.5	45	Gizi Buruk
68	L	21	11	80	Gizi Baik
69	L	22	12	83	Gizi Baik
70	L	9	9	68	Gizi Baik
71	L	10	9.1	72	Gizi Baik
72	L	15	7.5	72.2	Gizi Kurang
73	P	16	7.2	71	Gizi Kurang
74	P	19	7.5	73	Gizi Kurang
75	P	59	12.6	95	Gizi Kurang
76	L	6	6	62	Gizi Kurang
77	L	7	6.1	63	Gizi Kurang
78	P	14	14.7	84.3	Gizi Lebih
79	P	15	15.2	86.7	Gizi Lebih
80	P	18	16	90	Gizi Lebih
81	P	38	21.5	108	Gizi Lebih
82	L	60	27.8	110	Gizi Lebih

83	L	27	13	84	Gizi Baik
84	L	28	12	81.8	Gizi Baik
85	P	49	13	96	Gizi Baik
86	L	13	10	74	Gizi Baik
87	L	20	7.2	75	Gizi Buruk
88	L	21	12.2	83	Gizi Baik
89	P	22	11.8	81	Gizi Baik
90	P	19	12	76	Gizi Baik
91	L	16	6.8	71	Gizi Buruk
92	L	18	7.6	73.5	Gizi Buruk
93	P	19	7	75	Gizi Buruk
94	P	59	11.6	96	Gizi Buruk
95	L	8	6.5	60	Gizi Kurang
96	L	19	8	75.5	Gizi Kurang
97	P	19	7.9	80	Gizi Kurang
98	L	23	17	95	Gizi Lebih
99	P	20	13.8	79	Gizi Lebih
100	P	48	24	95	Gizi Lebih
101	P	57	29	112	Gizi Lebih
102	L	21	7.5	74	Gizi Buruk
103	P	17	6.7	73	Gizi Buruk
104	L	15	7.5	72.5	Gizi Kurang
105	L	20	55	98	Gizi Lebih
106	L	2	15	54.5	Gizi Lebih
107	P	3	16	55	Gizi Lebih
108	P	45	9.6	87.6	Gizi Buruk
109	P	46	10.2	88	Gizi Buruk
110	L	11	6	67	Gizi Buruk
111	P	29	9	81	Gizi Kurang
112	P	30	9.2	81	Gizi Kurang
113	P	31	9.1	81.2	Gizi Kurang
114	L	39	15	91	Gizi Baik
115	L	40	13.2	90	Gizi Baik
116	L	41	14	92	Gizi Baik
117	P	56	28.5	90	Gizi Lebih
118	P	57	29.4	110	Gizi Lebih
119	L	57	12.7	95	Gizi Kurang
120	P	31	8.7	80	Gizi Buruk

