

Penentuan Status Gizi Balita pada Kecamatan Kertosono dengan Menggunakan Algoritma C4.5

SKRIPSI

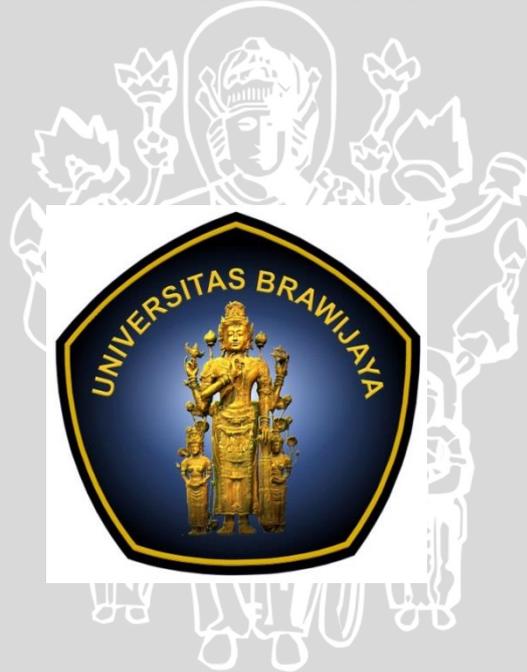
Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Dewi Enggar Wati

NIM: 115060807113036

UNIVERSITAS BRAWIJAYA



TEKNIK INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

PENENTUAN STATUS GIZI BALITA PADA KECAMATAN KERTOSONO DENGAN
MENGUNAKAN ALGORITMA C4.5

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Dewi Enggar Wati

NIM: 115060807113036

Skripsi ini telah diuji dan dinyatakan lulus pada
21 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Indriati, ST., M. Kom

NIP: 198310132015042002

Muhammad Tanzil furgon, S.Kom,M.Comp.Sc.

NIP: 198209302008011004

Mengetahui

Ketua Program Studi Teknik Informatika

Drs. Marji, M.T

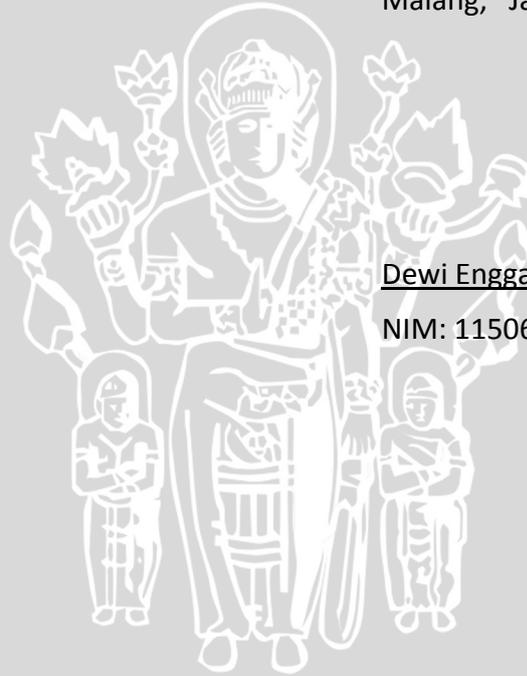
NIP: 16708011992031001

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, Januari 2016



Dewi Enggar Wati

NIM: 115060807113036

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Penentuan Status Gizi Balita Pada Kecamatan Kertosono Dengan Menggunakan Algoritma C4.5” ini dapat terselesaikan. Skripsi ini diajukan untuk memenuhi prasyarat kelulusan dalam menyelesaikan pendidikan dan memperoleh gelar Sarjana.

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

1. Indriati, S.T., M.Kom dan Muhammad Tanzil furqon, S.Kom,M.Comp.Sc selaku pembimbing satu dan dua yang telah sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. M. Ali Fauzi S.Kom., M.Kom selaku dosen yang telah meluangkan waktu untuk memberikan pengarahan dan masukan bagi penulis.
3. Ir. Sutrisno, M.T. selaku Ketua Fakultas Ilmu Komputer Universitas Brawijaya.
4. Drs. Marji, M.T. selaku Kepala Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
5. Segenap Bapak dan Ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
6. Orang tua, suami serta mertua terimakasih atas segala doa restu serta dukungannya baik secara materil maupun non materil sehingga penulis bisa menyelesaikan skripsi ini.
7. Seluruh teman-teman seperjuangan penulis di grup “ababil (Armeilya, Zain, Ayu, Lusi, Zha, Julia, dan Ermi)”, “TIF 2011” dan “UNP” yang tidak bisa saya sebutkan satu persatu, terimakasih atas segala motivasi dan semangat yang telah diberikan serta doa yang selalu dipanjatkan untuk penulis.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangatlah diharapkan oleh penulis. Dengan penuh harapan semoga apa yang telah dikerjakan penulis dalam laporan skripsi ini dapat bermanfaat bagi semua pihak yang menggunakannya.

Malang, 1 Januari 2015

Penulis

Dewi Enggar Wati

ABSTRAK

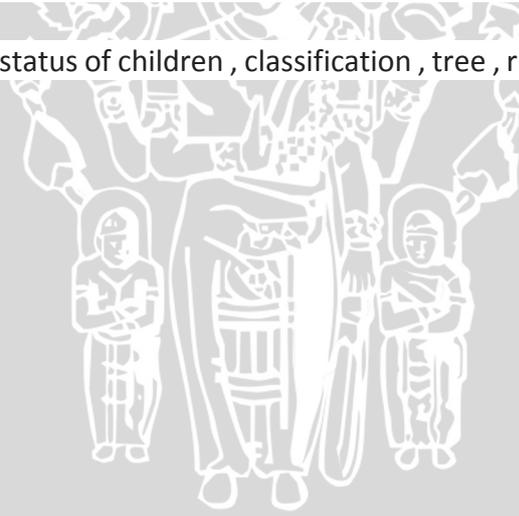
Indonesia sebagai negara berkembang selalu berupaya melakukan peningkatan derajat kesehatan masyarakat, karena pemerintah memiliki kewajiban terhadap kesejahteraan masyarakat, salah satunya melalui peningkatan gizi pada balita. Tidak hanya peningkatan gizi saja yang wujudkan tetapi penanganan yang tepat dan cepatlah yang seharusnya dilakukan sekarang ini. Pada pengelompokan status gizi balita berdasarkan standar kementerian kesehatan Indonesia yaitu Gizi baik, Gizi lebih, Gizi buruk dan Gizi Kurang, kategori ini digunakan sebagai alat bantu untuk memudahkan pengklasifikasian dengan algoritma klasifikasi yang baik dan cocok untuk diterapkan dalam permasalahan ini, yang nantinya akan mengefektifkan penanganan dan perhatian yang cepat dari pemerintah terkait. Adapun Algoritma klasifikasi yang dipilih peneliti dalam penyelesaian kasus ini adalah Algoritma C4.5. Hal ini disebabkan Algoritma C4.5 ini memiliki tingkat akurasi yang baik diterapkan untuk beberapa penelitian yang telah dilakukan sebelumnya. Penggunaan metode Algoritma C4.5 akan mengolah data yang ada untuk diklasifikasikan sehingga menghasilkan keluaran Status gizi balita (gizi baik, gizi lebih, gizi buruk dan gizi kurang dengan menggunakan masukan umur, berat badan dan jenis kelamin. Nilai rata-rata akurasi yang dihasilkan dalam kasus ini adalah sebesar 97,5%, hasil akurasi ini didapatkan melalui pengujian jumlah data *training* yang berbeda dan jumlah data *testing* yang sama.

Kata kunci: status gizi balita, klasifikasi, *tree*, *rule*, Algoritma C4.5

ABSTRACT

Indonesia as a developing country always seeks to improve community health status, because the government has an obligation to the welfare of the people, one of them through increased malnutrition in children under five. Not only improving nutrition are embodied but proper handling and hurry is supposed to do today. In the grouping of nutritional status of children based on the standard Ministry of Health Indonesia, nutrition good, nutrition is more, Malnutrition and Nutrition Less, this category is used sebagai tool to facilitate the classification of the classification algorithm is good and suitable for application in on this issue, which will streamline handlers and prompt attention of the relevant government. The classification algorithm is selected researchers in the settlement of this case is the algorithm C4.5. This is due to the C4.5 algorithm has a good degree of accuracy is applied to some of the research that has been done before. The use of the method C4.5 algorithms will process the data to be classified so as to produce output nutritional status (good nutrition, over nutrition, malnutrition and nutrition kuran using input age, weight and gender. The average value of accuracy resulting in This case is 97.5%, this accuracy results obtained through testing different amount of training data and testing the same amount of data.

Keywords : nutritional status of children , classification , tree , rule , algorithm C4.5



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	3
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan	4
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Pengertian Gizi	6
2.2.1 Pengertian Balita	6
2.2.2 Pengertian Status Gizi	7
2.2.3 Standar Baku Antropometri	7
2.2.4 Klasifikasi Status Gizi Balita	9
2.2.5 Metode Penilaian Gizi Balita	10
2.3 Pengertian Data Mining	10
2.3.1 Klasifikasi Data Mining	11
2.4 Pengertian Pohon Keputusan	12
2.4.1 Model Pohon Keputusan	12
2.4.2 Macam-Macam Algoritma Keputusan	13
2.4.3 Pengertian Algoritma C4.5	13

2.4.4 Tahap-Tahapan	13
2.4.5 Kelebihan ALgoritma C4.5	14
2.5 Perhitungan Akurasi	14
BAB 3 METODOLOGI	16
3.1 Metodologi Penelitian	16
3.1.1 Studi Literatur	16
3.1.2 Pengumpulan Data	17
3.1.3 Analisa Kebutuhan Sistem	17
3.1.4 Perancangan Sistem	18
3.1.5 Implementasi	18
3.1.6 Pengujian Sistem	18
3.1.7 Evaluasi	19
3.2 Perancangan Sistem	19
3.3 Perancangan Antarmuka	23
3.3.1 Tampilan Awal	24
3.3.2 Modul Data <i>Training</i>	24
3.3.3 Modul Data <i>testing</i>	25
3.3.4 Modul Klasisfikasi	25
3.4 Perhitungan Manual	26
BAB 4 IMPLEMENTASI	41
4.1 Lingkungan Implementasi	41
4.1.1 Lingkungan <i>Hadrware</i>	41
4.1.2 Lingkungan <i>Software</i>	41
4.2 Persiapan Data	41
4.3 Implementasi Antar Muka	42
4.3.1 Tampilan Awal	42
4.3.2 Modul <i>Training</i>	42
4.3.3 Modul <i>Testing</i>	44
4.3.4 Modul Klasifikasi	45
4.4 Implementasi Program	46
4.4.1 Program Proses Algoritma C4.5	46
BAB 5 PENGUJIAN DAN PEMBAHASAN	62



5.1 Sistematika Pengujian 62

5.1.1 Sistematika Pengujian Data *Training* dan Data *Testing* Sama 62

5.1.2 Sistematika Pengujian Pengaruh Perubahan Data *Training* dengan Data *Testing* yang Sama..... 62

5.2 Hasil Pengujian..... 62

5.2.1 Hasil Pengujian dengan Data *Training* dan Data *Testing* yang Sama 63

5.2.2 Hasil Uji Pengaruh Data *Training* dan Data *Testing*..... 63

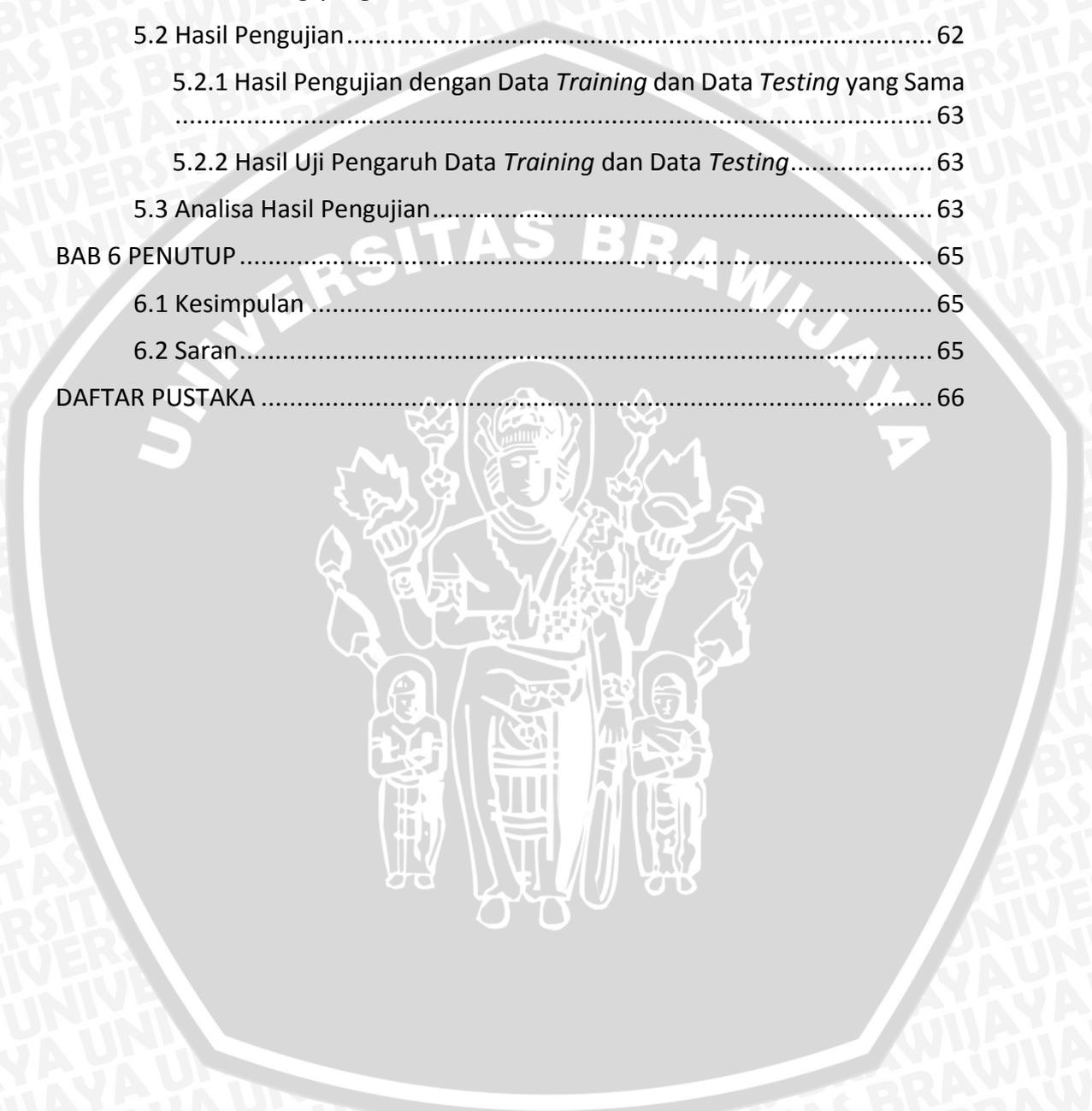
5.3 Analisa Hasil Pengujian..... 63

BAB 6 PENUTUP 65

6.1 Kesimpulan 65

6.2 Saran..... 65

DAFTAR PUSTAKA 66



DAFTAR TABEL

Tabel 2. 1 Kategori dan Ambang Batas Status Gizi Anak	8
Tabel 2. 2 contoh simpangan baku.....	8
Tabel 3. 1 Spesifikasi perangkat keras.....	17
Tabel 3.2 Tabel pengujian akurasi	19
Tabel 3.3 Data Latih pada Perhitungan Manual	26
Tabel 3.4 Berat badan dan tinggi badan ideal balita	27
Tabel 3.5 Perhitungan Jumlah Kasus dari Data Latih	28
Tabel 3.6 Hasil Perhitungan Node 1 Nilai Entropy dan Gain.....	30
Tabel 3.7 Hasil node cabang 1.1.....	31
Tabel 3.8 Data Node 1.1 balita umur 0 -12 tahun.....	32
Tabel 3.9 Data Balita Node 1.2 Umur 13 – 24 Tahun	32
Tabel 3.10 Data Balita Node 1.2 Umur 13 – 24 Tahun	33
Tabel 3.11 Data balita node 1.2.1 umur 13 – 24 tahun dan BB 0 – 9,9 kg	34
Tabel 3.12 Hasil node 1. 2.1	34
Tabel 3.13 Data balita node 1.3 umur 25 – 36 tahun.....	36
Tabel 3.14 Data balita node 1.3 umur 25 – 36 tahun.....	36
Tabel 3.15 Data balita node 1.3.1.....	37
Tabel 3.16 Perhitungan entropy dan gain node 1.3.1	37
Tabel 3.17 Data balita node 1.5	39
Tabel 3.18 Perhitungan entropy dan gain node 1.5.....	39

DAFTAR GAMBAR

Gambar 2. 1 Model Pohon Keputusan.....	12
Gambar 3. 1 Langkah-langkah penelitian	16
Gambar 3. 2 Diagram blok sistem pengklasifikasian data balita menggunakan algoritma C4.5	18
Gambar 3. 3 Diagram alir sistem pengklasifikasian status gizi balita menggunakan Algoritma C4.5	20
Gambar 3. 4 Gambar di diagram alir proses data testing.....	21
Gambar 3. 5 Gambar di diagram alir Algoritma C4.5	22
Gambar 3. 6 Diagram alir proses data pengujian data <i>testing</i>	23
Gambar 3. 7 Tampilan awal.....	24
Gambar 3. 8 Modul data <i>training</i>	24
Gambar 3. 9 Antarmuka modul data <i>testing</i>	25
Gambar 3. 10 Antar muka klasifikasi status gizi balita	25
Gambar 4.1 Tampilan Awal.....	42
Gambar 4. 2 Modul Training	43
Gambar 4. 3 Tampilan Decision Tree.....	43
Gambar 4. 4 Tampilan Decision Rule.....	44
Gambar 4. 5 Modul Testing.....	44
Gambar 4. 6 Modul Klasifikasi.....	45

DAFTAR LAMPIRAN

Lampiran A data Balita.....	67
Lampiran B standar antropometri	75



BAB 1 PENDAHULUAN

Pada bagian ini menjelaskan mengenai latar belakang muncul ide atau gagasan dari suatu masalah yang diangkat oleh peneliti. Selanjutnya merumuskan masalah dan batasan masalah yang diangkat, tujuan manfaat serta penulisan sistem pembahasan pada penelitian ini.

1.1 Latar belakang

Indonesia sebagai negara berkembang selalu berupaya melakukan peningkatan derajat kesehatan masyarakat, karena pemerintah memiliki kewajiban terhadap kesejahteraan masyarakat, salah satunya melalui peningkatan gizi pada balita. Hal ini sudah diwujudkan oleh pemerintah dengan adanya PUSKESMAS (Pusat Kesehatan Masyarakat) disetiap daerah, yang bertujuan untuk mengontrol dan memantau kesehatan serta asupan gizi balita pada setiap pemerintahan daerah masing-masing. Dengan adanya peran puskesmas disetiap daerah, pemerintah dapat mengantisipasi adanya gizi buruk pada balita, dan dapat menekan angka gizi buruk yang ada, sehingga para balita yang terjangkit gizi buruk mendapatkan perlakuan dan perhatian khusus.

Hingga sekarang ini belum ada aplikasi yang dapat menganalisis status gizi balita pada suatu puskesmas salah satunya di Puskesmas Kertosono Kabupaten Nganjuk. Status gizi balita bertujuan untuk memonitoring perkembangan dan kondisi balita oleh orang tua, oleh sebab itu status gizi sangat penting diketahui oleh orang tua agar tidak terjadi gizi kurang dan gizi lebih yang dapat mengakibatkan kematian dan penyakit kronis. Analisis yang diterapkan masih secara manual dengan menggunakan KMS (Kartu Menuju Sehat) balita. Penggunaan KMS ini untuk memantau pertumbuhan anak dibawah umur 5 tahun dan juga penggunaan KMS ini harus sesuai dengan peraturan yang sudah ditetapkan, (Menteri Kesehatan RI, 2011). Dengan perhitungan dan analisis secara manual tersebut, akibatnya akan membutuhkan waktu yang sangat lama untuk menganalisis status gizi setiap balita yang ada di Kecamatan Kertosono, karena setiap balita memiliki KMS yang berbeda-beda serta dari segi hasil perhitungan juga belum tentu akurat. Sehingga diperlukan sebuah aplikasi untuk mengatasi masalah tersebut.

Aplikasi yang nantinya diterapkan pada Puskesmas Kecamatan Kertosono ini merupakan aplikasi untuk penentuan status gizi balita. Dalam perancangan aplikasi penentuan status gizi balita ini, dapat dikategorikan menjadi 4 bagian status gizi yaitu gizi buruk, gizi kurang, gizi baik, dan gizi lebih dengan menggunakan masukan jenis kelamin, umur dan berat badan. Sehingga nantinya program yang dibuat sesuai dengan yang diharapkan, yaitu mampu menentukan status gizi pada balita secara akurat dan mampu menekan angka gizi buruk pada balita di Puskesmas Kertosono tersebut. Untuk menilai status gizi balita dan anak diperlukan standar antropometri yang mengacu pada Standar *World Health Organization* (WHO 2005), (Menteri Kesehatan RI, 2011).

Dalam penerapan aplikasi penentuan gizi balita, pastinya dibutuhkan algoritma dalam pengolahan datanya, sehingga peneliti memilih menggunakan Algoritma C4.5 yang dianggap tepat untuk diterapkan dalam pembuatan aplikasi penentuan status gizi balita tersebut. Algoritma C4.5 adalah sebuah algoritma yang digunakan untuk membangun *decision tree* (pengambilan keputusan). Pohon Keputusan yang merupakan metode klasifikasi dan prediksi yang sangat kuat dan terkenal, (Kusrini, 2009).

Algoritma C.45 adalah salah satu algoritma induksi pohon keputusan yaitu ID3 (*Iterative Dichotomiser 3*). ID3 dikembangkan oleh J. Ross Quinlan. Dalam prosedur algoritma ID3, input berupa sampel *training*, label *training* dan atribut. Algoritma C4.5 merupakan pengembangan dari ID3. Pohon dibangun dengan cara membagi data secara rekursif hingga tiap bagian terdiri dari data yang berasal dari kelas yang sama. Bentuk pemecahan (*split*) yang digunakan untuk membagi data tergantung dari jenis atribut yang digunakan dalam *split*. Algoritma C4.5 dapat menangani data numerik (kontinyu) dan diskret, (Kusrini, 2009).

Adapun pertimbangan peneliti memilih Algoritma C4.5 karena dari Algoritma C4.5 memiliki beberapa kelebihan dibandingkan metode atau algoritma yang lain, yaitu Algoritma C4.5 ini menggunakan konsep *information gain* atau *entropy reduction* untuk memilih percabangan yang optimal. Algoritma C4.5 secara rekursif mengunjungi tiap simpul keputusan, memilih percabangan optimal, sampai tidak ada cabang lagi yang mungkin dihasilkan, (Larose, 2005). Pada aplikasi yang dibuat oleh peneliti yaitu menerapkan Algoritma C4.5 dengan masukan yaitu Berat Badan (BB), Umur (U), dan Jenis Kelamin. Data *masukan* ini akan diolah menggunakan Algoritma C4.5, sehingga menghasilkan *keluaran* yang menentukan status gizi balita yaitu gizi buruk, gizi kurang, gizi baik, dan gizi lebih.

Sebelumnya ada 3 penelitian menggunakan Algoritma C4.5 yang pertama digunakan untuk penentuan jurusan mahasiswa dengan papernya yang berjudul "Penerapan Algoritma C4.5 Untuk Penentuan Jurusan Mahasiswa". Dalam penelitian tersebut peneliti membandingkan metode *Naïve bayes* dengan Algoritma C4.5. Hasil akurasi Algoritma C4.5 menghasilkan tingkat keakurasian lebih besar yaitu 93,31% dibandingkan metode *naïve bayes* yang menghasilkan tingkat akurasi 66,36%. Data yang digunakan yaitu data mahasiswa baru STMIK Indonesia Banjarmasin tahun 2008 sampai dengan 2009, 90% untuk membangun struktur pohon dan 10% sebagai data uji, (Liliana, 2013).

Penelitian yang kedua adalah Anik Andriani (2012) mengenai penerapan Algoritma C4.5 pada program klasifikasi mahasiswa *dropout*. Penelitian tersebut menggunakan atribut Waktu kuliah, IPK Semester I, Kehadiran Semester I, Jenis Kelamin, Usia masuk, Asal daerah, Jurusan SLTA, Orangtua, Penghasilan Ortu, Biaya studi, Bekerja, dan Beasiswa. Penelitian ini menghasilkan evaluasi dan validasi dengan Confussion matrix menunjukkan tingkat akurasi pada Algoritma C4.5 sebesar 97,75%, (Anik, 2013).

Penelitian yang ketiga dilakukan oleh Dwi Ayu Nursela (2014) yang berjudul penerapan Algoritma C4.5 untuk klasifikasi tingkat keganasan kanker payudara

dapat diselesaikan menggunakan teknik *data mining*, yaitu Algoritma C4.5. Dalam penelitian ini diterapkan pembobotan atribut adalah pembentukan *rules* dari *uniformity of cell size* yang sederhana. Data yang digunakan dalam penelitian ini adalah data sampel dari *public dataset* UCI, karena data dari pasien kanker payudara merupakan data yang *private* dan sulit untuk didapatkan. Tingkat akurasi yang dihasilkan dari pemodelan algoritma C4.5 dengan pembobotan atribut, yaitu sebesar 98,57%. Akurasi tersebut diperoleh dari kesesuaian antara prediksi klasifikasi dan hasil klasifikasi. Oleh karena itu, peneliti mengusulkan tugas akhir ini dengan judul “*Penentuan Status Gizi Balita pada Kecamatan Kertosono dengan Menggunakan Algoritma C4.5*”.

1.2 Rumusan masalah

Permasalahan yang ada pada penelitian ini dapat dirumuskan sebagai berikut :

1. Bagaimana penerapan Algoritma C4.5 untuk penentuan status gizi pada balita di Kecamatan Kertosono ?
2. Bagaimana hasil akurasi dari penerapan Algoritma C4.5 untuk penentuan status gizi pada balita di Kecamatan Kertosono ?

1.3 Tujuan

Sesuai dengan rumusan masalah dan latar belakang, tujuan yang ingin dicapai pada tugas akhir ini adalah menentukan status gizi balita pada Kecamatan Kertosono dengan menggunakan Algoritma C4.5.

1.4 Manfaat

Manfaat yang diharapkan dalam penelitian ini adalah hasil dari penentuan status gizi balita ini dapat dimanfaatkan oleh pemerintah untuk memonitor gizi balita pada Kecamatan Kertosono Kabupaten Nganjuk, sehingga diharapkan adanya perhatian khusus pada balita yang terjangkit gizi buruk mendapatkan penanganan khusus agar tidak terjadi kematian dan penyakit kronis.

1.5 Batasan masalah

Penelitian ini dibatasi oleh hal-hal berikut:

1. Data yang digunakan adalah data gizi balita dari Puskesmas Kecamatan Kertosono pada bulan November 2014 sampai Februari tahun 2015 yang digunakan hanya untuk menghitung status gizi balita saja.
2. Parameter yang akan dioptimasi adalah beberapa parameter masukan berupa Berat Badan (BB), Umur (U), dan Jenis Kelamin.
3. Hasil status gizi ada 4 kategori yaitu gizi buruk, gizi kurang, gizi baik, dan gizi lebih.

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

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

BAB IV HASIL

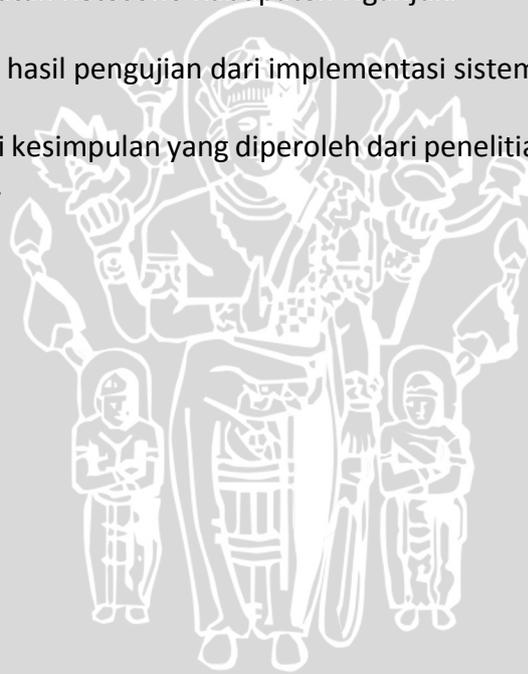
Membahas tentang implementasi Algoritma C4.5 untuk penentuan status gizi balita pada Kecamatan Ketosono Kabupaten Nganjuk.

BAB V PEMBAHASAN

Membahas tentang hasil pengujian dari implementasi sistem

BAB VI PENUTUP

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



BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi kajian pustaka dan dasar teori yang berkaitan dengan judul yang digunakan dalam penyusunan penelitian tugas akhir ini. Dalam tugas akhir ini peneliti mengambil judul Penentuan Status Gizi Balita pada Kecamatan Kertosono Menggunakan Algoritma C4.5. Adapun kajian pustaka mengenai penelitian yang dilakukan sebelumnya terkait dengan judul yang diangkat peneliti sekarang ini, sedangkan dasar teori digunakan sebagai teori pendukung dalam penelitian ini.

2.1 Kajian Pustaka

Berdasarkan judul penelitian yang diangkat oleh peneliti, peneliti menemukan beberapa penelitian sebelumnya yang nantinya mendukung dalam penelitian ini. Penelitian-penelitian yang sebelumnya menggunakan algoritma C4.5 dalam beberapa masalah klasifikasi.

Penelitian pertama adalah penelitian yang dilakukan oleh Liliana Swastina (2013) mengenai penerapan Algoritma C4.5 untuk penentuan jurusan mahasiswa. Atribut yang digunakan adalah NIM, Nama, Tanggal Lahir, Asal Sekolah, Nilai UN, dan jurusan yang dipilih. Penelitian ini menggunakan data mahasiswa baru STMIK Indonesia Banjarmasin tahun 2008 s.d 2009, yaitu sebanyak 90% akan digunakan untuk membangun struktur pohon keputusan melalui Algoritma C4.5 dan 10% digunakan sebagai data uji. Dalam penelitian yang dilakukan, peneliti menggunakan 2 Algoritma sebagai perbandingan yaitu Algoritma C4.5 dan Naïve Bayes, yang menghasilkan tingkat akurasi sebesar 82,64 % untuk Algoritma C4.5 dan 66,36% untuk Naïve Bayes. Tingkat akurasi terbesar yaitu menggunakan Algoritma C4.5 dalam pengolahan data penentuan jurusan mahasiswa, Liliana (2013).

Penelitian yang kedua adalah Anik Andriani (2012) mengenai penerapan Algoritma C4.5 pada program klasifikasi mahasiswa *dropout*. Penelitian tersebut menggunakan atribut Waktu kuliah, IPK Semester I, Kehadiran Semester I, Jenis Kelamin, Usia masuk, Asal daerah, Jurusan SLTA, Orangtua, Penghasilan Ortu, Biaya studi, Bekerja, dan Beasiswa. Penelitian ini menghasilkan evaluasi dan validasi dengan Confusion matrix menunjukkan tingkat akurasi pada Algoritma C4.5 sebesar 97,75%, (Anik, 2012).

Penelitian yang ketiga dilakukan oleh (Dwi Ayu Nursela, 2014) yang berjudul penerapan Algoritma C4.5 untuk klasifikasi tingkat keganasan kanker payudara dapat diselesaikan menggunakan teknik *data mining*, yaitu Algoritma C4.5. Dalam penelitian ini diterapkan pembobotan atribut adalah pembentukan *rules* dari *uniformity of cell size* yang sederhana. Data yang digunakan dalam penelitian ini adalah data sampel dari *public dataset* UCI, karena data dari pasien kanker payudara merupakan data yang *private* dan sulit untuk didapatkan. Tingkat akurasi yang dihasilkan dari pemodelan algoritma C4.5 dengan pembobotan atribut, yaitu sebesar 98,57%. Akurasi tersebut diperoleh dari kesesuaian antara prediksi klasifikasi dan hasil klasifikasi.

Dari tiga penelitian sebelumnya diatas, dapat disimpulkan bahwa akurasi yang dihasilkan dengan Algoritma C4.5 cukup tinggi, perbedaan yang ada pada ketiga penelitian tersebut dengan diterapkan Algoritma C4.5 diberbagai permasalahan atau kasus yang berbeda, sehingga dalam penelitian ini peneliti ingin menerapkan Algoritma C4.5 untuk penentuan status gizi balita pada Kecamatan Ketosono Kabupaten Nganjuk. Yang nantinya diharapkan dengan diterapkannya Algoritma C4.5 ini sudah sesuai dengan tujuan dan juga menghasilkan tingkat akurasi yang tinggi selayaknya tiga penelitian sebelumnya.

2.2 Pengertian Gizi

Gizi adalah suatu proses organisme menggunakan makanan yang dikonsumsi secara normal melalui proses pencernaan, absorpsi, transportasi, penyimpanan, metabolisme dan pengeluaran zat-zat yang tidak digunakan untuk mempertahankan kehidupan, pertumbuhan dan fungsi normal dari organ-organ, serta menghasilkan energy, (Supariasa, 2002). (Almatsier, 2005) Gizi yang seimbang dikelompokkan berdasarkan tiga fungsi utama yaitu:

- a. Sumber energi atau tenaga yaitu padi-padian atau serealisa seperti beras, jagung, gandum, umbi-umbian seperti ubi singkong dan talas serta hasil olahannya seperti tepung-tepungan, mie dan bihun.
- b. Sumber protein yaitu sumber protein hewani, seperti daging ayam telur dan susu. Sumber protein nabati, seperti kacang-kacangan: kacang kedelai kacang tanah, kacang hijau, kacang merah, kacang tolo, serta hasil olahannya seperti tempe, tahu, susu kedelai, dan oncom.
- c. Sumber zat pengatur seperti sayuran dan buah, sayuran diutamakan yang berwarna hijau dan jingga, seperti bayam, daun singkong, daun katuk, kangkung, wortel, serta sayur kacang kacang seperti kacang panjang, buncis dan kecipir. Buah-buahan yang diutamakan yang berwarna jingga dan kaya akan serat dan barasa asam, seperti pepaya, mangga, nanas, nangka masak, jambu biji, apel, sirsat dan jeruk.

Selain bahan makanan di atas dalam makanan sehari-hari kita mengenal sumber lemak murni, seperti minyak goreng, margarine, mentega serta sumber karbohidrat murni seperti gula pasir, gula merah madu dan sirup. Zat gizi seimbang tersebut telah dijadikan patokan oleh para ahli gizi sehingga lahirlah apa yang disebut Pedoman Umum Gizi Seimbang (PUGS).

2.2.1 Pengertian Balita

Pengertian balita itu sendiri adalah anak yang telah menginjak usia di atas satu tahun atau lebih populer dengan pengertian usia anak di bawah lima tahun, (Muaris, 2006). Balita adalah istilah umum bagi anak usia 1-3 tahun (batita) dan anak prasekolah (3-5 tahun). Saat usia batita, anak masih tergantung penuh ada orang tua untuk melakukan kegiatan penting, seperti mandi, buang air dan makan. Perkembangan berbicara dan berjalan sudah bertambah baik. Namun kemampuan lain masih terbatas. Masa balita merupakan periode penting dalam proses tumbuh kembang manusia. Perkembangan dan pertumbuhan di masa itu menjadi penentu keberhasilan pertumbuhan dan perkembangan anak di periode

selanjutnya. Masa tumbuh kembang di usia ini merupakan masa yang berlangsung cepat dan tidak akan pernah terulang, karena itu sering disebut *golden age* atau masa keemasan, (Sutomo, 2010).

2.2.2 Pengertian Status Gizi

Status Gizi adalah sesuatu keadaan tubuh akibat mengkonsumsi makanan serta penggunaan zat-zat gizi, (Almatsier, 2005). Status Gizi dapat didefinisikan sebagai ekspresi tubuh dari keadaan keseimbangan dalam bentuk variabel tertentu, (Supriasa, 2002). Didalam penerapan status gizi dipengaruhi oleh konsumsi makanan dan penggunaan zat-zat gizi di dalam tubuh. Bila tubuh memperoleh cukup zat-zat gizi dan digunakan secara efisien akan tercapai status gizi optimal yang memungkinkan pertumbuhan fisik, perkembangan otak, kemampuan kerja dan kesehatan secara umum pada tingkat setinggi mungkin, (Almatsier, 2005). Sedangkan menurut (Djoko Pekik Irianto, 2007), status gizi adalah ekspresi dari keadaan keseimbangan dalam bentuk variabel tertentu atau dapat dikatakan bahwa status gizi merupakan indikator baik buruknya penyediaan makanan sehari-hari.

2.2.3 Standar Baku Antropometri

Dalam penentuan status gizi pada balita harus mempunyai ukuran baku atau standar. Di Indonesia sekarang ini menggunakan ukuran baku antropometri yang mengacu pada standar *World Health Organization* (WHO 2005), (Menteri Kesehatan RI, 2011). Adapun penggunaan standar antropometri WHO 2005, (Menteri Kesehatan RI, 2011) adalah sebagai berikut:

- a. Umur dihitung dalam bulan penuh. Contoh umur 2 bulan 29 hari dihitung sebagai umur 2 bulan.
- b. Ukuran Panjang Badan (PB) digunakan untuk anak umur 0 sampai 2 bulan yang diukur telentang. Bila anak umur 0 sampai 24 bulan diukur berdiri, maka hasil pengukurannya dikoreksi dengan menambahkan 0 cm.
- c. Ukuran Tinggi Badan (TB) digunakan untuk anak umur diatas 24 bulan yang diukur berdiri. Bila anak umur diatas 24 bulan diukur telentang, maka hasil pengukurannya dikoreksi dengan mengurangi 0,7 cm.
- d. 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).
- e. Pendek dan sangat pendek adalah status gizi yang didasarkan pada indeks Panjang Badan menurut umur (PB/U) atau Tinggi Badan menurut Umur (TB/U) yang merupakan padanan istilah *stunted* (pendek) dan *severely stunted* (sangat pendek).
- f. Kurus dan sangat kurus adalah status gizi yang didasarkan pada indeks Berat Badan menurut Panjang Badan (BB/PB) atau Berat Badan menurut Tinggi Badan (BB/TB) yang merupakan padanan istilah *wasted* (kurus) dan *saverely wasted* (sangat kurus).

Selain penggunaan standar antropometri WHO 2005, pemerintah Indonesia juga menetapkan kategori dan juga ambang batas status gizi pada anak.

Kategori dan ambang batas status gizi pada anak adalah sebagai mana terdapat pada tabel di bawah ini Menteri Kesehatan RI 2011:

Tabel 2. 1 Kategori dan Ambang Batas Status Gizi Anak

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

Sumber: (Menteri Kesehatan RI, 2011)

Untuk pengukuran antropometri digunakan Berat Badan dibagi dengan Usia (BB/U) sebab memiliki kelebihan yaitu lebih mudah dan dapat dimengerti masyarakat pada umumnya. Untuk mengatur status gizi balita akut dan kronis, berat badan dapat berfluktuasi serta sensitif terhadap perubahan kecil dan juga dapat mendeteksi kegemukan [KHO-09]. Dalam penentuan status gizi ini menggunakan Standar Deviasi (SD) atau Z-skor digunakan sebagai ambang batas untuk mengetahui perkembangan serta klasifikasi status gizi. Adapun persamaan (2.1) Standar Deviasi (SD) adalah sebagai berikut ini (Anggraeni, 2010):

$$SD = \frac{\text{Nilai individu subyek} - \text{Nilai medianbaku rujukan}}{\text{Nilai simpang baku rujukan}} \quad (2.1)$$

Adapun sebagai contoh perhitungan status gizi balita menggunakan Z-Skor adalah sebagai berikut:

Seorang anak laki-laki berumur 26 bulan dan berat badan 15 kg, dan seorang anak laki-laki dengan umur 11 bulan serta berat badan 5 kg.
Distribusi Simpang Baku (BB/U)

Tabel 2. 2 contoh simpangan baku

Umur	Simpang Baku						
	-3 SD	-2 SD	-1 SD	Median	+1 SD	+2 SD	+3 SD
11 Bulan	6,8	7,6	8,4	9,4	10,5	11,7	13
26 Bulan	8,9	10,0	11,2	12,5	14,1	15,8	17,8

Sumber: (Menteri Kesehatan RI, 2011)

untuk kasus bayi 11 bulan, berat badannya (5 kg) lebih kecil daripada nilai median (9,4), maka dari itu nilai simpang baku rujukannya menjadi $9,4 - 8,4 = 1$ sehingga perhitungan *z score* :

$$\frac{5 - 9,4}{9,4 - 8,4} = -4,4$$

Karena nilai *z score* sudah mencapai -4,4 berarti status gizinya tergolong buruk.

untuk balita 26 bulan caranya sama dengan diatas. Karena berat badannya (15 kg) lebih besar daripada nilai simpang baku mediannya (12,5), maka dari itu nilai simpang baku rujukannya diperoleh dengan mengurangi nilai simpang baku +1SD dengan nilai median, yakni $14,1 - 12,5 = 1,6$

sehingga perhitungan *z score* menjadi :

$$\frac{15 - 12,5}{14,1 - 12,5} = 1,56$$

karena nilai *z score*-nya 1,56 maka status gizinya tergolong baik. Adapun penjelasan tabel penggolongan gizi balita terdapat pada lampiran.

2.2.4 Klasifikasi Status Gizi Balita

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

a. Gizi lebih (*Over weight*)

Gizi lebih terjadi bila tubuh memperoleh zat-zat gizi dalam jumlah berlebihan sehingga menimbulkan efek toksis atau membahayakan. 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.

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.

c. Gizi kurang (*under weight*)

Status gizi kurang terjadi bila tubuh mengalami kekurangan satu atau lebih zat-zat esensial.

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.

2.2.5 Metode Penilaian Gizi Balita

Adapun penjelasan (Supriasa, 2002) mengenai metode penilaian yang digunakan untuk menghitung gizi balita adalah sebagai berikut:

a. Antropometri

Antropometri sebagai indikator status gizi dapat dilakukan dengan mengukur beberapa parameter. Parameter adalah ukuran tunggal dari tubuh manusia, antara lain: umur, berat badan, tinggi badan, lingkaran lengan atas, lingkaran kepala, lingkaran dada, lingkaran panggul dan tebal lemak dibawah kulit. Ukuran tubuh manusia yang berhubungan dengan berbagai macam pengukuran dimensi tubuh dan komposisi tubuh dari berbagai tingkat umur dan tingkat gizi. Penggunaan untuk melihat ketidakseimbangan asupan protein dan energi.

b. Klinis

Metode ini didasarkan atas perubahan-perubahan yang terjadi yang dihubungkan dengan ketidakcukupan gizi. Hal ini dapat dilihat pada jaringan epitel seperti kulit, mata, rambut dan mukosa oral atau pada organ-organ yang dekat dengan permukaan tubuh seperti kelenjar tiroid. Penggunaan untuk survei klinis secara cepat.

c. Biokimia

Pemeriksaan spesimen yang diuji secara laboratorium yang dilakukan pada berbagai macam jaringan tubuh yang digunakan antara lain: darah, urine, tinja dan juga beberapa jaringan tubuh seperti hati dan otot. Penggunaan untuk suatu peringatan bahwa kemungkinan akan terjadi keadaan malnutrisi yang lebih parah lagi.

d. Biofisik

Penentuan status gizi dengan melihat kemampuan fungsi dan melihat perubahan struktur jaringan. Penggunaan dalam situasi tertentu seperti kejadian buta senja epidemic (*epidemic of night blindness*).

e. Survei konsumsi makanan

Metode penentuan gizi secara tidak langsung dengan melihat jumlah dan jenis zat gizi yang dikonsumsi. Penggunaan dengan pengumpulan data konsumsi makanan dapat memberikan gambaran tentang konsumsi berbagai zat gizi pada masyarakat, keluarga dan individu.

f. Statistik vital

Dengan menganalisis data beberapa statistik kesehatan seperti angka kematian berdasarkan umur, angka kesakitan dan kematian akibat penyebab tertentu dan data lainnya yang berhubungan dengan gizi. Penggunaan sebagai bahan indikator tidak langsung pengukuran status gizi masyarakat.

2.3 Pengertian Data Mining

Data mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam *database*. Data mining adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan dan machine learning

untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai *database* besar, (Turban et al, 2005). Menurut Pramudiono Data mining adalah proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara maual dan menganalisis otomatis dari data yang berjumlah besar atau kompleks dengan tujuan menemukan pola tau kecenderungan yang penting yang biasanya tidak didasari keberadaannya, (Pramudiono, 2006).

2.3.1 Klasifikasi Data Mining

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan (Larose, 2005):

a. Deskripsi

Terkadang peneliti dan analis secara sederhana ingin mencoba melakukan percobaan mencari cara untuk menggambar 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 kearah numeric dari pada kearah kategori. Model dibangun menggunakan *record* lengkap yang menyediakan nilai dari variable 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 aka nada 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*, pengamtan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. Kluster adalah 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.

2.4 Pengertian Pohon Keputusan

Pohon keputusan merupakan metode klasifikasi dan prediksi yang sangat kuat dan terkenal. 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, (Berry, 2004).

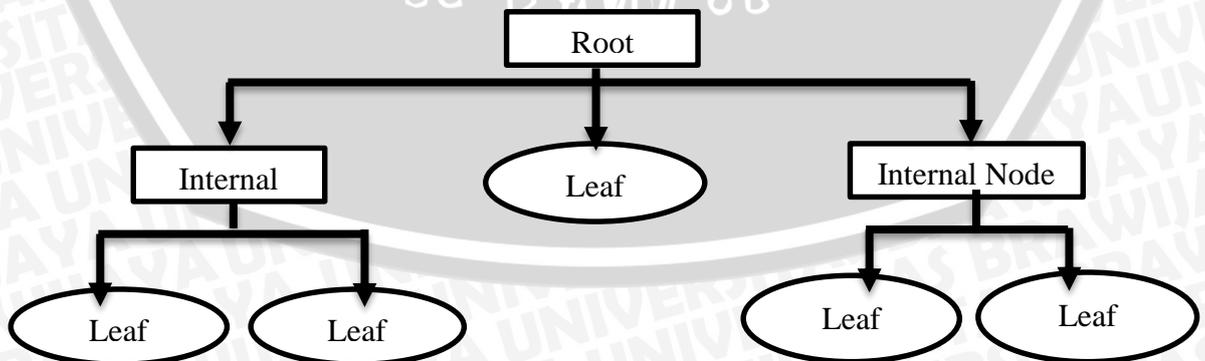
2.4.1 Model Pohon Keputusan

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*.

Berikut ini adalah Gambar 2.1 yang merupakan model pohon keputusan :



Gambar 2. 1 Model Pohon Keputusan

Sumber : (Larose, 2005)

2.4.2 Macam-Macam Algoritma Keputusan

Banyak algoritma yang dapat digunakan dalam pembentukan pohon keputusan antara lain, (Larose, 2005):

a. ID3

Algoritma ID3 atau *Iterative Dichotomiser 3* (ID3) merupakan sebuah metode yang digunakan untuk membuat pohon keputusan. Algoritma pada metode ini menggunakan konsep dari entropi informasi.

b. CART

CART merupakan pohon biner dimana tiap simpul wajib memiliki dua cabang. CART secara rekursif membagi *records* pada data latihan ke dalam subset-subset yang memiliki nilai atribut target (kelas) yang sama. Algoritma CART mengembangkan pohon keputusan dengan memilih percabangan yang paling optimal bagi tiap simpul. Pemilihan dilakukan dengan menghitung segala kemungkinan pada tiap variabel.

c. C4.5

Algoritma C4.5 merupakan pembentukan pohon keputusan (*decision tree*) yang dikembangkan dari algoritma ID3.

2.4.3 Pengertian Algoritma C4.5

Algoritma C.45 adalah salah satu algoritma induksi pohon keputusan yaitu ID3 (*Iterative Dichotomiser 3*). ID3 dikembangkan oleh J. Ross Quinlan. Dalam prosedur algoritma ID3, input berupa sampel *training*, label *training* dan atribut. Algoritma C4.5 merupakan pengembangan dari ID3. Pohon dibangun dengan cara membagi data secara rekursif hingga tiap bagian terdiri dari data yang berasal dari kelas yang sama. Bentuk pemecahan (*split*) yang digunakan untuk membagi data tergantung dari jenis atribut yang digunakan dalam *split*. Algoritma C4.5 dapat menangani data numerik (kontinyu) dan diskret, Marwana (2014). Algoritma C4.5 merupakan algoritma yang digunakan untuk membangun sebuah pohon keputusan (*decision tree*) dari data. Algoritma C4.5 secara rekursif mengunjungi tiap simpul keputusan, memilih percabangan optimal, sampai tidak ada cabang lagi yang mungkin dihasilkan, (Larose, 2005).

2.4.4 Tahap-Tahapan

Membangun klasifikasi dengan *Decision Tree* yang menggunakan Algoritma C4.5, (Larose, 2005) melalui beberapa tahapan sebagai berikut:

- Pertama siapkan data *training* yang biasanya diambil dari data histori atau data masa lampau yang kemudian dibuat ke dalam kelas-kelas tertentu.
- Menghitung nilai *entropy* yang akan digunakan untuk menghitung nilai *gain* dari masing-masing atribut sehingga diperoleh atribut dengan nilai *gain* yang tertinggi yang selanjutnya akan digunakan menjadi akar pohon. Rumus menghitung *gain* dan *entropy* seperti yang ditunjukkan dalam persamaan (2.2) dan (2.3).

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i) \quad (2.2)$$

Keterangan :

S : Himpunan Kasus

A : Atribut

n : Jumlah partisi atribut A

|S_i| : Jumlah kasus pada partisi ke-i

|S| : Jumlah kasus dalam S

Sementara itu, perhitungan nilai *entropy* dapat dilihat pada persamaan (2.3) berikut ini :

$$Entropy(S) = \sum_{i=1}^n -p_i * \log_2 p_i \quad (2.3)$$

Keterangan :

S : Himpunan kasus

A : Fitur

n : Jumlah partisi S

P_i : Proporsi dari S_i terhadap S

- c. Ulangi terus langkah sebelumnya yaitu menghitung nilai tiap atribut berdasarkan nilai *gain* yang tertinggi hingga semua *record*/atribut terpartisi.
- d. Proses dari *Decision Tree* ini akan berhenti jika semua *record* dalam simpul N mendapat kelas yang sama, tidak ada atribut di dalam *record* yang dipartisi lagi, dan tidak ada *record* di dalam cabang yang kosong.
- e. Apabila dalam *tree* terdapat 2 keluaran hasil status klasifikasinya maka diambil probabilitas dari hasil keluaran tersebut nilai tertinggi yang dimiliki dalam kasus tersebut, menggunakan persamaan (2.4) berikut ini:

$$p(c|v) = \frac{1}{T} \sum_t P_t(c|v) \quad (2.4)$$

2.4.5 Kelebihan Algoritma C4.5

Algoritma C4.5 memiliki beberapa kelebihan dibandingkan metode atau algoritma yang lain, yaitu Algoritma C4.5 ini menggunakan konsep *information gain* atau *entropy reduction* untuk memilih percabangan yang optimal. Algoritma C4.5 secara rekursif mengunjungi tiap simpul keputusan, memilih percabangan optimal, sampai tidak ada cabang lagi yang mungkin dihasilkan. Algoritma C4.5 juga dapat mengolah data numerik (kontinyu) dan kategori (diskret), dapat menangani nilai atribut yang hilang dan menghasilkan aturan-aturan yang mudah diinterpretasikan, (Larose, 2005).

2.5 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 (2.5) berikut ini:

$$\text{Akurasi} = \frac{\sum \text{Data Uji Benar Diklasifikasi}}{\sum \text{Data Keseluruhan}} * 100\% \quad (2.5)$$

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, (Kuncoro, 2012).

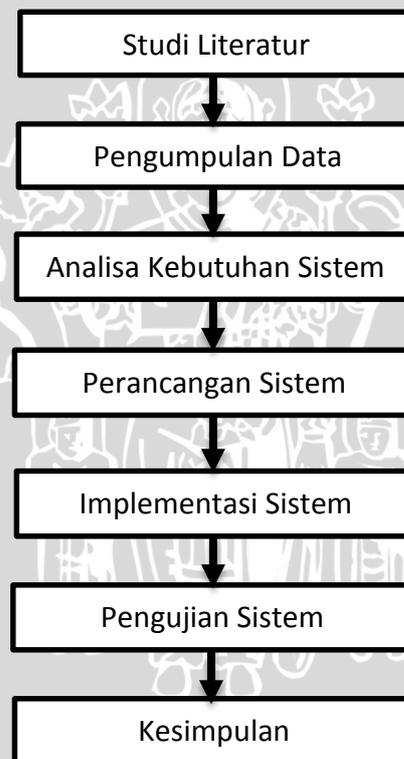


BAB 3 METODOLOGI

Pada bab metodologi ini berisi penjelasan mengenai metode, rancangan yang digunakan, dan langkah-langkah yang dilakukan dalam penelitian dan pembuatan perangkat lunak implementasi sistem penentuan status gizi balita. Metodologi penelitian yang digunakan dalam penelitian ini dibagi dalam beberapa tahapan agar proses penelitian dapat berjalan secara sistematis, sesuai dengan tujuan, dan mempermudah dalam menganalisa permasalahan yang dihadapi.

3.1 Metodologi Penelitian

Pada bab metodologi penelitian dan perancangan akan membahas tentang metode-metode yang akan digunakan dalam Penentuan Status Gizi Balita Pada Kecamatan Kertosono Dengan Menggunakan Algoritma C4.5. Metode penelitian yang diterapkan ditunjukkan pada Gambar 3.1 berikut ini :



Gambar 3. 1 Langkah-langkah penelitian

3.1.1 Studi Literatur

Studi literatur dalam penelitian ini yaitu mempelajari tentang dasar teori yang digunakan untuk menunjang penulisan tugas akhir. Teori-teori pendukung diambil dari berbagai sumber meliputi sumber buku, jurnal, skripsi terdahulu dan *browsing* dari internet dan teori yang digunakan peneliti adalah sebagai berikut :

- Balita
- Gizi serta penilaian status gizi
- Data mining
- Metode Algoritma C4.5
- Perhitungan akurasi
- Data Gizi balita dari Puskesmas Kertosono Kab. Nganjuk

3.1.2 Pengumpulan Data

Data penelitian yang digunakan diperoleh dari data gizi balita di beberapa desa yang berasal dari bidan desa yang bersangkutan yang ada pada wilayah Kertosono, serta data gizi buruk yang diakumulasi satu kecamatan pada Puskesmas Kertosono, data ini lebih spesifik diambil dari 4 desa yang ada di Kertosono, kabupaten Nganjuk. Data latih yang digunakan sejumlah 277 data yang terdiri dari status gizi buruk, gizi kurang, gizi baik dan gizi lebih. Sedangkan untuk pengujian data yang digunakan sejumlah 50 berupa jenis kelamin, umur, berat 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 Kebutuhan Sistem

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

- 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® Core™ i3 @ 2.53 GHz 32 bit
Memory	2GB
Hardisk	500GB
Network	WiFi 802.11n, Bluetooth

- Lingkungan *Software*

Perangkat lunak yang digunakan untuk implementasi penelitian ini adalah :

- Sistem Operasi Windows 8
- Editor Java (Netbean 7.3)
- JDK 7 update 45 32 bit

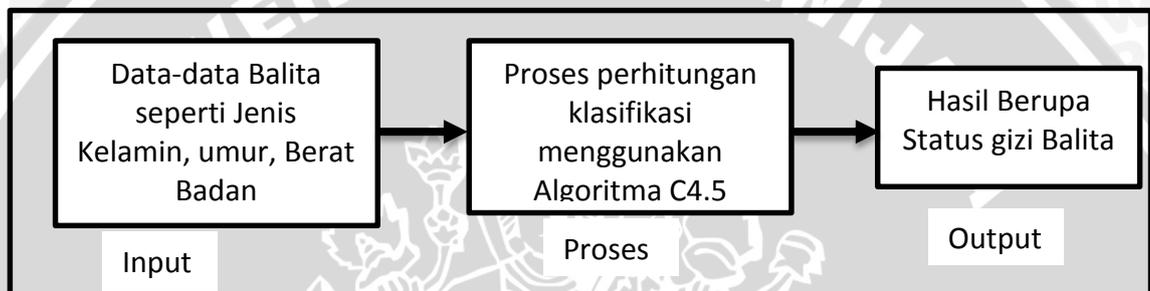
3.1.3.2 Kebutuhan Fungsional

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

3.1.4 Perancangan Sistem

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 Diagram blok sistem pengklasifikasian data balita menggunakan algoritma C4.5

3.1.5 Implementasi

Implementasi ini mengacu pada perancangan sistem yang telah dibentuk. Pada tahap ini menjelaskan tentang penerapan metode (Algoritma C4.5) untuk klasifikasi status gizi balita.

Proses pengimplementasian perangkat lunak ini menggunakan bahasa pemrograman *java* dan Netbeans sebagai *compiler*-nya serta *tools* pendukung lain yang digunakan. Berikut merupakan tahapan yang ada pada implementasi aplikasi:

1. Pembuatan antarmuka.
2. Penerapan metode *Algoritma C4.5* pada sistem untuk proses klasifikasi status gizi balita
3. *Output* yang dihasilkan oleh aplikasi ini nantinya berupa status gizi setiap balita dari perhitungan *Algoritma C4.5*

3.1.6 Pengujian Sistem

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. Hasil pengujian akurasi nantinya ditunjukkan dengan sebuah tabel, yang dicontohkan pada Tabel 3.2.

Tabel 3.2 Tabel pengujian akurasi

No.	Jumlah data latihan	Jumlah data benar	Akurasi

3.1.7 Evaluasi

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.

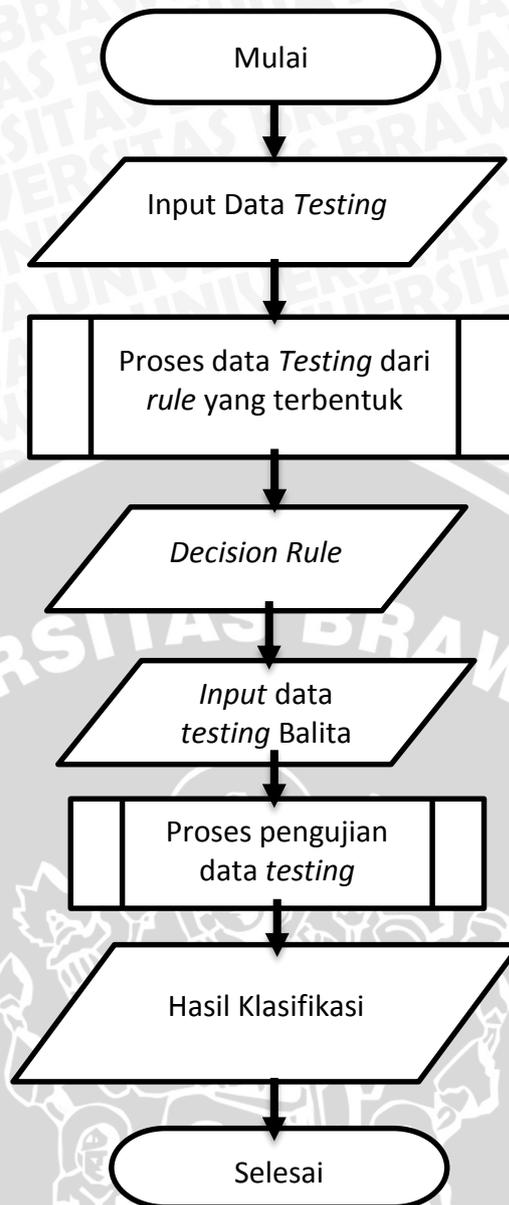
Kesimpulan didapatkan berdasarkan dari hasil pengujian dan analisis terhadap sistem yang dibangun. Penulisan saran berguna untuk memperbaiki kesalahan-kesalahan yang terjadi serta memberikan pertimbangan jika ada pengembangan sistem kelak.

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 masukan yang berupa Jenis Kelamin, Umur, Berat badan. Kemudian memasukkan data balita yang akan diklasifikasikan status gizi nya saja. Pengguna bisa langsung mengetahui balita tersebut termasuk ke golongan status gizi yang ada.

Berdasarkan analisis yang telah dilakukan, secara umum sistem pengklasifikasian data balita ini menggunakan Algoritma C4.5 memiliki beberapa tahap seperti pada gambar 3.3.



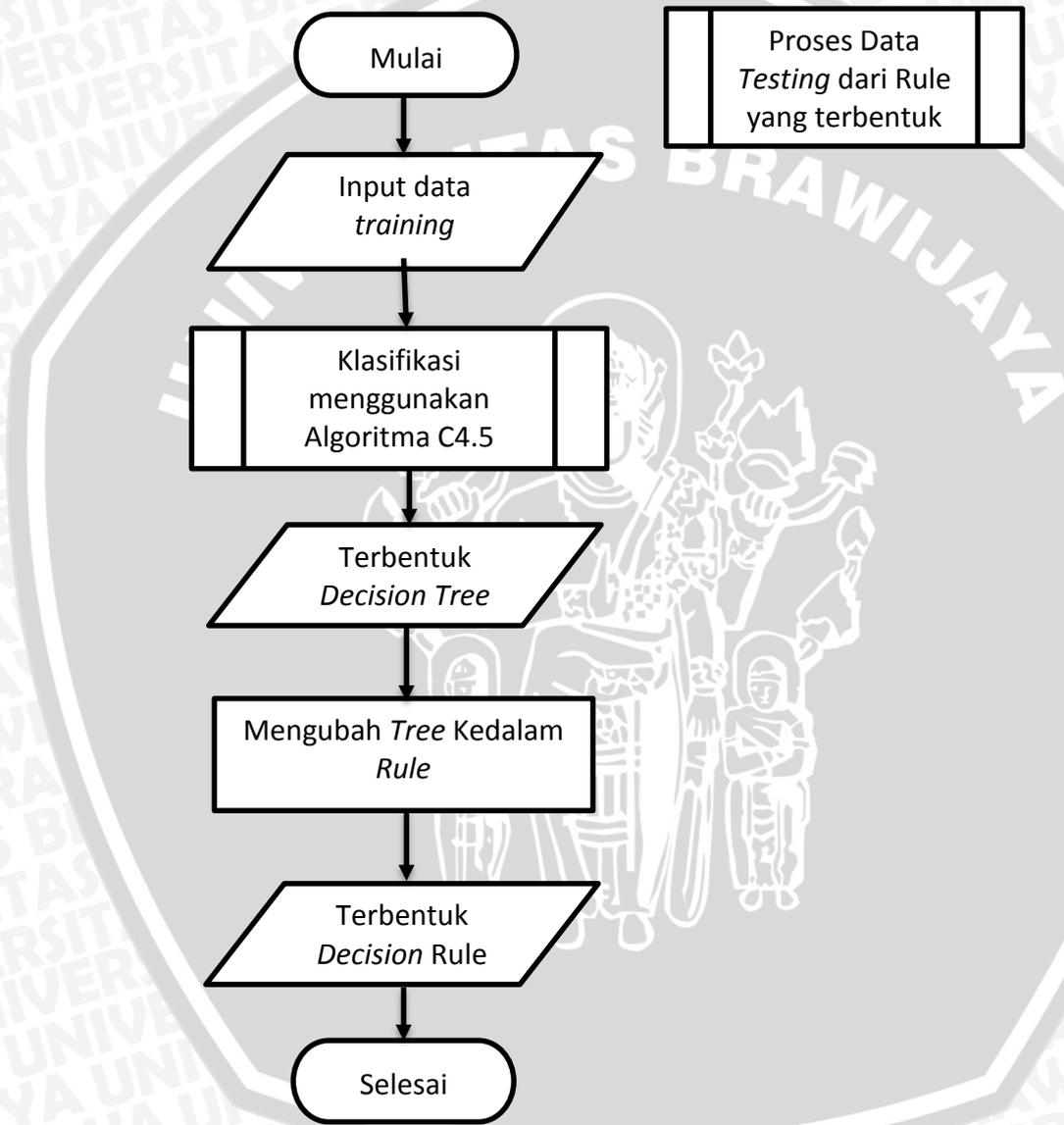
Gambar 3. 3 Diagram alir sistem pengklasifikasian status gizi balita menggunakan Algoritma C4.5

Pada gambar 3.3 diagram alir sistem pengklasifikasian status gizi balita menggunakan Algoritma C4.5, dimulai saat masukan data testing yang berupa jenis kelamin, umur, berat badan kemudian diproses data testing dari *rule* yang terbentuk yang didalamnya menggunakan Algoritma C4.5 dan menghasilkan *decision rule*. Input data *testing* balita selanjutnya disubproses pada pengujian data testing yang menghasilkan klasifikasi gizi balita.

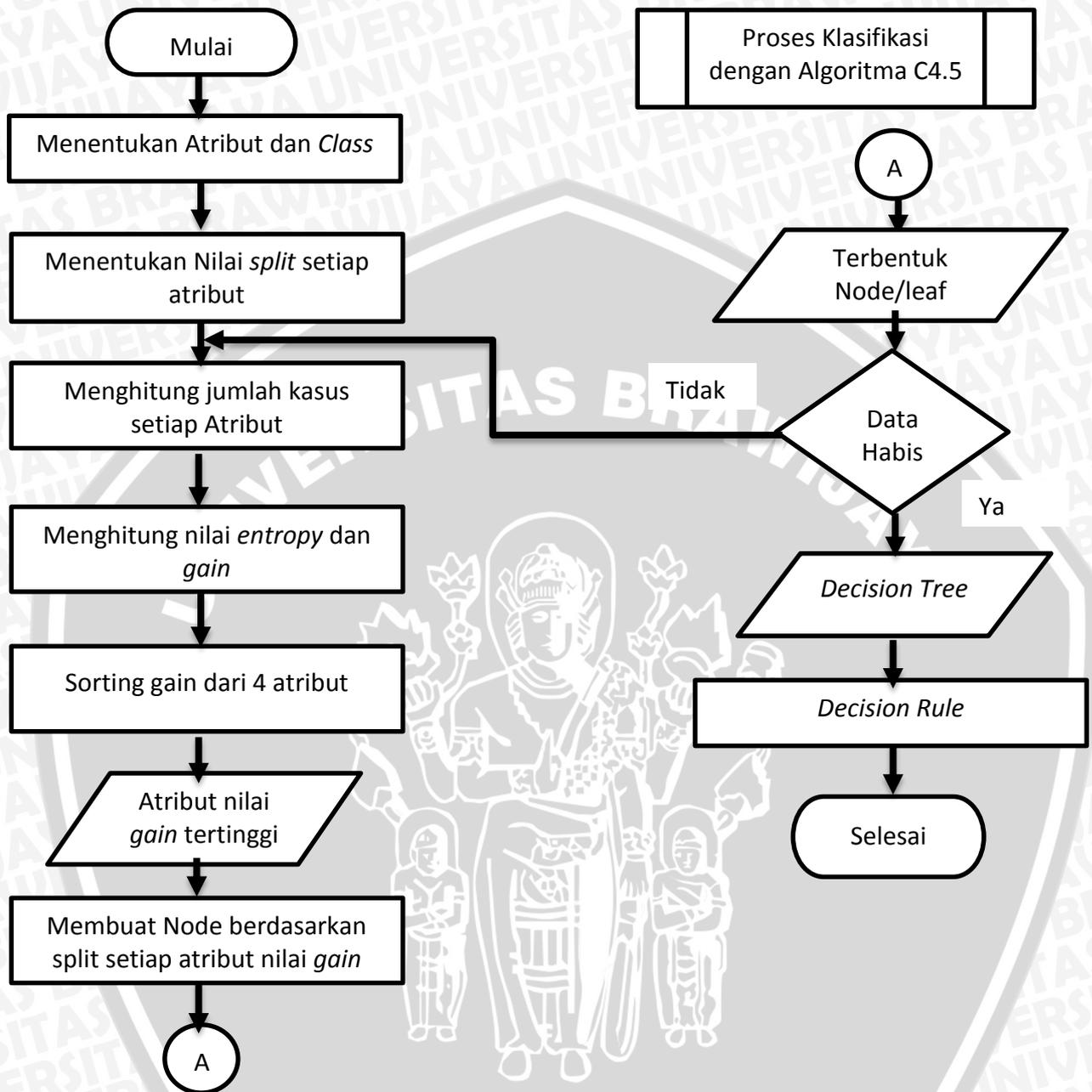
Adapun subproses data testing adalah pada gambar 3.4 sebagai berikut berdasarkan diagram alir sistem:

Untuk input data balita berupa nama, jenis kelamin, umur, berat badan yang harus dimasukkan untuk memperoleh klasifikasi status gizi bayi menggunakan Algoritma C4.5.

Setelah data-data balita yang dibutuhkan selesai dimasukkan, barulah tahapan selanjutnya menghasilkan *decision tree* menggunakan algoritma C4.5 kemudian dari *decision tree* tersebut akan diproses kembali sehingga menghasilkan *decision Rule*. Adapun subproses dari Klasifikasi dengan menggunakan Algoritma C4.5 seperti pada gambar 3.4 berikut ini :



Gambar 3. 4 Gambar diagram alir proses data testing



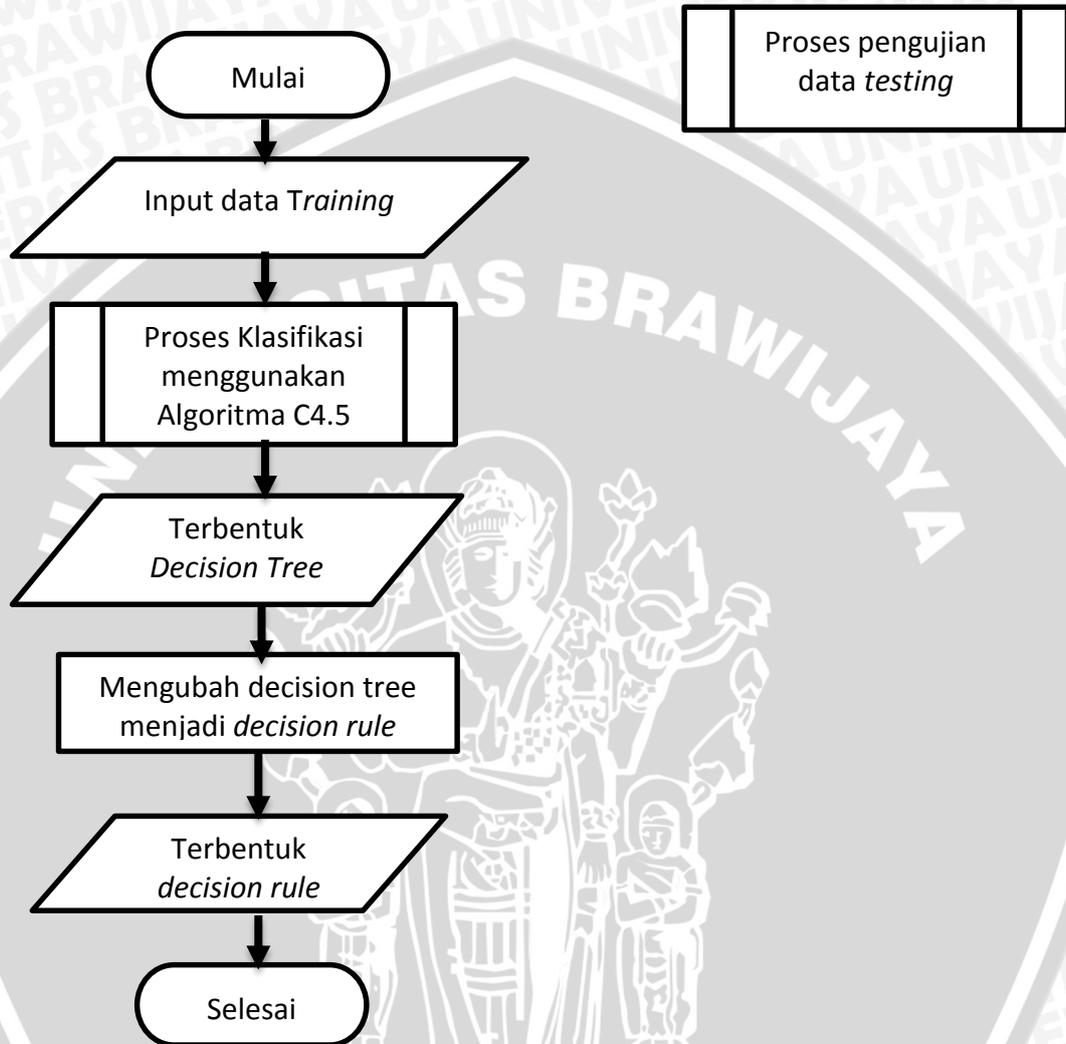
Gambar 3. 5 Gambar diagram alir Algoritma C4.5

Dari Gambar 3.5 di atas menunjukkan tentang langkah-langkah perhitungan klasifikasi menggunakan Algoritma C4.5, diawali dengan menentukan atribut dan *class* yang digunakan dari data. Setelah atribut dan *class* ditentukan selanjutnya menentukan skala yang digunakan pada setiap atribut, kemudian menentukan setiap jumlah kasus pada setiap atribut yang digunakan dan dihitung nilai entropy dan gain.

Setelah melalui perhitungan *entropy* dan *gain*, maka dihasilkanlah nilai gain yang tertinggi. Selanjutnya membuat *Node* berdasarkan *split* setiap atribut

berdasarkan, kemudian proses dilakukan berulang hingga didapatkan *decision tree* untuk keseluruhan data. Langkah terakhir setelah semua data masuk dalam *decision tree* yaitu membuat *decision Rule*-nya.

Adapun proses data latih berfungsi sebagai data pelatihan pada system ini. Subproses data latih ditunjukkan pada gambar 3.6 berikut ini:



Gambar 3. 6 Diagram alir proses data pengujian data *testing*

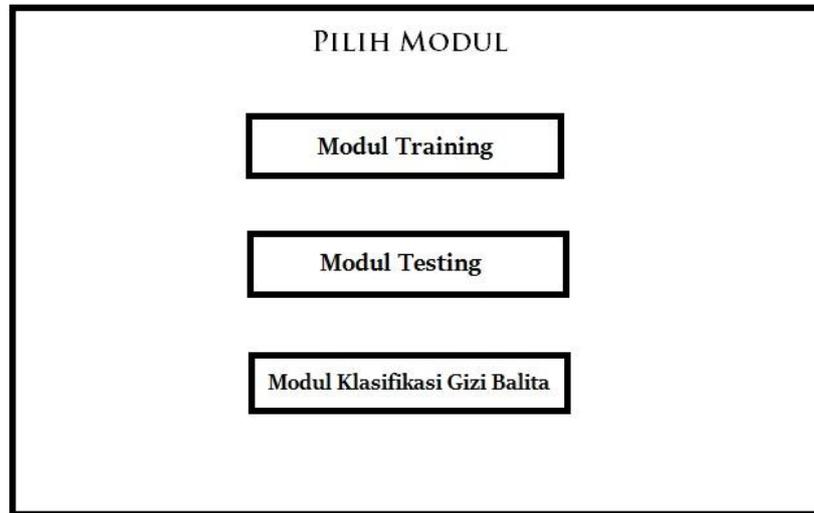
3.3 Perancangan Antarmuka

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 diantaranya:

- Tampilan Awal
- Modul Data *training*
- Modul Data *testing*
- Modul Klasifikasi

3.3.1 Tampilan Awal

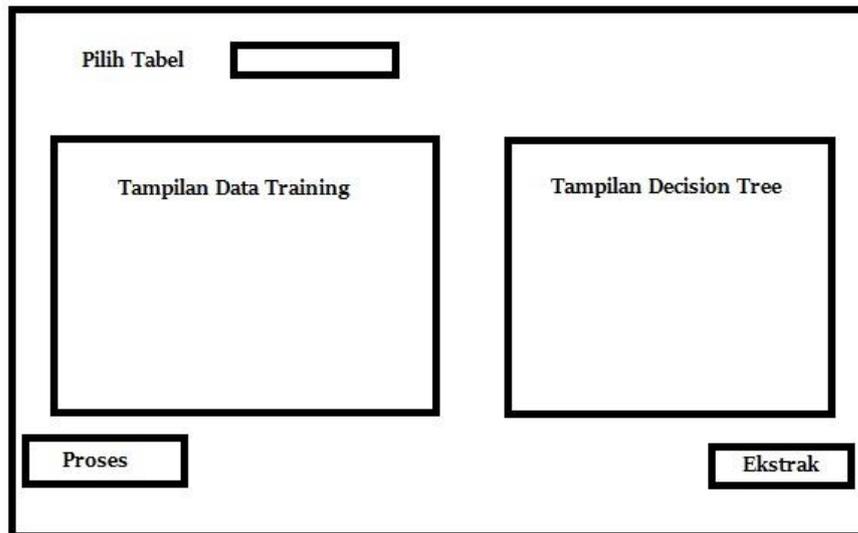
Tampilan awal adalah tampilan pembuka yang terdiri dari 3 modul pilihan yaitu modul *training*, modul *testing* dan modul klasifikasi gizi balita. Adapun tujuannya, tiap modul dipilih sesuai dengan kebutuhan pengguna.



Gambar 3. 7 Tampilan awal

3.3.2 Modul Data Training

Antarmuka untuk modul data latih berupa tabel data latih dan kolom *decision rule*. Rule dari Algoritma C4.5 ini diproses untuk menghasilkan data latih. Untuk rancangan antarmuka modul data latih ditunjukkan pada Gambar 3.9 berikut ini :



Gambar 3. 8 Modul data training

3.3.3 Modul Data testing

Untuk melakukan pengujian dilakukan pada modul data *testing* disini. Pengguna nantinya akan memilih tabel data balita yang akan di uji untuk menentukan klasifikasi dan akurasinya. Pada Gambar 3.9, merupakan perancangan antarmuka modul data *testing*:

The interface for the 'Modul Data testing' consists of a main container with three elements: a label 'Pilih Tabel' followed by a text input field, a large rectangular area labeled 'Tampilan Tabel Data Testing', and a button labeled 'Proses' located at the bottom right.

Gambar 3. 9 Antarmuka modul data *testing*

3.3.4 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 balita. Hasil dari masukan tersebut berupa klasifikasi status gizi balita menggunakan Algoritma C4.5. Adapun status gizi balita berupa gizi baik, gizi buruk, gizi kurang, gizi lebih. Modul klasifikasi ditunjukkan oleh Gambar 3.10 di bawah ini :

The interface for 'Klasifikasi Status Gizi Balita Menggunakan Algoritma C4.5' features a title at the top. Below it, the text 'Inputkan Data Balita :' is followed by four input fields labeled 'Nama', 'JenisKelamin', 'Umur', and 'Berat Badan'. A 'Proses Data' button is positioned below these fields. To the right, the text 'Hasil Klasifikasi' is followed by a large empty rectangular box for the output.

Gambar 3. 10 Antar muka klasifikasi status gizi balita

3.4 Perhitungan Manual

Perhitungan manual ini menjelaskan tentang semua proses dari data yang sudah direkap dengan narasumber pukesmas kertoso serta masukan sampai hasil keluaran pengolahan data menggunakan Algoritma C4.5 ini. Dalam perhitungan manual dengan menggunakan Algoritma C4.5 ini peneliti menggunakan 17 data dengan keluaran klasifikasi gizi balita yang berkategori gizi baik, gizi lebih, gizi kurang dan gizi buruk. Adapun rincian data yang digunakan dalam proses penentuan gizi balita dalam perhitungan manual ini dapat ditunjukkan pada Tabel 3.3 berikut ini :

Tabel 3.3 Data Latih pada Perhitungan Manual

No	NAMA BALITA	Jenis Kelamin	Umur (Bln)	BB (Kg)	Status
1	Nurfarhan	L	7	7.2	Gizi Baik
2	MOH. NAFAL A.	L	9	9	Gizi Baik
3	Imeida Reva P.	P	31	10.9	Gizi Baik
4	SIFAATUL. K	P	19	12	Gizi Baik
5	FAJAR ISMU K.	L	30	13	Gizi Baik
6	M.Izan Aufa F.	P	28	15.5	Gizi Baik
7	MAHARDIKA. P	L	18	7.6	Gizi Buruk
8	M. RIFKI M	L	58	11.5	Gizi Buruk
9	NADILA SYAFA	P	59	11.6	Gizi Buruk
10	FLORA DWIYANTI	P	17	6.7	Gizi Buruk
11	FITRIANA	P	26	9	Gizi Kurang
12	SITI MASRUOH	P	40	11	Gizi Kurang
13	M. OKTA FADHIL	L	15	7.5	Gizi Kurang
14	SUSI SARIDEWI	P	5	10	Gizi Lebih
15	Siti Aminah	P	31	10.9	Gizi Lebih
16	Aizin Azna	L	23	17	Gizi Lebih
17	ALEA	P	57	29	Gizi Lebih

Dalam Tabel 3.3, akan dibuat pohon keputusan yang nantinya akan digunakan sebagai penentuan status gizi balita dengan melihat data yang digunakan dalam perhitungan menggunakan Algoritma C4.5 ini, meliputi jenis kelamin, umur, berat badan saja. Kemudian tahap pertama setelah data latih sudah ditentukan yaitu :

- Menentukan *split* dari setiap atribut dan kelas yang digunakan dalam klasifikasi status gizi balita ini, *split* yang digunakan untuk atribut Berat Badan, Umur. Berdasarkan tabel standart Berat badan ideal balita dibawah ini yang ditunjukkan pada tabel 3.4:

Tabel 3.4 Berat badan dan tinggi badan ideal balita

Umur (bulan)	Berat (gram)		Tinggi (cm)	
	Standar	80%standar	Standar	80%standar
Lahir	3,4	2,7	50,5	40,5
0-1	4,3	3,4	55,6	43,5
2	5,0	4,0	58	46
3	3,7	4,5	60	48
4	6,3	5,0	62,5	49,5
5	6,9	5,5	64,5	51
6	7,4	5,9	66	52,5
7	8	6,3	67,5	54
8	8,4	6,7	69	55,5
9	8,9	7,1	70	56,5
10	9,3	7,4	72	57,5
11	9,6	7,7	73	58,5
12	9,9	7,9	74,5	60
1 tahun 3 bulan	10,6	8,5	78	62,5
6 bulan	11,3	9	81,5	65
9 bulan	11,9	9,6	84,5	67,5
2 tahun 0 bulan	12,4	9,9	87	69,5
3 bulan	12,9	10,5	89,5	71,5
6 bulan	13,5	10,8	92	73,5
9 bulan	14	11,2	94	75
3 tahun 0 bulan	14,4	11,6	96	77
3 bulan	15	12	98	78,5
6 bulan	13,5	12,4	99,5	79,5
9 bulan	16	12,9	101,5	81,5
4 tahun 0 bulan	16,5	13,2	103,5	82,5
3 bulan	17	13,6	105	83,5
6 bulan	17,4	14	107	85,5
9 bulan	17,9	14,4	108	86,5
5 tahun 0 bulan	18,4	14,7	109	87

Sumber : Departemen kesehatan RI

Dari tabel diatas dapat ditentukan oleh penulis atribut serta *split* adalah sebagai berikut :

- Jenis Kelamin (JK) :
 1. Laki-Laki
 2. Perempuan
- Umur (tahun) :
 1. 0 – 12
 2. 13 – 24
 3. 25 – 36
 4. 37 – 48
 5. 49 – 60
- Berat Badan (Kg):
 1. 0 – 9,9
 2. 10 – 14
 3. 14,5 – 16,5
 4. >16,5

b. Menghitung jumlah kasus, jumlah kasus dari setiap kategori status gizi balita yaitu gizi baik, gizi lebih, gizi kurang dan gizi buruk dari seluruh data latih yang sudah ada pada Tabel 3.3. Hasil dari perhitungan jumlah kasusnya dapat ditunjukkan pada Tabel 3.5 berikut ini :

Tabel 3.5 Perhitungan Jumlah Kasus dari Data Latih

			jumlah kasus	Gizi buruk	Gizi kurang	Gizi baik	Gizi lebih
	total		17	4	3	6	4
	jk	l	7	2	1	3	1
		p	10	2	2	3	3
	umur	0-12	3	0	0	2	1
		13-24	5	2	1	1	1
		25-36	5	0	1	3	1
		37-48	1	0	1	0	0
		49-60	3	2	0	0	1
	BB	0-9.9	6	2	2	2	0
		10-14	8	2	1	4	1
		14.5-16.5	1	0	0	1	0
		>16.5	2	0	0	0	2

c. Langkah selanjutnya menghitung nilai *entropy* yang akan digunakan untuk menghitung nilai *gain* dari masing-masing atribut sehingga diperoleh atribut dengan nilai *gain* yang tertinggi yang selanjutnya akan digunakan menjadi akar pohon. Rumus menghitung *gain* dan *entropy* seperti yang ditunjukkan pada rumus berikut ini :

$$Entropy (S) = \sum_{i=1}^n -pi * \log_2 pi$$



Keterangan :

- S : Himpunan kasus
- A : Fitur
- n : Jumlah partisi S
- P_i : Proporsi dari S_i terhadap S

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i)$$

Keterangan :

- S : Himpunan Kasus
- A : Atribut
- n : Jumlah partisi atribut A
- |S_i| : Jumlah kasus pada partisi ke-i
- |S| : Jumlah kasus dalam S

Dari tabel 3.2 dapat dilihat dari jumlah kasus ada sebanyak 26 dengan rincian sebagai berikut pada gizi buruk sebanyak 6 balita, gizi kurang 5 balita, dan gizi baik 9 balita. Setelah semuanya di hitung jumlah setiap kasus nya dari data latih yang digunakan selanjutnya menghitung nilai entropy. Maka nilai entropy totalnya adalah sebagai berikut ini :

$$Entropy = \left(- \frac{Gizi\ Buruk}{Jumlah\ kasus} * \log_2 \left(\frac{Gizi\ Buruk}{Jumlah\ kasus} \right) \right) + \left(- \frac{Gizi\ Kurang}{Jumlah\ kasus} * \log_2 \left(\frac{Gizi\ Kurang}{Jumlah\ kasus} \right) \right) + \left(- \frac{Gizi\ Baik}{Jumlah\ kasus} * \log_2 \left(\frac{Gizi\ Baik}{Jumlah\ kasus} \right) \right) + \left(- \frac{Gizi\ Lebih}{Jumlah\ kasus} * \log_2 \left(\frac{Gizi\ Lebih}{Jumlah\ kasus} \right) \right)$$

Sehingga langsung saja memasukkan nilai datanya yang sudah diklasifikasikan ke setiap kasus berikut ini :

$$Entropy(total) = \left(- \frac{4}{17} * \log_2 \left(\frac{4}{17} \right) \right) + \left(- \frac{3}{17} * \log_2 \left(\frac{3}{17} \right) \right) + \left(- \frac{6}{17} * \log_2 \left(\frac{6}{17} \right) \right) + \left(- \frac{4}{17} * \log_2 \left(\frac{4}{17} \right) \right) = 1,95425$$

Menghitung Entropy dan gain Setiap Atribut, misalnya menghitung atribut Jenis Kelamin.

Jenis Kelamin (Laki-Laki) :

$$Entropy(Jenis\ Kelamin - Laki - laki) = \left(- \frac{2}{7} * \log_2 \left(\frac{2}{7} \right) \right) + \left(- \frac{1}{7} * \log_2 \left(\frac{1}{7} \right) \right) + \left(- \frac{3}{7} * \log_2 \left(\frac{3}{7} \right) \right) + \left(- \frac{1}{7} * \log_2 \left(\frac{1}{7} \right) \right) = 1,84237$$

Jenis Kelamin Perempuan :

$$Entropy(Jenis\ Kelamin - Perempuan) = \left(- \frac{2}{10} * \log_2 \left(\frac{2}{10} \right) \right) + \left(- \frac{2}{10} * \log_2 \left(\frac{2}{10} \right) \right) + \left(- \frac{3}{10} * \log_2 \left(\frac{3}{10} \right) \right) + \left(- \frac{3}{10} * \log_2 \left(\frac{3}{10} \right) \right) = 1,97095$$

Menghitung gain Atribut Jenis Kelamin :

$$Gain = Total\ Entropy - \left(\frac{Jumlah\ Kasus\ JK(L)}{Total\ Jumlah\ Kasus} * Entropy\ JK(L) \right) + \left(\frac{Jumlah\ Kasus\ JK(P)}{Total\ Jumlah\ Kasus} * Entropy\ JK(P) \right)$$



Masukkan rumus ke dalam nilai entropy yang sudah didapatkan diatas:

$$Gain JK = 1,95425 - \left(\frac{7}{17} * 1,84237\right) + \left(\frac{10}{17} * 1,97095\right) = 0,03624$$

Dalam Setiap 4 atribut yang ada yaitu jenis kelamin, umur, BB, dan TB perhitungan entropy dan gain menggunakan rumus yang sama seperti contoh diatas sehingga akan menghasilkan *entropy* dan *gain* pada setiap atributnya seperti tabel 3.6 berikut ini :

Tabel 3.6 Hasil Perhitungan Node 1 Nilai Entropy dan Gain

		jumlah kasus	Gizi buruk	Gizi kurang	Gizi baik	Gizi lebih	entropy	gain
total		17	4	3	6	4	1.954247	
jk	l	7	2	1	3	1	1.842371	0.036241
	p	10	2	2	3	3	1.970951	
umur	0-12	3	0	0	2	1	0.918296	0.661649
	13-24	5	2	1	1	1	1.921928	
	25-36	5	0	1	3	1	1.370951	
	37-48	1	0	1	0	0	0	
	49-60	3	2	0	0	1	0.918296	
BB	0-9.9	6	2	2	2	0	1.584963	0.571319
	10-14	8	2	1	4	1	1.75	
	14.5-16.5	1	0	0	1	0	0	
	>16.5	2	0	0	0	2	0	

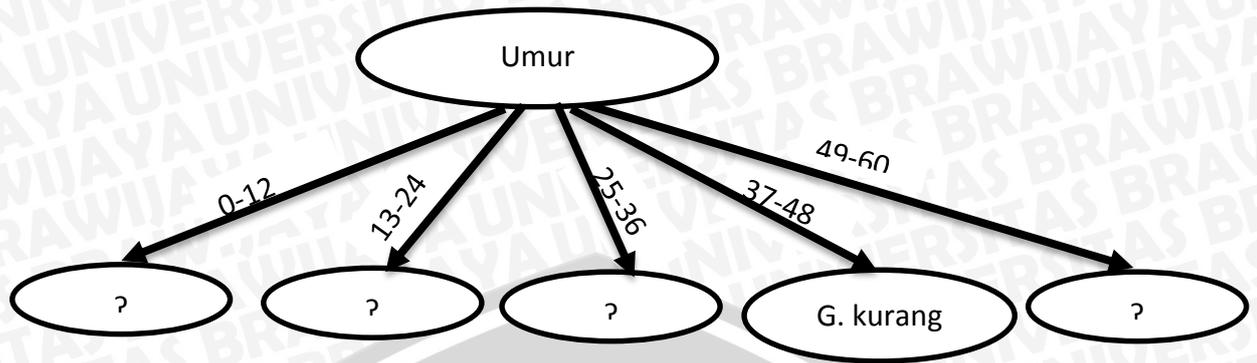
d. Menentukan *Decision Tree*

Keterangan istilah pada *Decision tree* diantaranya adalah sebagai berikut ini:

- G. Baik : Gizi Baik
- G. Kurang : gizi kurang
- G. Lebih : Gizi lebih
- G. buruk : gizi Lebih
- JK : Jenis kelamin
- BB: Berat Badan

Langkah awal menentukan atribut mana yang akan menjadi *node* akar, *node* akar ditentukan berdasarkan nilai *gain* yang tertinggi. Pada tabel 3.8 perhitungan nilai *entropy* dan *gain* dapat diketahui bahwa nilai *gain* yang tertinggi adalah atribut umur.

Setelah ditentukan *Node* serta *split-nya* yaitu rentang 0-12, 13-24, 25-36, 37-48, 49-60. Sehingga akan menghasilkan Node 1 sebagai berikut ini :



Pada rentang 37 – 48 tahun sudah dapat diketahui hasilnya yaitu masuk pada gizi kurang karena pada rentang ini memiliki balita sebanyak 1 dan pada gizi lebih gizi buruk dan gizi baik pada berat badan rentang 37 – 48 bernilai 0 (nol) sehingga dapat langsung disimpulkan bahwa berat badan rentang 37 – 48 masuk pada kategori gizi buruk. Pada tabel 3.6 diketahui bahwa umur rentang 0 – 12 tahun tidak dapat langsung diklasifikasikan masuk pada kategori gizi apa, sebab pada rentang ini setiap kategori gizi mempunyai nilai, begitu pula pada rentang umur 13 – 24 dan 25 – 36 serta rentang 49 – 60 tahun. Sehingga diperlukan lagi perhitungan *entropy* dan *gain* pada ke-empat rentang tersebut, yang nantinya nilai *gain* tertinggi akan menjadi *node* cabang.

Untuk perhitungan *entropy* dan *gainnya* menggunakan rumus yang sama seperti perhitungan atribut jenis kelamin yang sudah dipaparkan di atas pada tabel 3.6. Dan catatan yang paling penting adalah setiap atribut yang sudah menjadi *root* (umur) tidak perlu dihitung kembali. Adapun hasil setelah melewati perhitungan *entropy* dan *gain* akan ditemukan node 1.1 umur 0 – 12 tahun adalah sebagai berikut :

Tabel 3.7 Hasil node cabang 1.1

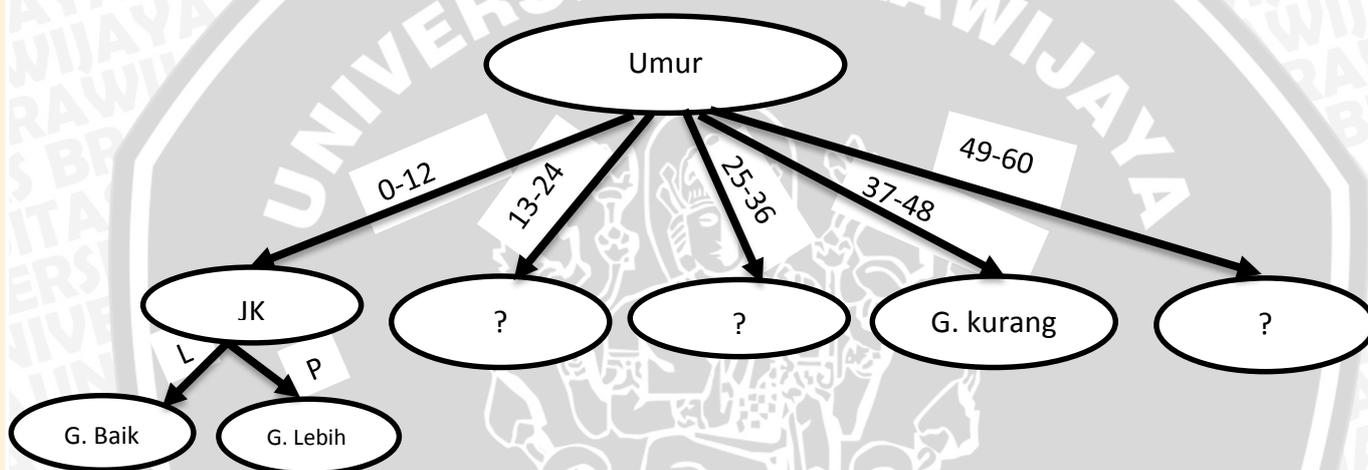
node UMUR 0-12			jumlah kasus	Gizi buruk	Gizi kurang	Gizi baik	Gizi lebih	entropy	gain
1	total		3	0	0	2	1	0.918296	
	jk	l	2	0	0	2	0	0	0.918296
		p	1	0	0	0	1	0	
	BB	0-9.9	2	0	0	2	0	0	0.918296
		10-14	1	0	0	0	1	0	
		14.5-16.5	0	0	0	0	0	0	
		>16.5	0	0	0	0	0	0	

Diambil dari 17 data pada tabel 3.5 diatas yang mempunyai umur 0 – 12 tahun sehingga didapatkan data pada tabel 3.10 berikut ini :

Tabel 3.8 Data Node 1.1 balita umur 0 -12 tahun

No	NAMA BALITA	Jenis Kelamin	Umur	BB	Status
			(Bln)	(Kg)	
1	Nurfarhan	L	7	7.2	Gizi Baik
2	MOH. NAFAL A.	L	9	9	Gizi Baik
14	SUSI SARIDEWI	P	5	10	Gizi Lebih

Berdasarkan tabel 3.7 diatas akan diperoleh *gain* tertinggi yaitu Jenis Kelamin dan Berat badan karena nilai *gainnya* sama boleh dipilih salah satu saja,yang akan dijadikan *node* cabang 1.1 dan Jenis Kelamin dipilih untuk *root* selanjutnya, sehingga hasil *decision tree* sebagai berikut :



Selanjutnya setelah diperoleh *node* cabang 1.1 umur 0 – 12 adalah JK kemudian pada atribut JK ini memiliki 2 kategori yaitu Laki-laki (L) dan Perempuan (P). Pada tabel 3.7 sudah dapat diketahui hasil status gizi balitanya, JK laki-laki memiliki 2 orang balita status gizi baik dan JK perempuan memiliki 1 orang balita dengan status gizi lebih. Sehingga pada kasus ini tidak perlu lagi menghitung nilai *entropy* dan *gain* karena hasilnya sudah dapat diketahui.

Selanjutnya menghitung *node* 1.2 yaitu umur rentang 13 – 24 tahun dengan cara menghitung *entropy* dan *gain* sama seperti cara yang diatas sebelumnya. Berikut adalah data balita dari 17 data yang dijadikan perhitungan manual adalah yang ditunjukkan pada tabel 3.9 dibawah ini :

Tabel 3.9 Data Balita Node 1.2 Umur 13 – 24 Tahun

No	NAMA BALITA	Jenis Kelamin	Umur	BB	Status
			(Bln)	(Kg)	
13	M. OKTA FADHIL	L	15	7.5	Gizi Kurang
10	FLORA DWIYANTI	P	17	6.7	Gizi Buruk

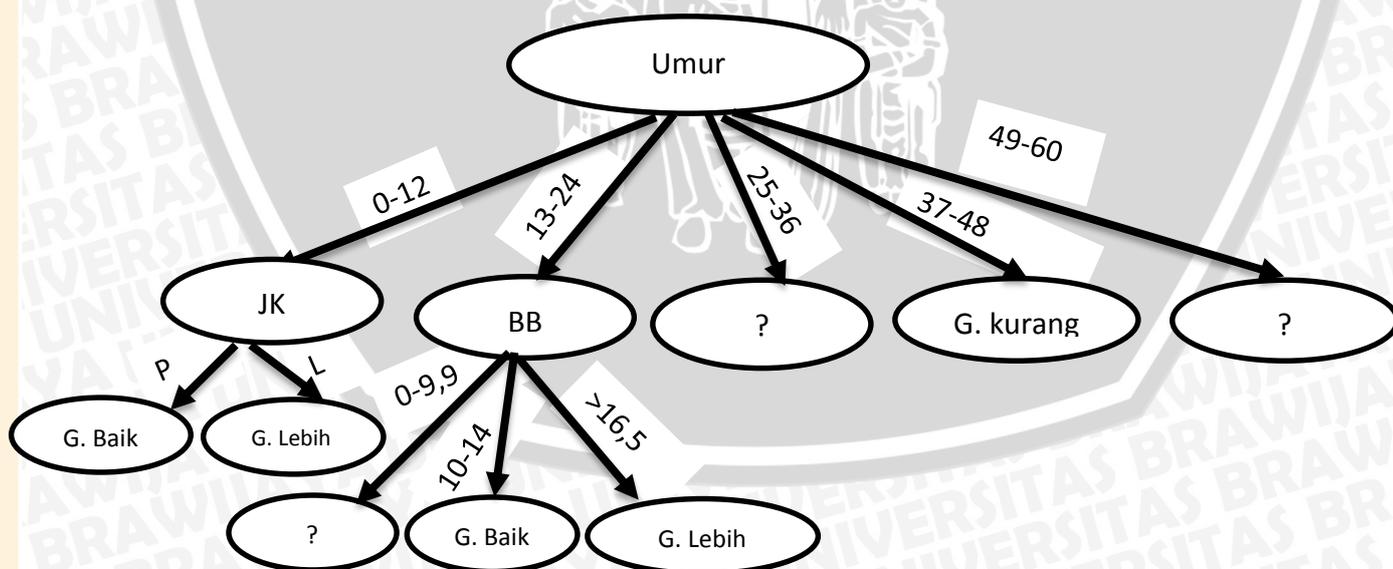
7	MAHARDIKA PUTRA RAMADANI	L	18	7.6	Gizi Buruk
4	SIFAATUL KHASANAH	P	19	12	Gizi Baik
16	Aizin Azna	L	23	17	Gizi Lebih

Dan berikut ini adalah perhitungan *Entropy* dan *gainnya*, yang ditunjukkan pada tabel 3.10 dibawah ini :

Tabel 3.10 Data Balita Node 1.2 Umur 13 – 24 Tahun

node	UMUR		jumlah kasus	Gizi buruk	Gizi kurang	Gizi baik	Gizi lebih	entropy	gain
1	total		5	2	1	1	1	1.921928	
	jk	l	3	1	1	0	1	1.584963	0.570951
		p	2	1	0	1	0	1	
	BB	0-9.9	3	2	1	0	0	0.918296	1.370951
		10-14	1	0	0	1	0	0	
		14.5-16.5	0	0	0	0	0	0	
		>16.5	1	0	0	0	1	0	

Dari perhitungan *entropy* dan *gain* umur rentang 13 – 24 tahun diperoleh nilai *gain* yang tertinggi adalah BB, atribut tersebut yang digunakan sebagai *node* 1.2, sehingga penulis memilih BB. Berikut ini adalah tampilan *decision tree* dari node 1.2:



Pada kasus ini Balita umur 13 – 24 tahun dengan berat badan 0 – 9,9 kg masih belum menemukan status gizi balitanya, sebab dalam berat badan 0 – 9,9 kg memiliki 2 balita gizi buruk dan 1 balita gizi kurang. Sehingga masih perlu dihitung *entropy* dan *gain* lagi. Berikut tabel 3.13 menunjukkan data balita dengan umur 13 – 24 tahun dengan berat badan 0 – 9,9 kg :

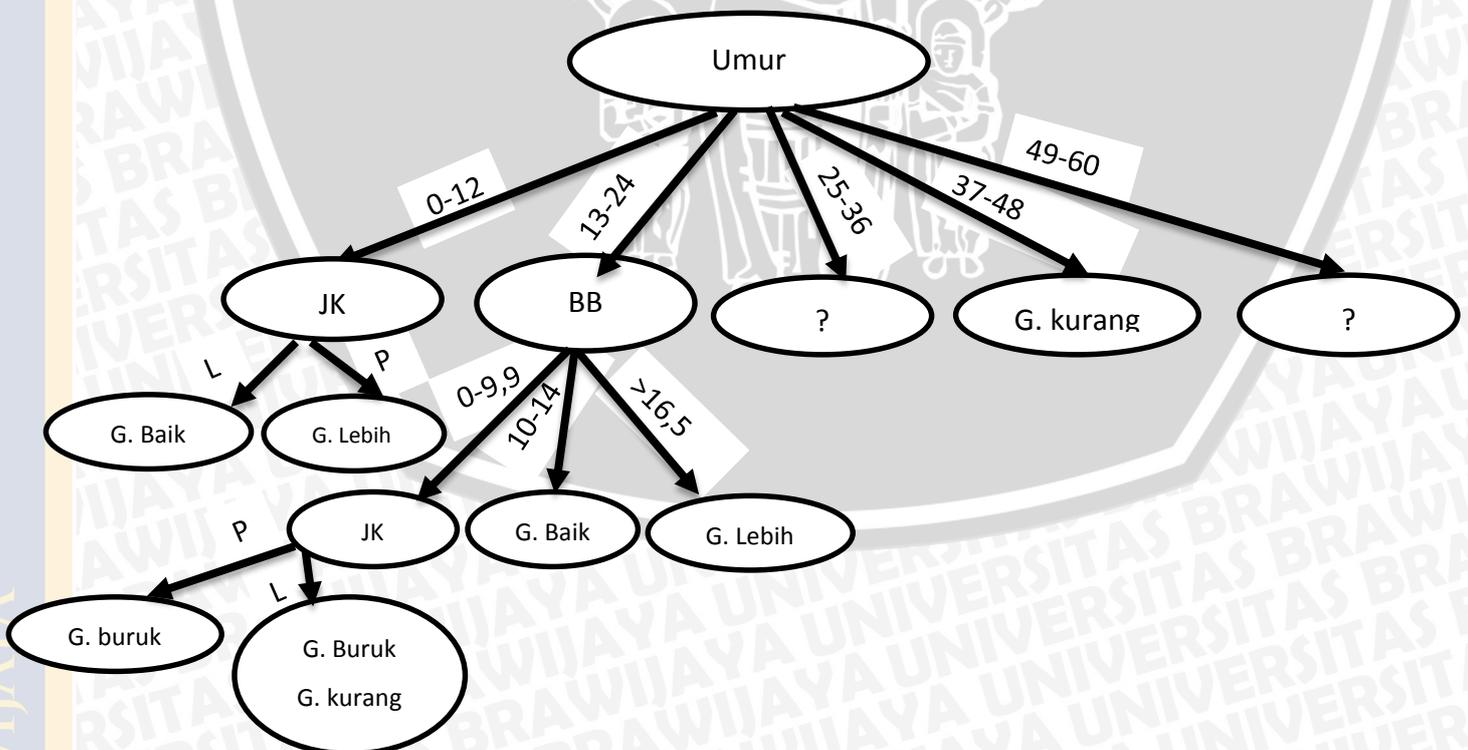
Tabel 3.11 Data balita node 1.2.1 umur 13 – 24 tahun dan BB 0 – 9,9 kg

No	NAMA BALITA	Jenis kelamin	Umur	BB	Status
			(Bln)	(Kg)	
13	M. OKTA FADHIL	L	15	7.5	Gizi Kurang
10	FLORA DWIYANTI	P	17	6.7	Gizi Buruk
7	MAHARDIKA PUTRA RAMADANI	L	18	7.6	Gizi Buruk

Tabel 3.12 Hasil node 1. 2.1

node bb			jumlah kasus	Gizi buruk	Gizi kurang	Gizi baik	Gizi lebih	entropy	gain
1	total		3	2	1	0	0	0.918296	
	jk	l	2	1	1	0	0	1	0.251629
		p	1	1	0	0	0	0	

Dari hasil perhitungan *entropy* dan *gain* di atas maka akan diperoleh *tree* dari node 1.2.1 adalah sebagai berikut :



Pada node 1.2.1 ini atribut jenis kelamin memiliki 2 keluaran status gizi balita yaitu buruk dan kurang yang biasa disebut *multiclass* sehingga untuk menentukan keluar yang sesuai maka diperlukan perhitungan probabilitas gizi buruk dan gizi kurang pada 17 data yang digunakan sebagai perhitungan manual tersebut, sehingga nantinya probabilitas yang paling tinggilah yang dijadikan keluar tersebut.

- Probabilitas Gizi Buruk

$$P(\text{G.Buruk}) = \frac{\text{banyaknya gizi buruk}}{\text{jumlah data}} = \frac{4}{17}$$

$$P(\text{C}|\text{G.Buruk}) = \frac{P(\text{C} \wedge \text{G.Buruk})}{P(\text{G.buruk})} = \frac{1}{\frac{4}{17}} = \frac{17}{4}$$

Sehingga diperoleh probabilitas gizi buruknya adalah sebesar $\frac{17}{4}$.

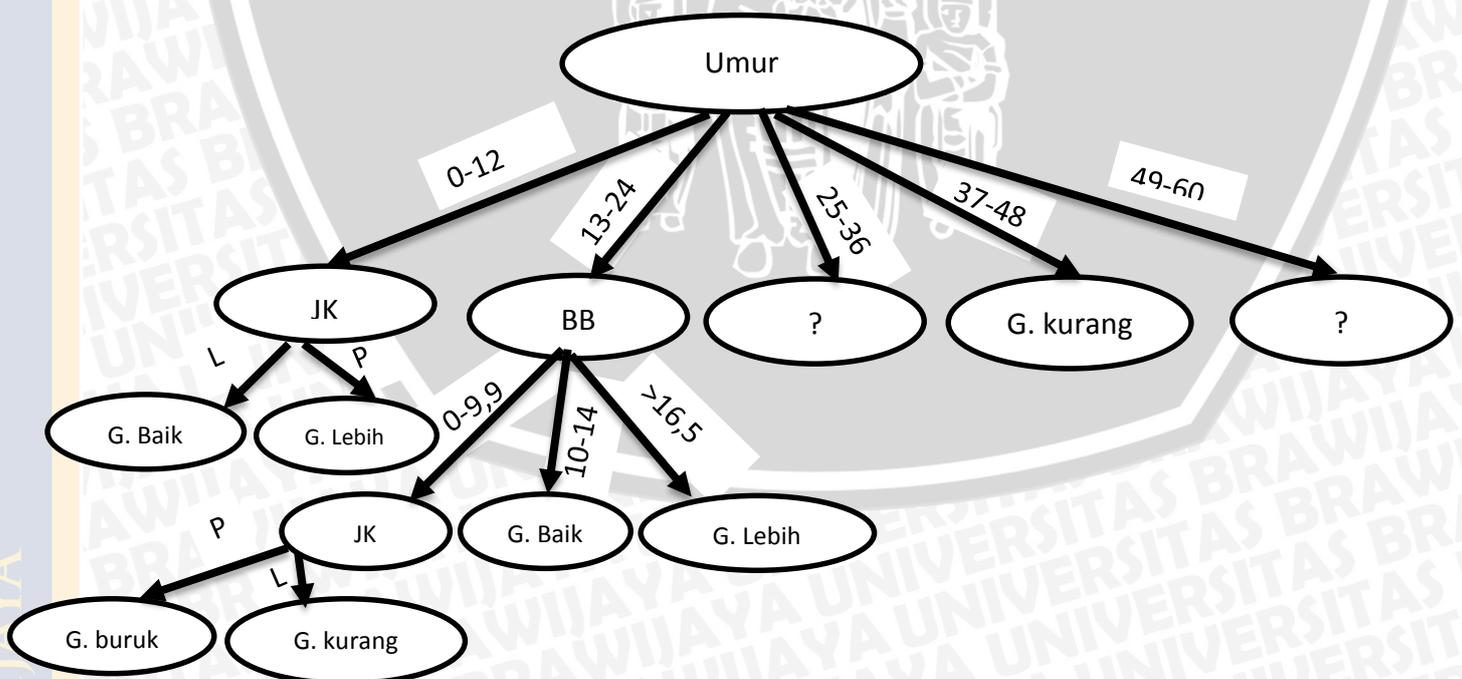
- Probabilitas Gizi Kurang

$$P(\text{G.Kurang}) = \frac{\text{banyaknya gizi kurang}}{\text{jumlah data}} = \frac{3}{17}$$

$$P(\text{C}|\text{G.Kurang}) = \frac{P(\text{C} \wedge \text{G.kurang})}{P(\text{G.kurang})} = \frac{1}{\frac{3}{17}} = \frac{17}{3}$$

Sehingga diperoleh probabilitas gizi kurang sebesar $\frac{17}{3}$

Dari kedua probabilitas gizi buruk dan gizi kurang nilai yang paling tinggi adalah gizi kurang. Selanjutnya gizi kurang akan dijadikan keluaran dri JK Laki-laki yang akan menghasilkan *tree* berikut ini :



Selanjutnya perhitungan *Node* 1.3 dengan atribut umur 25 – 36 tahun dengan data 17 akan didapatkan hasil data Tabel 3.13 dan hasil perhitungan *entropy* serta *gain* tabel 3.14 berikut ini :

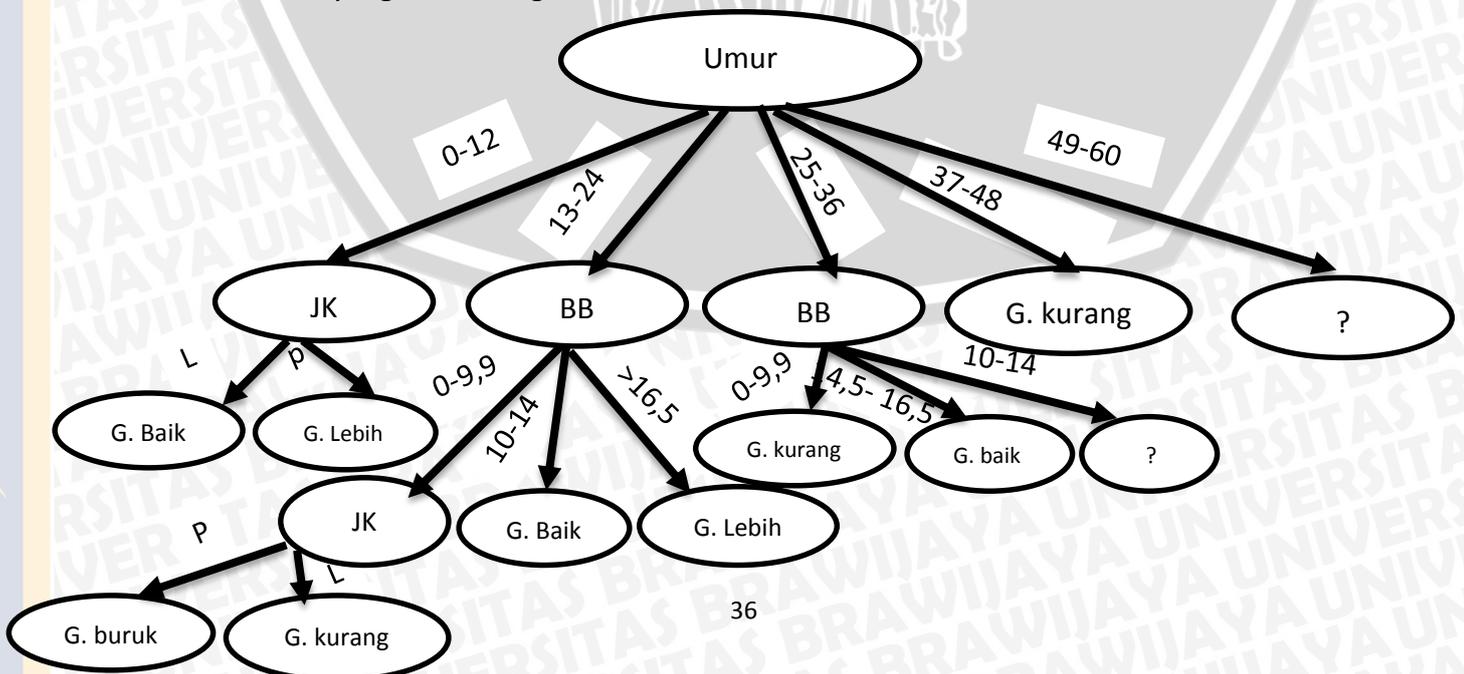
Tabel 3.13 Data balita node 1.3 umur 25 – 36 tahun

No	NAMA BALITA	Jenis Kelamin	Umur (Bln)	BB (Kg)	Status
11	FITRIANA	P	26	9	Gizi Kurang
6	M.Izan Aufa F.	P	28	15.5	Gizi Baik
5	FAJAR ISMU K.	L	30	13	Gizi Baik
3	Imeida Reva P.	P	31	10.9	Gizi Baik
15	Siti Aminah	P	31	10.9	Gizi Lebih

Tabel 3.14 Data balita node 1.3 umur 25 – 36 tahun

node	UMUR	jumlah kasus	Gizi buruk	Gizi kurang	Gizi baik	Gizi lebih	entropy	gain
1	total	5	0	1	3	1	1.370951	
	jk	l	1	0	1	0	0	0.170951
		p	4	1	2	1	1.5	
	BB	0-9.9	1	1	0	0	0	0.819973
		10-14	3	0	2	1	0.918296	
		14.5-16.5	1	0	1	0	0	
		>16.5	0	0	0	0	0	

Sesudah dihasilkan nilai *entropy* dan nilai *gain* tertinggi yang akan dijadikan *node* 1.3 yang akan menghasilkan *tree* berikut ini :



Selanjutnya perhitungan *Node* 1.3.1 dengan atribut umur 25 – 36 tahun dengan BB 10 – 14 kg dengan data 17 akan didapatkan hasil data Tabel 3.15 dan hasil perhitungan *entropy* serta *gain* tabel 3.16 berikut ini :

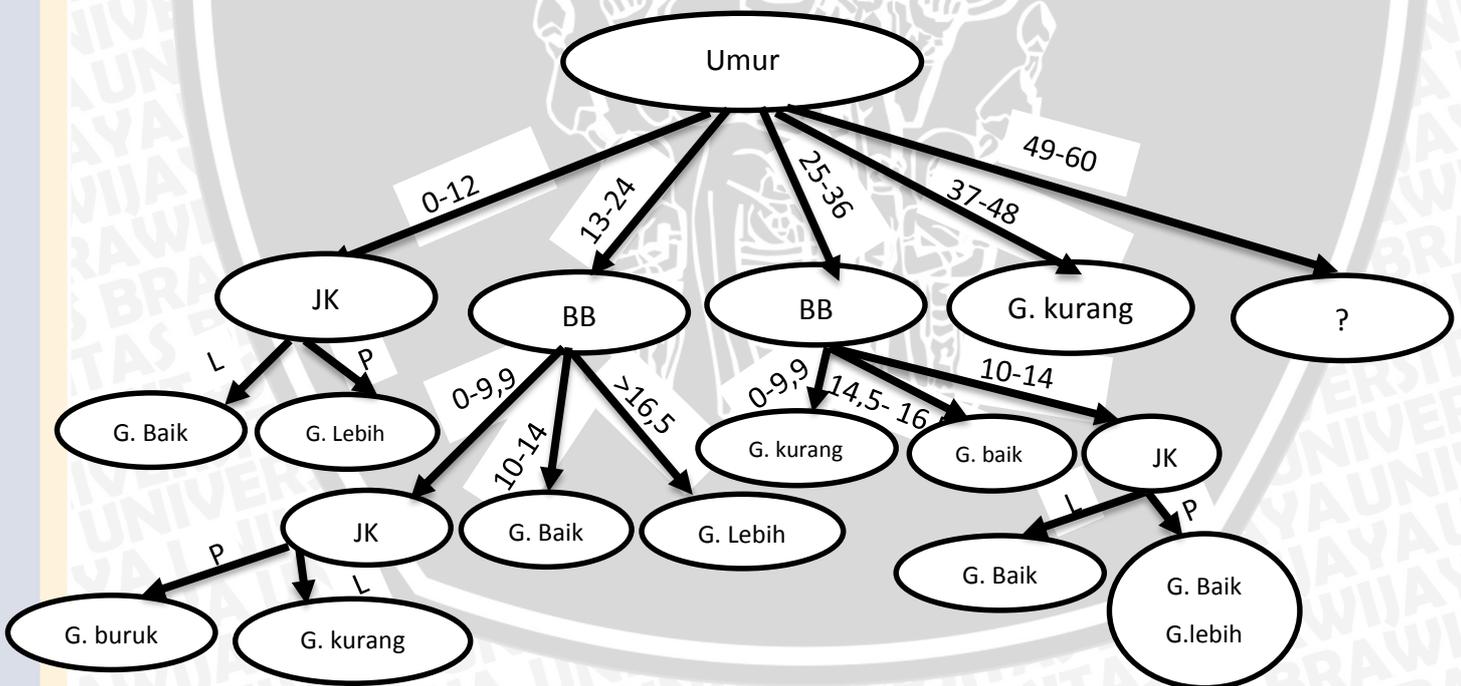
Tabel 3.15 Data balita node 1.3.1

No	NAMA BALITA	Jenis Kelamin	Umur	BB	Status
			(Bln)	(Kg)	
5	FAJAR ISMU K.	L	30	13	Gizi Baik
3	Imeida Reva P.	P	31	10.9	Gizi Baik
15	Siti Aminah	P	31	10.9	Gizi Lebih

Tabel 3.16 Perhitungan entropy dan gain node 1.3.1

node	BB 10-14		jumlah kasus	Gizi buruk	Gizi kurang	Gizi baik	Gizi lebih	entropy	gain
1	total		3	0	0	2	1	0.918296	
	jk	l	1	0	0	1	0		0.251629
		p	2	0	0	1	1		

Sehingga akan menghasilkan *Tree* pada Node 1.3.1 adalah sebagai berikut:



Pada node 1.3.1 ini atribut jenis kelamin memiliki 2 keluaran status gizi balita yaitu baik dan lebih yang biasa disebut *multiclass* sehingga untuk menentukan keluar yang sesuai maka diperlukan perhitungan probabilitas gizi baik dan gizi lebih pada 17 data yang digunakan sebagai perhitungan manual tersebut, sehingga nantinya probabilitas yang paling tinggilah yang dijadikan keluar tersebut.

- Probabilitas Gizi Baik

$$P(G.Buruk) = \frac{\text{banyaknya gizi baik}}{\text{jumlah data}} = \frac{6}{17}$$

$$P(C|G.Buruk) = \frac{P(C \cap G.Buruk)}{P(G.buruk)} = \frac{1}{\frac{6}{17}} = \frac{17}{6}$$

Sehingga diperoleh probabilitas gizi buruknya adalah sebesar $\frac{17}{6}$.

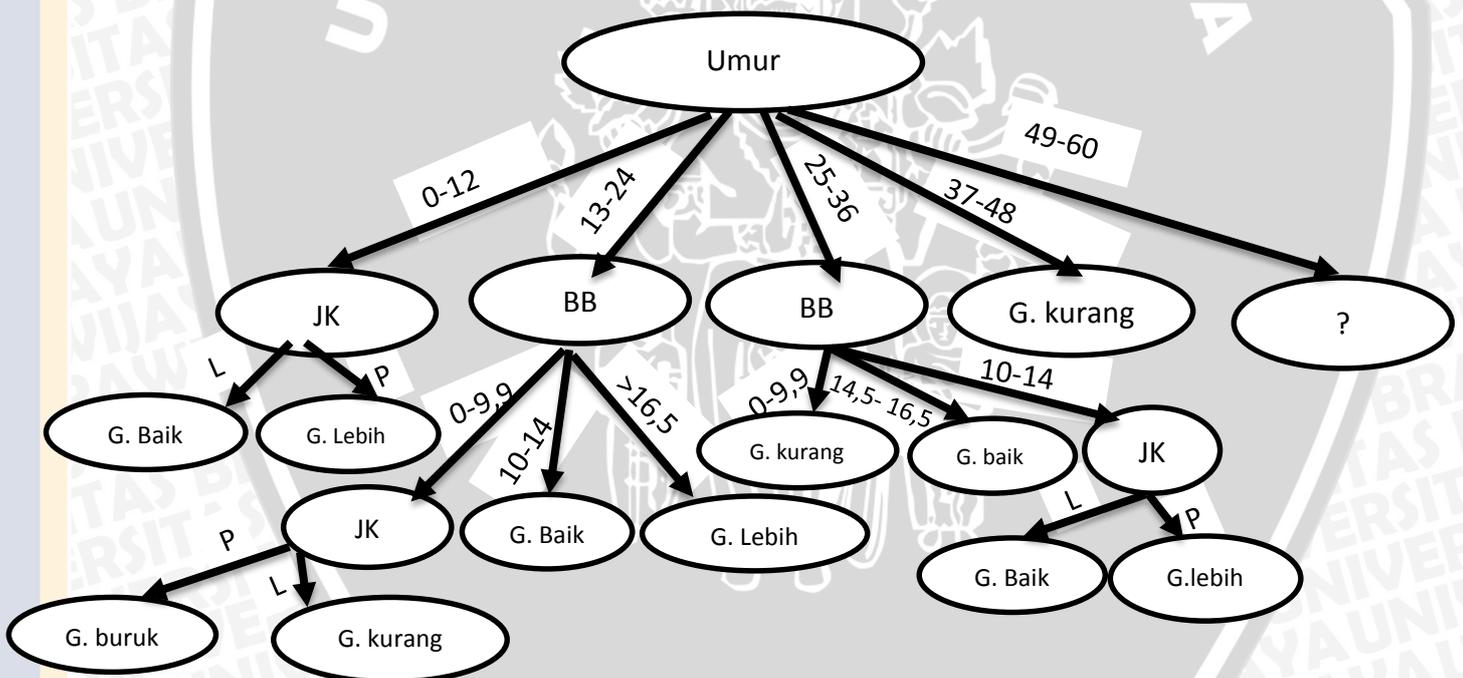
- Probabilitas Gizi Lebih

$$P(G.Kurang) = \frac{\text{banyaknya gizi lebih}}{\text{jumlah data}} = \frac{4}{17}$$

$$P(C|G.Kurang) = \frac{P(C \cap G.kurang)}{P(G.kurang)} = \frac{1}{\frac{4}{17}} = \frac{17}{4}$$

Sehingga diperoleh probabilitas gizi kurang sebesar $\frac{17}{4}$

Dari kedua probabilitas gizi baik dan gizi lebih nilai yang paling tinggi adalah gizi lebih. Selanjutnya gizi lebih akan dijadikan keluaran dri JK perempuan yang akan menghasilkan *tree* berikut ini :



Selanjutnya perhitungan *Node* 1.5 dengan atribut umur 49 – 60 tahun dengan data 17 akan didapatkan hasil data Tabel 3.17 dan hasil perhitungan *entropy* serta *gain* tabel 3.18 berikut ini:

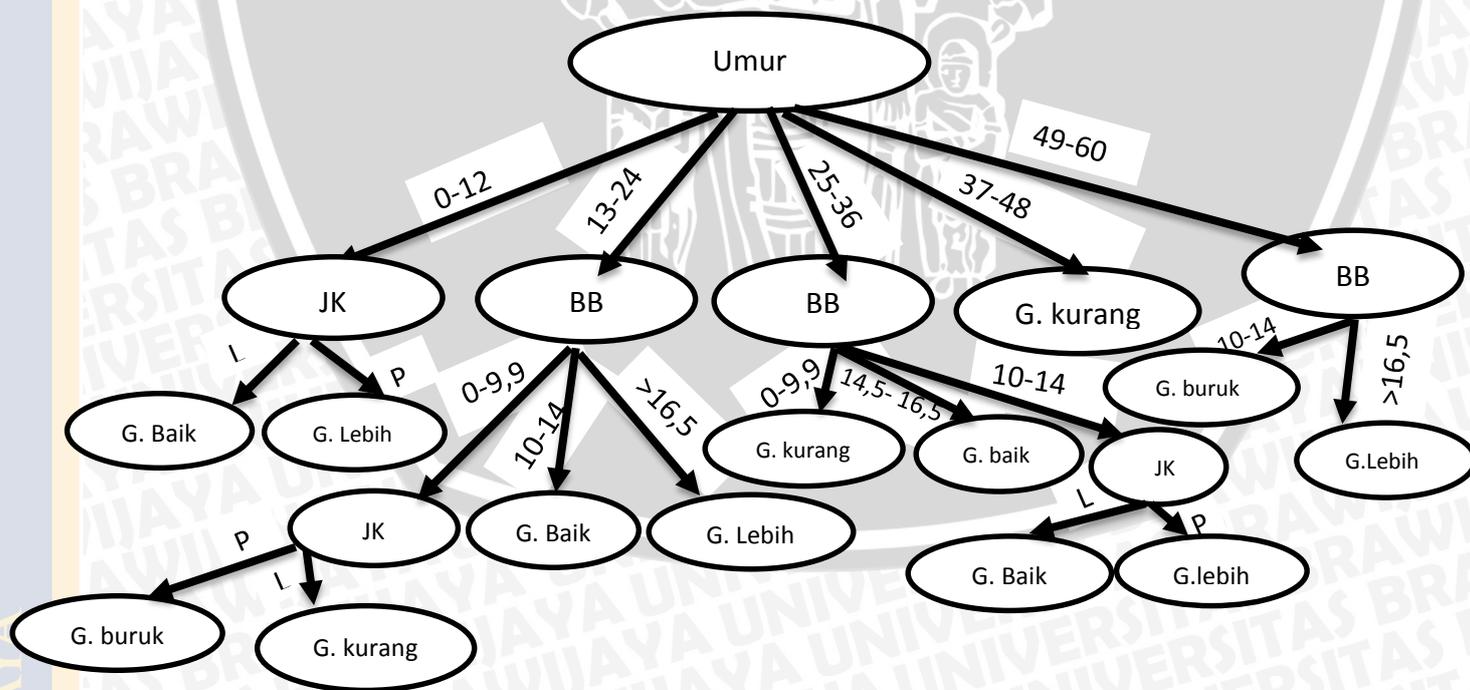
Tabel 3.17 Data balita node 1.5

No	NAMA BALITA	Jenis Kelamin	Umur	BB	Status
			(Bln)	(Kg)	
17	ALEA	P	57	29	Gizi Lebih
8	M. RIFKI M	L	58	11.5	Gizi Buruk
9	NADILA SYAFA	P	59	11.6	Gizi Buruk

Tabel 3.18 Perhitungan entropy dan gain node 1.5

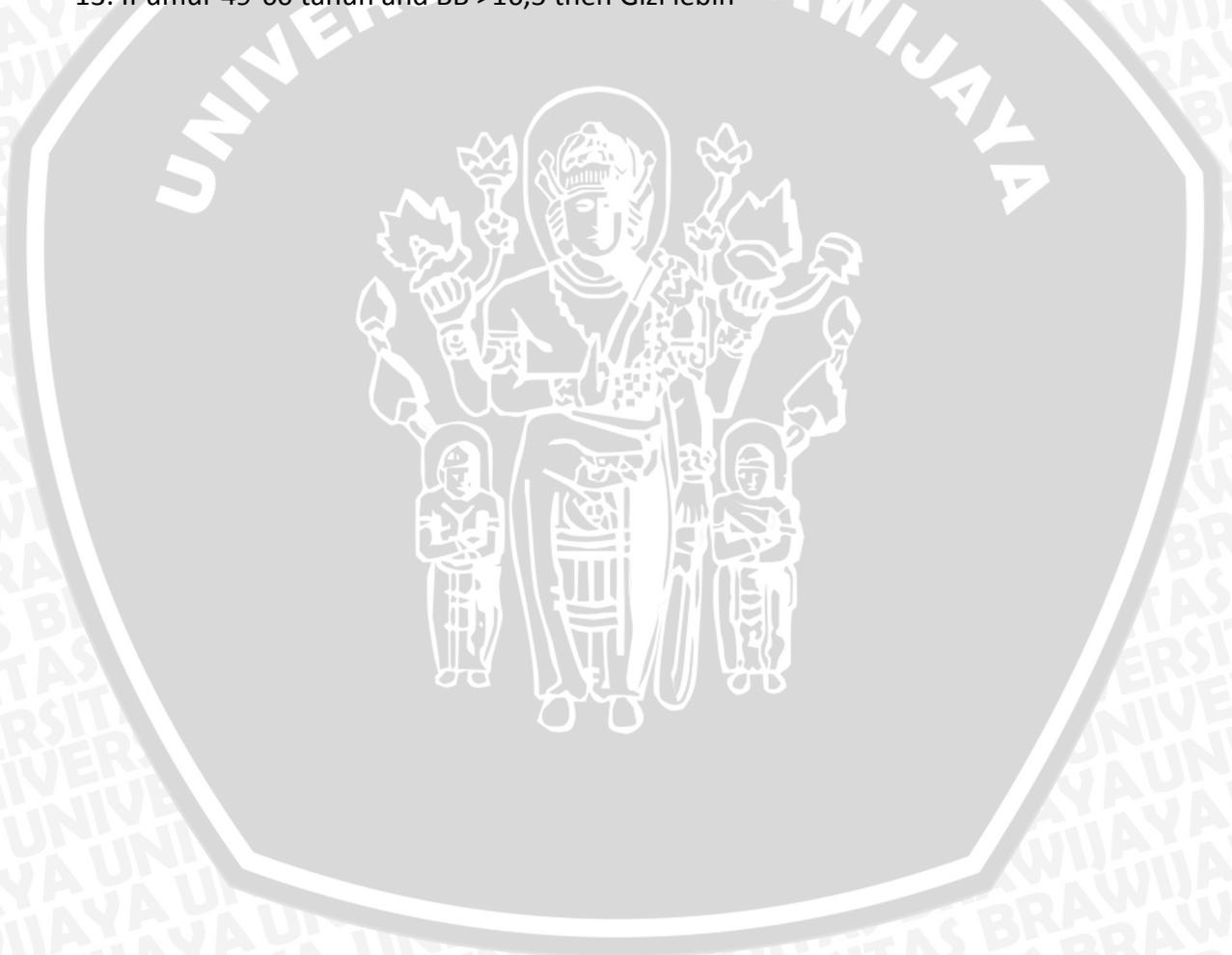
node	UMUR		jumlah kasus	Gizi buruk	Gizi kurang	Gizi baik	Gizi lebih	entropy	gain
1	total		3	2	0	0	1	0.918296	
	jk	l	1	1	0	0	0	0	0.584963
		p	2	1	0	0	1	0.5	
	BB	0-9.9	0	0	0	0	0	0	0.918296
		10-14	2	2	0	0	0	0	
		14.5-16.5	0	0	0	0	0	0	
		>16.5	1	0	0	0	1	0	

Berikut ini hasil dari keseluruhan tree yang diambil dari 17 data perhitungan manual:



Dari decision tree yang sudah ada dapat dibuat decision rule sebagai berikut :

1. If umur = 0-12 tahun and JK laki-laki then Gizi baik
2. If umur = 0-12 tahun and JK perempuan then Gizi lebih
3. If umur 13 – 24 tahun and BB 0 – 9,9 and JK Perempuan then gizi buruk
4. If umur 13 – 24 tahun and BB 0 – 9,9 and JK Laki-laki then gizi kurang
5. If umur 13 – 24 tahun and BB 10 – 14 then gizi baik
6. If umur 13 – 24 tahun and BB >16,5 then gizi lebih
7. If umur 25 – 36 tahun and BB 0 – 9,9 then gizi kurang
8. If umur 25 – 36 tahun and BB 14,5 – 16,5 then gizi baik
9. If umur 25 – 36 tahun and BB 10 – 14 and JK laki-laki then gizi baik
10. If umur 25 – 36 tahun and BB 10 – 14 and JK perempuan then gizi lebih
11. If umur 37 – 48 tahun then gizi kurang
12. If umur 49-60 tahun and BB 10-14 then Gizi buruk
13. If umur 49-60 tahun and BB >16,5 then Gizi lebih



BAB 4 IMPLEMENTASI

Pada Bab ini menjelaskan mengenai hasil implementasi dari algoritma yang digunakan, yaitu algoritma C4.5 ke dalam bentuk program, serta menjelaskan tiap fungsi halaman pada program yang dibuat.

4.1 Lingkungan Implementasi

Lingkungan implementasi yang digunakan untuk penelitian ini terdiri dari 2 lingkungan, yaitu lingkungan perangkat keras (*hardware*) dan lingkungan perangkat lunak (*software*).

4.1.1 Lingkungan *Hardware*

Perangkat keras (*hardware*) yang digunakan dalam implementasi penelitian ini adalah sebagai berikut:

- Jenis : *Notebook*
- Prosesor : Intel® Core™ i3 @ 2.53 GHz 32 bit
- Memory : 2GB
- Hardisk : 500GB
- Network : WiFi 802.11n, Bluetooth

4.1.2 Lingkungan *Software*

Adapun perangkat lunak yang digunakan dalam implementasi penelitian ini adalah sebagai berikut ini:

- Editor Java (Netbean 7.3)
- JDK 7 update 10 32 bit
- Microsoft Office
- Sistem Operasi Windows 8

4.2 Persiapan Data

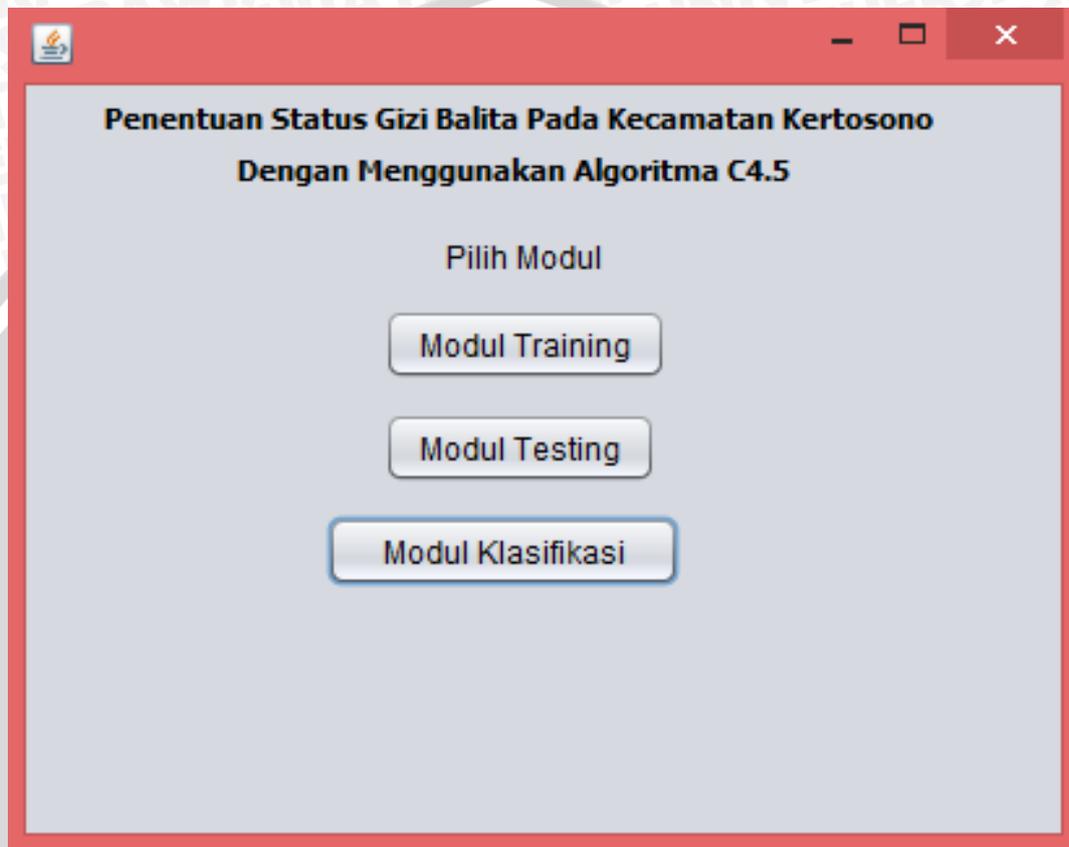
Pada penelitian ini data yang digunakan berupa data *testing* dan data *training*. Adapun rincian isi data yaitu umur, jenis kelamin, berat badan, status gizi balita di beberapa desa yang ada di Kecamatan Kertosono Kab. Nganjuk. Untuk data testing berupa data bayi berstatus gizi berjumlah 327 data dengan rincian yang sudah disebutkan pada Bab 3.

Data-data tersebut digunakan untuk data *testing* dan data *training* untuk aplikasi penelitian ini. Metode penyimpanan data pada penelitian ini dilakukan dengan mengumpulkan data balita yang disimpan dalam tabel-tabel.

4.3 Implementasi Antar Muka

4.3.1 Tampilan Awal

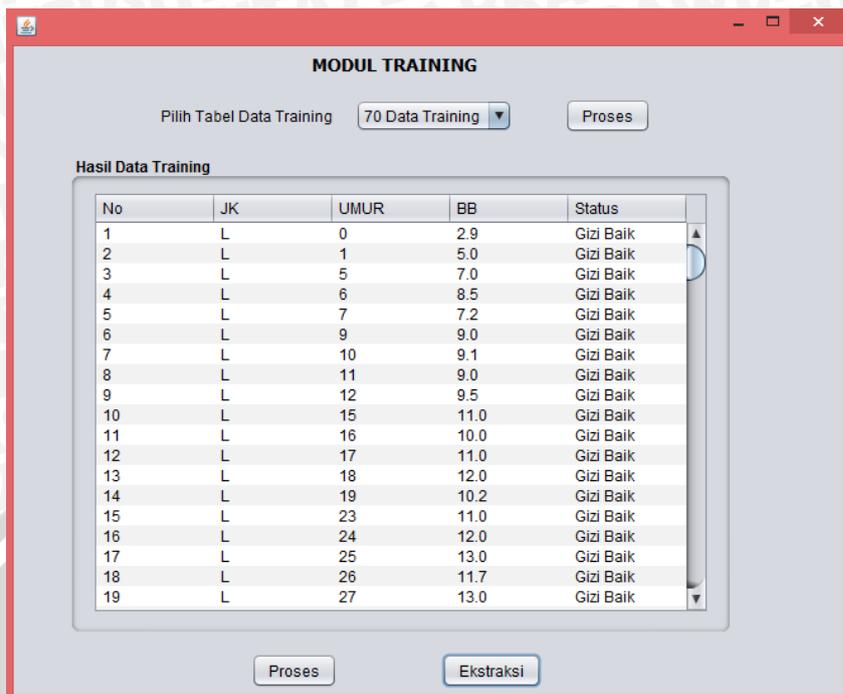
Pada Gambar 4.1 merupakan tampilan antarmuka awal, yang mana didalamnya terdapat 3 pilihan modul yaitu modul *training*, modul *testing*, dan modul klasifikasi.



Gambar 4.1 Tampilan Awal

4.3.2 Modul *Training*

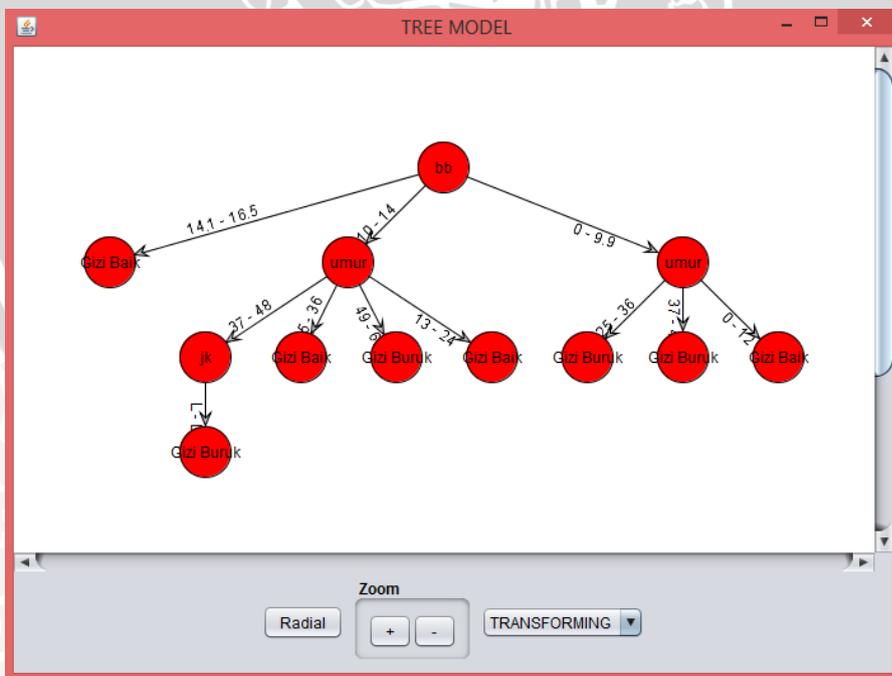
Pada Gambar 4.2 ini merupakan tampilan antarmuka dari modul *training*. Pada modul ini pengguna dapat memilih tabel mana yang akan dijadikan sebagai kumpulan data *training* kemudian sesudah memilih tabel akan tampil tabel yang dipilih.



Gambar 4. 2 Modul Training

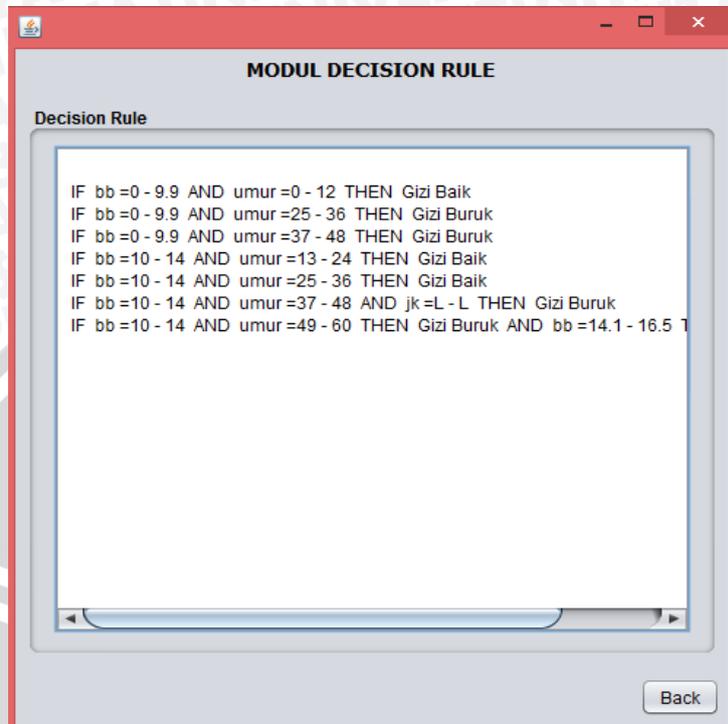
Pada bagian bawah terdapat *button* proses akan memunculkan *rule* apabila di klik dan *button* ekstrak akan menampilkan *tree* dari tabel data *training* yang dipilih.

Setelah memilih data *training* kemudian proses akan menampilkan tabel yang dipilih, pada *button* proses pada bagian bawah adalah menunjukkan



Gambar 4. 3 Tampilan Decision Tree

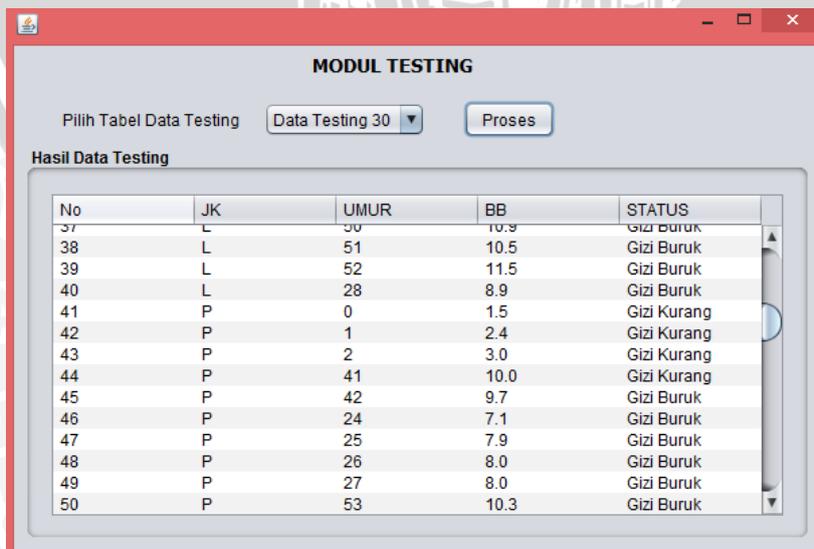
tampilan *tree* yang terbentuk dari tabel yang dipilih akan ditampilkan pada Gambar 4.3 dan Gambar 4.4 adalah tampilan ekstraksi dari *tree* menjadi *rule*.



Gambar 4. 4 Tampilan Decision Rule

4.3.3 Modul Testing

Pada Gambar 4.5 merupakan tampilan antarmuka dari modul *testing*. Pada modul ini terdapat pilihan tabel sesuai dengan pilihan menggunakan menyesuaikan dengan data *training* kemudian diproses setelah itu akan muncul status gizi balita pada tabel tersebut.



Gambar 4. 5 Modul Testing

4.3.4 Modul Klasifikasi

Pada Gambar 4.6 modul ini merupakan tampilan antarmuka dari modul klasifikasi menggunakan Algoritma C4.5. Pada modul ini penggunaan dapat mengetahui status gizi balita per individu dengan cara menginputkan nama, jenis kelamin, berat badan dan umur kemudian klik *button* proses dibawahnya akan menampilkan dari status gizi balita yang telah diinputkan tersebut dengan pengolahan data menggunakan Algoritma C4.5.

The screenshot shows a web application window titled "KLASIFIKASI STATUS GIZI BALITA" with the subtitle "MENGGUNAKAN ALGORITMA C4.5". The interface is divided into two main sections: "Input Data Balita" and "Hasil Klasifikasi".

Input Data Balita:

- Nama:** A text input field containing the name "dewi".
- Jenis Kelamin:** A dropdown menu currently set to "P".
- Umur:** A text input field containing the number "15".
- Berat Badan:** A text input field containing the number "9", followed by the unit "kg".

A "Proses" button is located at the bottom right of the input section.

Hasil Klasifikasi:

The result section displays the classification outcome: ***Gizi Buruk***.

Gambar 4. 6 Modul Klasifikasi

4.4 Implementasi Program

4.4.1 Program Proses Algoritma C4.5

Proses penerapan Algoritma C4.5 secara rancangan dapat dilihat pada Gambar 3.5 diagram alir Algoritma C4.5. Kemudian untuk penerapannya dibagi kedalam beberapa fungsi pada pemrogramannya. Diantaranya adalah menginisialisasi split per atributnya, menghitung jumlah kasus per atribut, menghitung nilai *entropy* tiap split per atributnya, menghitung *gain* per atribut, menentukan *gain* tertinggi atau *gain* max pada setiap atributnya, menentukan probabilitinya apabila terjadi 2 keluaran, kemudian pembentukan *tree* serta hasil klasifikasinya.

- **Menginisialisasi split setiap atribut**

Pada method ini menentukan atau menginisialisasi split pada tiap atribut, yang meliputi atribut jenis kelamin, umur dan berat badan. Pada inialisasi *case* 1 menyatakan JK (jenis kelamin), *case* 2 adalah umur dan *case* 3 yaitu BB (berat badan).

```
1 public String getNamaAtribut(int fitur,int index,int opsi) {
2     String atrib="";
3
4     if(feitur==1){
5         if(index==0){
6             atrib="L";
7         }
8         if(index==1){
9             atrib="P";
10        }
11    }
12    if(feitur==2){
13        if(index==0) {
14            if(opsi==1)atrib="0";
15            if(opsi==2)atrib="12";
16        }
17        if(index==1) {
18            if(opsi==1)atrib="13";
19            if(opsi==2)atrib="24";
20        }
21        if(index==2) {
22            if(opsi==1)atrib="25";
23            if(opsi==2)atrib="36";
24        }
25        if(index==3) {
26            if(opsi==1)atrib="37";
27            if(opsi==2)atrib="48";
28    }
```

```

29         if(index==4) {
30             if(opsi==1) atrib="49";
31             if(opsi==2) atrib="60";
32         }
33     }
34     if(fitur==3){
35         if(index==0) {
36             if(opsi==1) atrib="0";
37             if(opsi==2) atrib="9.9";
38         }
39         if(index==1) {
40             if(opsi==1) atrib="10";
41             if(opsi==2) atrib="14";
42         }
43         if(index==2) {
44             if(opsi==1) atrib="14.1";
45             if(opsi==2) atrib="16.5";
46         }
47         if(index==3) {
48             if(opsi==1) atrib="16.6";
49             if(opsi==2) atrib="200";
50         }
51     }
52     return atrib;
53 }

```

Kode Sumber 4.1 Inisialisasi Split dan Atribut

- **Menghitung Jumlah Kasus Pada Setiap Atribut Dan Split Yang Ada**

Pada Kode sumber ini menunjukkan perhitungan semua jumlah kasus pada setiap atribut beserta split yang sudah ditentukan sebelumnya. Peneliti mendeklarasikan perhitungan jumlah kasus dengan menginisialisasi peratribut. Misalnya pada atribut umur, perintah tersebut ditunjukkan pada kode sumber baris 5 – 23, disitu melakukan perulangan pada jumlah data yang memenuhi syarat. Kode sumber ini diterapkan kembali pada atribut yang lainnya.

```

1     public void iterasi(String namaFitur, Node node) {
2         System.out.println(" array tree " + pohon.toString());
3         if (namaFitur.equals("umur")) {
4
5             for (int z = 0; z < umur.length; z++) {
6                 dt.add(new
7                 data(node.getKelasData().getCountUmur(node.getKelasData().getMinUmur()[z],
8                 node.getKelasData().getMaxUmur()[z]));
9             }
10        }
11    }

```

```
        int rowcount =
node.getKelasData().getCountUmur(node.getKelasData().getMinUmur()[z],
8 node.getKelasData().getMaxUmur()[z]);
9
10     if (rowcount != 0) {
11         if (isPunyaAnak("umur", z, node)) {
12             for (int i = 0; i < rowcount; i++) {
13                 int index = node.getKelasData().
getIndexByUmur(dt.get(dt.size() - 1).getMinUmur()[z], dt.get(dt.size() -
13 1).getMaxUmur()[z])[i];
14                 sekuensialSetData(index, i, node);
15             }
16             System.out.println("sekuensial perhitungan UMUR");
17             sekuensialPerhitungan(2, node,z);
18         } else {
19             node.addChild(new
Node(gizi,node,getNamaAtribut(getIdFitur("umur"),z,1),
getNamaAtribut(getIdFitur("umur"),z,2));
20
21         }
22     }
23 }
24 } else if (namaFitur.equals("jk")) {
25
26     for (int z = 0; z < jk.length; z++) {
27         dt.add(new
data(node.getKelasData().getCountSex(node.getKelasData().getMinMaxJk()[z]));
28         int rowcount =
node.getKelasData().getCountSex(node.getKelasData().getMinMaxJk()[z]);
29         if (rowcount != 0) {
30             if (isPunyaAnak("jk", z, node)) {
31                 System.out.println(" JK PUNYA ANAK");
32                 for (int i = 0; i < rowcount; i++) {
33                     int index = node.getKelasData().
getIndexBySex(dt.get(dt.size() - 1).getMinMaxJk()[z])[i];
34                     sekuensialSetData(index, i, node);
35                 }
36                 System.out.println("sekuensial perhitungan jk");
37                 sekuensialPerhitungan(1, node,z);
38             } else {
39                 node.addChild(new Node(gizi,node,
getNamaAtribut(getIdFitur("jk"),z,1),getNamaAtribut(getIdFitur("jk"),z,2));
40             }
41         }
42     }
43 }
44 }
45 if (namaFitur.equals("bb")) {
46
47     for (int z = 0; z < bb.length; z++) {
```

```

48         dt.add(new data(node.getKelasData().getCountB
49         b(node.getKelasData().getMinBb()[z], node.getKelasData().getMaxBb()[z]));
50
51         int rowcount =
52         node.getKelasData().getCountBb(node.getKelasData().getMinBb()[z],
53         node.getKelasData().getMaxBb()[z]);
54         System.out.println("ROWCOUNT BB =" +
55         rowcount + " z =" + z + " bb length " + bb.length);
56
57         if (rowcount != 0) {
58             if (isPunyaAnak("bb", z, node)) {
59                 for (int i = 0; i < rowcount; i++) {
60                     int index = node.getKelasData().
61                     getIndexByBb(dt.get(dt.size() - 1).getMinBb()[z], dt.get(dt.size() - 1).
62                     getMaxBb()[z])[i];
63
64                     sekuensialSetData(index, i, node);
65                 }
66                 sekuensialPerhitungan(3, node, z);
67             } else {
68                 node.addChild(new
69                 Node(gizi, node, getNamaAtribut(getIdFitur("bb"), z, 1),
70                 getNamaAtribut(getIdFitur("bb"), z, 2)));
71             }
72         }
73     }
74     else if (namaFitur.equals("end")) {
75     }
76 }

```

Kode Sumber 4.2 Menentukan Jumlah Kasus pada Split dan Atribut

- Menghitung nilai *entropy* pada setiap split dan atribut

Rumus *entropy* yang diterapkan untuk menghitung setiap split yang ada pada setiap atribut yang sudah ditentukan, ditunjukkan pada kode sumber 4.3 baris 7 dan diulang untuk keseluruhan data.

```

1 public void hitungEntropy(Node node) {
2     int dataHitung[][] = isiDataHitung(hitungNBaris(node), node);
3     showData(dataHitung, " SHOW DATA 2D");
4     entropy = new double[dataHitung.length];
5     for (int i = 0; i < entropy.length; i++) {
6         for (int j = 0; j < 4; j++) {
7             if (dataHitung[i][j + 1] != 0) {
8                 entropy[i] += (-(bagi(dataHitung[i][j + 1],
9                 dataHitung[i][0])) * log(dataHitung[i][j] + 1, dataHitung[i][0]));

```

Kode Sumber 4.3 Perhitungan *Entropy*

- **Menghitung nilai *gain* pada setiap atribut**

Menghitung rumus *gain* pada setiap atribut saja, pada *sourcecode* dibawah ini ditentukan pada atribut JK diinisialisasikan pada hitung 1, hitung 2 adalah atribut umur dan hitung 3 yaitu atribut BB.

```
1 public void hitungGain(Node node) {
2     int dataHitung[][] = isiDataHitung(hitungNBaris(node), node);
3     gain = new double[3];
4     int counter = 1;
5     if (node.getFiturHitung(1)) {
6         counter += 2;
7         for (int i = 1; i < counter; i++) {
8             gain[0] += (bagi(dataHitung[i][0], dataHitung[0][0]) *
9 entropy[i]);
10        }
11        gain[0] = entropy[0] - gain[0];
12    }
13    if (node.getFiturHitung(2)) {
14        counter += 5;
15        for (int i = counter - 5; i < counter; i++) {
16            gain[1] += (bagi(dataHitung[i][0], dataHitung[0][0]) *
17 entropy[i]);
18        }
19        gain[1] = entropy[0] - gain[1];
20    }
21    if (node.getFiturHitung(3)) {
22        counter += 4;
23        for (int i = counter - 4; i < counter; i++) {
24            gain[2] += (bagi(dataHitung[i][0], dataHitung[0][0]) *
25 entropy[i]);
26        }
27        gain[2] = entropy[0] - gain[2];
28    }
29 }
30
31
32
33
34 }
```

Kode Sumber 4.4 Menghitung Nilai *Gain* Pada Setiap Atribut

- **Menentukan *Gain* Tertinggi Dari Semua Atribut**

Dari semua nilai *gain* yang sudah dihitung pada proses ini adalah *sorting* nilai *gain* yang tertinggi dari nilai *gain* yang sudah dihitung sebelumnya. *Sorting* data ditunjukkan pada kode sumber baris 4 – 8 dan diambil nilai yang tertinggi.

```

1   public String getMaxGain() {
2       int index = 1;
3       double inisial = gain[0];
4       for (int i = 0; i < gain.length - 1; i++) {
5           if (inisial < gain[i + 1]) {
6               inisial = gain[i + 1];
7               index = i + 2;
8           } else {
9               // index=i+1;
10          }}

```

Kode Sumber 4.5 Menentukan *Gain* Tertinggi Atribut

- **Menghitung *Probability* Bila Terjadi 2 Output**

Pada perhitungan ini dilakukan ketika pada perhitungan mendapatkan 2 keluaran/*output* maka dipakailah rumus *probability* ini yang bertujuan untuk menghasilkan satu keluaran saja. Ketika hanya memiliki satu status gizi balita maka rumus *probability* ini tidak dipakai. Rumus inti *probability* ditunjukkan pada kode sumber 46 – 48. Rumus *probability* ini diterapkan untuk setiap atribut.

```

1   public boolean isPunyaAnak(String fitur, int z, Node node) {
2       boolean punyaAnak = false;
3       if (fitur.equals("umur")) {
4           int nZero = 0;
5           int index = 0;
6           for (int j = 1; j < umur[0].length; j++) {
7               if (umur[z][j] == 0) {
8                   nZero++;
9               } else {
10                  index = j;
11              }
12          }
13          if (umur[0].length - 1 - nZero == 1) {
14              punyaAnak = false;
15              gizi = getNamaGiziById(index);
16          } else {
17              punyaAnak = true;
18              System.out.println("node parent = " +
19              node.getParent());

```

```
20     Node ank;
21     ank = node.getParent();
22     if (ank != null) {
23         if(ank.getParent()!=null){
24
25             dt.get(0).showData();
26             double probabilitas = 0;
27             int indexProba = 0;
28             for (int i = 1; i < umur[z].length; i++) {
29                 String namaGizi = "";
30                 if (i == 1) {
31                     namaGizi = "Gizi Buruk";
32                 }
33                 if (i == 2) {
34                     namaGizi = "Gizi Kurang";
35                 }
36                 if (i == 3) {
37                     namaGizi = "Gizi Baik";
38                 }
39                 if (i == 4) {
40                     namaGizi = "Gizi Lebih";
41                 }
42                 if (umur[z][i] > 0) {
43                     double p = 0;
44                     double a =
dt.get(0).getCountStatus(namaGizi);
45                     double b =
dt.get(0).getCountAllData();
46                     p = a / b;
47
48                     p = umur[z][i] / p;
49
50                     if (p >= probabilitas) {
51                         probabilitas = p;
52                         indexProba = i;
53                     }
54                 }
55             }
56             gizi = getNamaGiziById(indexProba);
57
58             punyaAnak = false;
59         }
60     }
61 }
62 }
63 } else if (fitur.equals("jk")) {
```

```
74     int nZero = 0;
75     int index = 0;
76     for (int j = 1; j < jk[0].length; j++) {
77         if (jk[z][j] == 0) {
78             nZero++;
79         } else {
80             index = j;
81         }
82     }
83
84     if (jk[0].length - 1 - nZero == 1) {
85         punyaAnak = false;
86         gizi = getNamaGiziById(index);
87         System.out.println("gizi " + gizi);
88     } else {
89         punyaAnak = true;
90
91         Node ank;
92         ank = node.getParent();
93         if (ank != null) {
94             if (ank.getParent() != null) {
95
96                 dt.get(0).showData();
97                 double probabilitas = 0;
98                 int indexProba = 0;
100                for (int i = 1; i < jk[z].length; i++) {
101                    String namaGizi = "";
102                    if (i == 1) {
103                        namaGizi = "Gizi Buruk";
104                    }
105                    if (i == 2) {
106                        namaGizi = "Gizi Kurang";
107                    }
108                    if (i == 3) {
109                        namaGizi = "Gizi Baik";
110                    }
111                    if (i == 4) {
112                        namaGizi = "Gizi Lebih";
113                    }
114                    if (jk[z][i] > 0) {
115                        double p = 0;
116                        double a =
117                        dt.get(0).getCountStatus(namaGizi);
118                        double b =
119                        dt.get(0).getCountAllData();
120                        p = a / b;
```

```
120         p = jk[z][i] / p;  
121  
122         if (p >= probabilitas) {  
123             probabilitas = p;  
124             indexProba = i;  
125         }  
126     }  
127 }  
128 gizi = getNamaGiziById(indexProba);  
129  
130     punyaAnak = false;  
131 }  
132 }  
133 }  
134 }  
135     } else if (fitur.equals("bb")) {  
136         int nZero = 0;  
137         int index = 0;  
138         for (int j = 1; j < bb[0].length; j++) {  
139             if (bb[z][j] == 0) {  
140                 nZero++;  
141             } else {  
142                 index = j;  
143             }  
144         }  
145         if (bb[0].length - 1 - nZero == 1) {  
146             punyaAnak = false;  
147             gizi = getNamaGiziById(index);  
148             System.out.println("gizi " + gizi);  
149         } else {  
150             punyaAnak = true;  
151  
152             Node ank;  
153             ank = node.getParent();  
154             if (ank != null) {  
155                 if (ank.getParent() != null) {  
156  
157                     if (getNUnikStatus(node) <= bb.length) {  
158                         dt.get(0).showData();  
159                         double probabilitas = 0;  
160                         int indexProba = 0;  
161                         for (int i = 1; i < bb[z].length; i++) {  
162                             String namaGizi = "";  
163                             if (i == 1) {
```

```
164         namaGizi = "Gizi Buruk";
165     }
166     if (i == 2) {
167         namaGizi = "Gizi Kurang";
168     }
169     if (i == 3) {
170         namaGizi = "Gizi Baik";
171     }
172     if (i == 4) {
173         namaGizi = "Gizi Lebih";
174     }
175     if (bb[z][i] > 0) {
176         double p = 0;
177         double a =
178 dt.get(0).getCountStatus(namaGizi);
179 dt.get(0).getCountAllData();
180         p = a / b;
181         System.out.println(" ini p " + p);
182         p = bb[z][i] / p;
183         if (p >= probabilitas) {
184             probabilitas = p;
185             indexProba = i;
186         }
187     }
188     }
189     }
190     gizi = getNamaGiziById(indexProba);
191     punyaAnak = false;
192     }
193     }
194     }
195     }
196     }
197     }
198     }
199     return punyaAnak;
200 }
201 }
```

Kode Sumber 4.6 Perhitungan Probabilitas

- **Pembentukan *Tree***

Kode sumber 4.7 menunjukkan penerapan pembentukan *tree*.

```
1 package c45;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Node<T> {
7
8     public T data;
9     public List<Node<T>> children;
10    public data kelasData;
11    boolean fiturUmur = true;
12    boolean fiturTb = true;
13    boolean fiturBb = true;
14    boolean fiturJk = true;
15    public Node<T> parent=null;
16    String atribut1="";
17    String atribut2="";
18    String rule="";
19
20    public Node(T data,Node ortu,String atrib1,String atrib2) {
21        this();
22        setData(data);
23        parent=ortu;
24        atribut1=atrib1;
25        atribut2=atrib2;
26    }
27
28    public List<Node<T>> getChildren() {
29        if (this.children == null) {
30            return new ArrayList<Node<T>>();
31        }
32        return this.children;
33    }
34
35    public int getNumberOfChildren() {
36        if (children == null) {
37            return 0;
38        }
39        return children.size();
40    }
}
```

```
41
42     public void addChild(Node<T> child) {
43         if (children == null) {
44             children = new ArrayList<Node<T>>();
45         }
46         children.add(child);
47
48     }
49
50     public void insertChildAt(int index, Node<T> child) throws
IndexOutOfBoundsException {
51         if (index == getNumberOfChildren()) {
52
53             addChild(child);
54             return;
55         } else {
56             children.get(index);
57             children.add(index, child);
58         }
59     }
60
61     public void removeChildAt(int index) throws
IndexOutOfBoundsException {
62         children.remove(index);
63     }
64
65     public T getData() {
66         return this.data;
67     }
68
69     public void setData(T data) {
70         this.data = data;
71     }
72     public data getKelasData(){
73         return kelasData;
74     }
75
76     public String toString2() {
77         StringBuilder sb = new StringBuilder();
78         sb.append("{").append(getData().toString()).append(", [");
79         int i = 0;
80         for (Node<T> e : getChildren()) {
81             if (i > 0) {
82                 sb.append(", ");
83             }
84             sb.append(e.getData().toString());
```

```
85         i++;
86     }
87     sb.append("]").append(")");
88     return sb.toString();
89 }
90 public String toString(){
91     return this.getData().toString();
92 }
93 public void setFiturTakDihitung(int index) {
94     Node ortu=parent;
95
96
97     switch (index) {
98         case 1:
99             fiturJk = false;
100            break;
101         case 2:
102             fiturUmur = false;
103            break;
104         case 3:
105             fiturBb = false;
106            break;
107
108
109
110         default:
111             fiturJk = true;
112             fiturUmur = true;
113             fiturBb = true;
114             fiturTb = true;
115             break;
116     }
117 }
118 public boolean getFiturHitung(int index) {
119     boolean bolFiturHitung = false;
120     switch (index) {
121         case 1:
122             bolFiturHitung = fiturJk;
123             break;
124         case 2:
125             bolFiturHitung = fiturUmur;
126             break;
127         case 3:
128             bolFiturHitung = fiturBb;
```

```

129         break;
130     case 4:
131         bolFiturHitung = fiturTb;
132         break;
133     }
134     return bolFiturHitung;
135 }
136 public void samakanFitur(Node node) {
137     if (fiturBb)
138         this.fiturBb = node.fiturBb;
139     if (fiturJk)
140         this.fiturJk = node.fiturJk;
141     if (fiturUmur)
142         this.fiturUmur = node.fiturUmur;
143     if (fiturTb)
144         this.fiturTb = node.fiturTb;
145 }
146
147 public Node<T> getParent() {
148     return parent;
149 }
150 public String getAtrib1() {
151     return atribut1;
152 }
153 public String getAtrib2() {
154     return atribut2;
155 }
156 public void setRule(String aturan) {
157     rule = aturan;
158 }
159 public String getRule() {
160     return rule;
161 }
162 }

```

Kode Sumber 4.7 Pembentukan Tree

- **Menampilkan Tree**

Kode sumber 4.8 ini menerapkan *tree* yang sudah terbentuk untuk ditampilkan ke layar.

```

1 private void createTree() throws IOException {
2
3     graph.addVertex(pohon.getRootElement());
4     iterasi(pohon.getRootElement(), pohon.getRootElement().getNumberOfChildren());
5 }

```

```

6     }
7     int counter=0;
8     public void iterasi(Node node,int numberOfChildren){
9         for (int i = 0; i < numberOfChildren; i++) {
10            Node nodeAnak=(Node)node.getChildren().get(i);
11            graph.addVertex(nodeAnak);
12
13            graph.addEdge(new edge(nodeAnak.getAtrib1(),
14 nodeAnak.getAtrib2()),node,nodeAnak);
15
16            iterasi(nodeAnak, nodeAnak.getNumberOfChildren());
17
18        }

```

Kode Sumber 4.8 Menampilkan Tree

- **Menentukan Hasil Klasifikasi Menggunakan Algoritma C4.5**

Kode sumber 4.9 ini menerapkan hasil klasifikasi dari Algoritma C4.5 dari percabangan yang ditunjukkan pada kode sumber mulai dari baris 6 dicari node paling bawah untuk ditampilkan di status tabel yang berisi nilai status gizi.

```

        public void iterasiImplementationRule(Node node, int numberOfChildren,
1 String jk, int umur, double bb) {
2     Node nodeCek = new Node();
3     for (int i = 0; i < numberOfChildren; i++) {
4         Node anak = (Node) node.getChildren().get(i);
5         boolean benar = false;
6         if (cekNodeBb2(bb, anak)) {
7             benar = true;
8
9         } else if (cekNodeJk2(jk, anak)) {
10            benar = true;
11        } else if (cekNodeUmur2(umur, anak)) {
12            benar = true;
13        }
14        if (benar) {
15
16            i = numberOfChildren + 1;
17            nodeCek = anak;
18        }
19
20    }
21    if (nodeCek.getChildren().size() > 0) {
22        for (int i = 0; i < nodeCek.getChildren().size(); i++) {
23            Node anak = (Node) nodeCek.getChildren().get(i);
24            boolean benar = false;

```

```
25     if (cekNodeBb2(bb, anak)) {
26         benar = true;
27
28     } else if (cekNodeJk2(jk, anak)) {
29         benar = true;
30     } else if (cekNodeUmur2(umur, anak)) {
31         benar = true;
32     }
33     if (benar) {
34         System.out.println("INI JALAN");
35         i = nodeCek.getChildren().size() + 1;
36         nodeCek = anak;
37     }
38
39 }
40 }
41 if (nodeCek.getChildren().size() > 0) {
42     for (int i = 0; i < nodeCek.getChildren().size(); i++) {
43         Node anak = (Node) nodeCek.getChildren().get(i);
44         boolean benar = false;
45         if (cekNodeBb2(bb, anak)) {
46             benar = true;
47
48         } else if (cekNodeJk2(jk, anak)) {
49             benar = true;
50         } else if (cekNodeUmur2(umur, anak)) {
51             benar = true;
52         }
53         if (benar) {
54             i = nodeCek.getChildren().size() + 1;
55             nodeCek = anak;
56         }
57
58     }
59 }
60 }
```

Kode Sumber 4.9 Klasifikasi Algoritma C4.5

BAB 5 PENGUJIAN DAN PEMBAHASAN

Pada Bab 5 ini peneliti akan memaparkan mengenai tahapan pengujian serta analisis dari sistem penentuan status gizi pada balita yang telah diimplementasikan pada bab sebelumnya.

5.1 Sistematika Pengujian

Pada tahap pengujian ini terdapat 2 macam pengujian yang dilakukan oleh peneliti untuk mengetahui tingkat akurasi yang dihasilkan oleh sistem. Pengujian yang pertama dilakukan dengan menguji data *training* dan data *testing* yang sama. Pengujian yang kedua adalah pengujian dengan menggunakan jumlah data *training* yang berbeda dan jumlah data *testing* yang sama menggunakan data yang diperoleh peneliti. Dari kedua pengujian yang dilakukan, maka nantinya dapat diketahui valid atau tidaknya data serta keakurasian sistem yang dibuat menggunakan algoritma C4.5 tersebut.

5.1.1 Sistematika Pengujian Data *Training* dan Data *Testing* Sama

Pada pengujian ini dilakukan dengan menggunakan data *training* dan data *testing* sama. Pengujian ini dilakukan untuk mengetahui tingkat kesamaan antara data *training* dan data *testing* menghasilkan akurasi yang tertinggi atau yang rendah. Sistematika pengujiannya adalah sebagai berikut:

- Data 1 : 322 data *training* dan data *testing*
- Data 2 : 23 data *training* dan data *testing*
- Data 3 : 65 data *training* dan data *testing*
- Data 4 : 98 data *training* dan data *testing*

5.1.2 Sistematika Pengujian Pengaruh Perubahan Data *Training* dengan Data *Testing* yang Sama

Pada pengujian yang kedua ini dilakukan dengan cara menguji pengaruh perubahan data *training* dan data *testing* untuk mengetahui tingkat akurasi sistem yang digunakan untuk mengetahui status gizi balita ini menggunakan Algoritma C4.5. Pada pengujian perubahan data akan dilakukan sebanyak 3 kali perubahan data *training*, yaitu sebagai berikut ini:

- 50 data *training* dan 50 data *testing*
- 82 data *training* dan 50 data *testing*
- 145 data *training* dan 50 data *testing*
- 287 data *training* dan 50 data *testing*

5.2 Hasil Pengujian

Hasil pengujian ini dilakukan untuk mengetahui seberapa tepatnya algoritma yang diterapkan dalam sistem dengan data yang ada. Yang nantinya juga akan menghasilkan nilai akurasi data dengan sistem aplikasi.

5.2.1 Hasil Pengujian dengan Data *Training* dan Data *Testing* yang Sama

Berdasarkan pengujian data *training* dan data *testing* menggunakan data yang sama, maka dapat ditunjukkan hasil akurasi yang diperoleh pada Tabel 5.1 berikut ini :

Tabel 5.1 Hasil Uji Data Training dan Data Testing yang Sama

No	Jenis Pengujian	Nilai AKurasi (%)
1	322 data <i>training</i> dan data <i>testing</i>	100%
2	23 data <i>training</i> dan data <i>testing</i>	100%
3	65 data <i>training</i> dan data <i>testing</i>	100%
4	98 data <i>training</i> dan data <i>testing</i>	100%
Rata-Rata		100%

5.2.2 Hasil Uji Pengaruh Data *Training* dan Data *Testing*

Pada pengujian pengaruh data *training* dan data *testing* digunakan 50 data *training* dan 50 data *testing*, 82 data *training* dan 50 data *testing* serta 145 data *training* dan 50 data *testing*. Sehingga setelah diuji akan mendapatkan nilai akurasi pada Tabel 5.2 berikut ini :

Tabel 5.2 Hasil Akurasi Perubahan Jumlah Data

No	Jenis Pengujian	Nilai Akurasi (%)
1	50 data <i>training</i> dan 50 data <i>testing</i>	96%
2	82 data <i>training</i> dan 50 data <i>testing</i>	98%
3	145 data <i>training</i> dan 50 data <i>testing</i>	98%
4	287 data <i>training</i> dan 50 data <i>testing</i>	98%
Rata-rata		97,5%

5.3 Analisa Hasil Pengujian

Berdasarkan hasil pengujian yang dilakukan sebanyak 2 macam pengujian telah didapatkan nilai akurasinya pada Tabel 5.1 dan Tabel 5.2 di atas. Pada pengujian pertama menggunakan jumlah data *training* dan data *testing* yang sama menghasilkan nilai akurasi rata-rata 100% dari jumlah data yang digunakan sebanyak 4 macam data, hal ini dimaksudkan untuk melihat valid atau tidaknya data terhadap kesesuaian metode yang digunakan atau bias juga dikatakan untuk melihat kesesuaian implementasi dengan teori.

Sedangkan pada pengujian kedua menggunakan jumlah data *training* dan jumlah data *testing* yang berbeda untuk mengetahui nilai akurasi apabila data

yang digunakan mengalami perubahan atau tidak. Adapun hasil dari pengujian kedua dengan 50 data *training* dan 50 data *testing* mendapatkan nilai akurasi sebesar 96%, data *training* 82 dan data *testing* 50 data *testing* menghasilkan nilai akurasi 98% dan data yang terakhir yaitu 145 data *training* dan 50 data *testing* menghasilkan nilai akurasi sebanyak 98%. Pada pengujian kedua menggunakan jumlah data *training* berbeda dan jumlah data *testing* yang sama mendapatkan nilai rata-rata akurasi sebesar 97,5%. Pada pengujian kedua memiliki nilai akurasi yang berbeda hal ini disebabkan karena perbandingan jumlah data *training* yang digunakan, sehingga nilai akurasi akan berpengaruh. Apabila semakin besar jumlah data *training* yang digunakan maka nilai akurasi juga semakin tinggi.

Sehingga dari kedua pengujian yang telah dilakukan dapat disimpulkan hasil dari nilai akurasinya berbeda. Nilai akurasi pada pengujian yang pertama, sebesar 100% yang bertujuan untuk membuktikan validasi data dan kesesuaian implementasi dan teori yang digunakan. Hal tersebut dikarenakan jumlah data yang digunakan sebagai *training* dan *testing* adalah data yang sama. Adapun pengujian yang kedua menggunakan jumlah data *training* yang berbeda dan jumlah data *testing* yang sama nilai akurasinya juga masih tinggi, dapat dilihat apabila data *training* bertambah maka nilai akurasi juga semakin tinggi. Dan dapat disimpulkan bahwa dari nilai akurasi sebesar 97,5% Algoritma C4.5 ini sangat tepat, cocok dan efektif dalam menentukan klasifikasi status gizi balita. Adapun kesesuaiannya bisa saat menentukan *split* dan atribut yang digunakan, apabila *split* dan atribut yang digunakan tepat maka akan pula menghasilkan *node-node tree* dan *rule* yang bagus.



BAB 6 PENUTUP

Pada Bab 5 ini memuat mengenai kesimpulan dan saran peneliti dari terhadap skripsi ini. Adapun penulisan kesimpulan dan saran ditulis secara terpisah, dengan penjelasan sebagai berikut ini :

6.1 Kesimpulan

Kesimpulan yang dapat diperoleh dari penelitian ini yaitu mengacu pada rumusan masalah, tujuan serta hasil analisis yang telah dilakukan, maka dapat disimpulkan bahwa:

1. Penerapan Algoritma C4.5 untuk pengklasifikasian status gizi balita menggunakan 4 kelas yaitu gizi buruk, gizi kurang, gizi baik, gizi lebih. Adapun atribut yang digunakan untuk menentukan status gizi balita adalah jenis kelamin, umur dan berat badan balita, diolah menggunakan Algoritma C4.5 untuk mendapatkan hasil status gizi balita dengan proses sebagai berikut :
 - Menentukan split dan atribut yang digunakan
 - Menghitung jumlah kasus dari semua data pada setiap *split* dan atribut
 - Menghitung nilai *entropy*
 - Menghitung nilai *gain*
 - Menghitung nilai probabilitasnya apabila terjadi 2 *output* status gizi balita
 - Menentukan *Tree* dari nilai *gain* tertinggi (root)
 - Menentukan *Decision Rule* dari *tree* yang terbentuk
2. Hasil akurasi yang diperoleh dari pengujian yang telah dilakukan pada data training dan testing yang sama menunjukkan nilai rata-rata akurasi sebesar 100% dan pengujian kedua menghasilkan nilai rata-rata akurasi sebesar 97,5% hal tersebut menunjukkan Algoritma C4.5 ini sangatlah cocok digunakan untuk menghitung klasifikasi status gizi balita.

6.2 Saran

1. Untuk penelitian selanjutnya harap memperhatikan penentuan split yang digunakan karena nantinya juga akan berpengaruh pada hasil akhir yang diperoleh baik itu *tree* ataupun *rulanya*.
2. Untuk metode klasifikasi Algoritma C4.5 dapat diperbandingkan performanya dengan metode klasifikasi yang lain.

DAFTAR PUSTAKA

- Almatsier, S. 2005. *Prinsip Dasar Gizi*. Jakarta: Gramedia Pustaka Utama
- Anggraeni, Reni dan Indriati, Aviarini. 2010. *Klasifikasi status Gizi Balita Berdasarkan indeks Antropometri (BB/U) menggunakan jaringan syaraf tiruan*. Fakultas Ilmu Komputer dan Teknologi Informasi universitas Gunadarma. <http://repository.gunadarma.ac.id/853/1/KLASIFIKASI%20STATAUS%20GIZI%20BALITA%20BERDASARKAN%20INDEKS%20ANTROPOMETRI%20MENGUNAKAN%20JARINGAN%20SARAF%20TIRUAN> UG.pdf. [Diakses 19 Maret 2015]
- Anik Andriani. 2013. *Penerapan Algoritma C4.5 pada Program Klasifikasi Mahasiswa Dropout*. Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK)
- Berry, Michael J.A dan Linoff, Gordon S. 2004. *Data Mining Techniques For Marketing, Sales, Customer Relationship Management Second Editon*. United States of America: Wiley Publishing, Inc
- Djoko Pekik Irianto. 2007. *Panduan Gizi Lengkap Keluarga dan Olahragawan*. Yogyakarta: Andi Offset.
- Kusrini. 2009. *Algoritma Data Mining*. Yogyakarta: Andi Offset
- Larose, D. T. 2005. *Discovering Knowledge in Data*. New Jersey : John Willey & Sons, Inc.
- Menkes, RI. 2011. *Standar Antropometri Penilaian Status Gizi Anak*. Jakarta
- Muaris H. 2006. *Sarapan Sehat Untuk Anak Balita*. Jakarta: PT Gramedia Pustaka
- Pramudiono, I. 2006. *Apa itu Data Mining?*. <http://datamining.japati.net/cgi-bin/indodm.cgi>. [Diakses tanggal 26 Februari 2015]
- Supariasa, dkk. 2002. *Penilaian Status Gizi*. Jakarta: Penerbit Kedokteran RGC
- Sutomo, B dan Anggraini, D.Y. 2010. *Makanan Sehat Pendamping ASI*. Jakarta: PT. Agromedia Pustaka.
- Turban, Efraim, et al. 2005. *Decision Support System and Intelligent System 7th Ed*. New Jersey: Pearson Education
- Quinlan, J.R. 1993. *C4.5 program for machine learning*. San Fransisco: Morgan Kaufmann Publishers Inc

LAMPIRAN A DATA BALITA

Indeks	Jenis Kelamin	Umur	Berat Badan	Status Gizi Balita
1	L	13	7.4	Gizi Kurang
2	L	15	7.5	Gizi Kurang
3	L	16	7.8	Gizi Kurang
4	L	17	7.9	Gizi Kurang
5	L	18	8.2	Gizi Kurang
6	L	19	8.3	Gizi Kurang
7	L	23	8.8	Gizi Kurang
8	L	37	10.5	Gizi Kurang
9	L	38	10.4	Gizi Kurang
10	L	39	10.7	Gizi Kurang
11	L	40	10.9	Gizi Kurang
12	L	41	10.7	Gizi Kurang
13	L	42	10.8	Gizi Kurang
14	L	43	11	Gizi Kurang
15	L	44	11.3	Gizi Kurang
16	L	45	11	Gizi Kurang
17	L	48	11.7	Gizi Kurang
18	P	1	2.8	Gizi Kurang
19	P	2	3.6	Gizi Kurang
20	P	3	4.1	Gizi Kurang
21	P	4	4.6	Gizi Kurang
22	P	5	5	Gizi Kurang
23	P	9	6	Gizi Kurang
24	P	10	6	Gizi Kurang
25	P	11	6.3	Gizi Kurang
26	P	12	6.5	Gizi Kurang
27	P	13	6.5	Gizi Kurang
28	P	14	6.8	Gizi Kurang
29	P	15	6.9	Gizi Kurang
30	P	16	7.2	Gizi Kurang
31	P	19	7.5	Gizi Kurang
32	P	21	7.8	Gizi Kurang
33	P	22	8	Gizi Kurang
34	P	23	8.2	Gizi Kurang
35	P	24	8.2	Gizi Kurang
36	P	37	10	Gizi Kurang
37	P	38	10	Gizi Kurang
38	P	39	10.3	Gizi Kurang
39	P	40	10.4	Gizi Kurang
40	P	41	10.5	Gizi Kurang

41	P	42	10.8	Gizi Kurang
42	P	46	10.9	Gizi Kurang
43	P	47	11.2	Gizi Kurang
44	P	48	11.4	Gizi Kurang
45	P	49	11.3	Gizi Kurang
46	P	50	11.5	Gizi Kurang
47	P	52	11.8	Gizi Kurang
48	P	53	11.9	Gizi Kurang
49	P	56	12.2	Gizi Kurang
50	P	59	12.6	Gizi Kurang
51	L	20	8.4	Gizi Kurang
52	L	21	8.5	Gizi Kurang
53	L	22	8.6	Gizi Kurang
54	P	6	5.3	Gizi Kurang
55	P	7	5.5	Gizi Kurang
56	P	8	5.8	Gizi Kurang
57	P	54	12	Gizi Kurang
58	P	55	12.1	Gizi Kurang
59	P	0	2.1	Gizi Kurang
60	P	17	7.2	Gizi Kurang
61	P	18	7.4	Gizi Kurang
62	P	43	11	Gizi Kurang
63	P	44	10.9	Gizi Kurang
64	P	45	11.2	Gizi Kurang
65	P	60	12.4	Gizi Kurang
66	L	46	11.6	Gizi Kurang
67	L	47	11.5	Gizi Kurang
68	P	57	12.4	Gizi Kurang
69	P	58	12.3	Gizi Kurang
70	P	20	7.7	Gizi Kurang
71	L	24	9	Gizi Kurang
72	P	1	2.8	Gizi Kurang
73	P	2	3.6	Gizi Kurang
74	P	13	6.5	Gizi Kurang
75	L	39	10.7	Gizi Kurang
76	L	14	7.5	Gizi Kurang
77	P	51	11.6	Gizi Kurang
78	L	9	13	Gizi Lebih
79	L	10	12.8	Gizi Lebih
80	L	11	13.8	Gizi Lebih
81	L	12	13.9	Gizi Lebih
82	L	14	15	Gizi Lebih
83	L	15	14.8	Gizi Lebih

84	L	16	15.2	Gizi Lebih
85	L	17	15.9	Gizi Lebih
86	L	21	16.8	Gizi Lebih
87	L	22	17.5	Gizi Lebih
88	L	23	17.8	Gizi Lebih
89	L	24	18.2	Gizi Lebih
90	L	25	18	Gizi Lebih
91	L	26	18.5	Gizi Lebih
92	L	28	19.2	Gizi Lebih
93	L	31	20	Gizi Lebih
94	L	33	20.4	Gizi Lebih
95	L	34	20.6	Gizi Lebih
96	L	35	21	Gizi Lebih
97	L	36	21.3	Gizi Lebih
98	L	37	24	Gizi Lebih
99	L	38	23	Gizi Lebih
100	L	39	22	Gizi Lebih
101	L	40	23.3	Gizi Lebih
102	L	41	24	Gizi Lebih
103	L	42	24.5	Gizi Lebih
104	L	45	25	Gizi Lebih
105	L	47	25.7	Gizi Lebih
106	L	48	24	Gizi Lebih
107	L	49	25	Gizi Lebih
108	L	50	24.6	Gizi Lebih
109	L	53	27	Gizi Lebih
110	L	54	26.5	Gizi Lebih
111	L	55	27.3	Gizi Lebih
112	L	56	28	Gizi Lebih
113	L	57	27	Gizi Lebih
114	L	58	28.4	Gizi Lebih
115	L	59	27.9	Gizi Lebih
116	P	6	11.3	Gizi Lebih
117	P	7	11.7	Gizi Lebih
118	P	8	12.3	Gizi Lebih
119	P	9	13.4	Gizi Lebih
120	P	10	13	Gizi Lebih
121	P	11	13.5	Gizi Lebih
122	P	12	13.4	Gizi Lebih
123	P	14	14.7	Gizi Lebih
124	P	15	15.2	Gizi Lebih
125	P	18	16	Gizi Lebih
126	P	19	16.2	Gizi Lebih

127	P	20	16.7	Gizi Lebih
128	P	21	17	Gizi Lebih
129	P	22	17.8	Gizi Lebih
130	P	23	16.9	Gizi Lebih
131	P	24	17.4	Gizi Lebih
132	P	25	17.8	Gizi Lebih
133	P	29	19.3	Gizi Lebih
134	P	30	20.3	Gizi Lebih
135	P	31	19.7	Gizi Lebih
136	P	32	20.4	Gizi Lebih
137	P	33	21	Gizi Lebih
138	P	34	20.7	Gizi Lebih
139	P	35	21.2	Gizi Lebih
140	P	36	21	Gizi Lebih
141	P	37	22.2	Gizi Lebih
142	P	39	22.5	Gizi Lebih
143	P	40	22	Gizi Lebih
144	P	42	24	Gizi Lebih
145	P	43	24	Gizi Lebih
146	P	47	24.9	Gizi Lebih
147	P	48	24	Gizi Lebih
148	P	49	25.6	Gizi Lebih
149	P	50	26	Gizi Lebih
150	P	51	27.3	Gizi Lebih
151	P	52	27	Gizi Lebih
152	P	53	28.3	Gizi Lebih
153	P	54	28	Gizi Lebih
154	P	55	28.6	Gizi Lebih
155	P	56	28.5	Gizi Lebih
156	P	57	29.4	Gizi Lebih
157	L	4	10	Gizi Lebih
158	L	5	10.9	Gizi Lebih
159	L	32	20.2	Gizi Lebih
160	L	18	16.2	Gizi Lebih
161	L	19	16	Gizi Lebih
162	L	20	17	Gizi Lebih
163	P	5	10.5	Gizi Lebih
164	P	26	18.5	Gizi Lebih
165	P	27	18.7	Gizi Lebih
166	P	28	19	Gizi Lebih
167	P	58	29	Gizi Lebih
168	P	59	30	Gizi Lebih
169	P	60	30.4	Gizi Lebih

170	P	44	24.5	Gizi Lebih
171	P	45	24.3	Gizi Lebih
172	P	46	25	Gizi Lebih
173	P	16	15	Gizi Lebih
174	P	17	15.7	Gizi Lebih
175	L	43	23.5	Gizi Lebih
176	L	44	21	Gizi Lebih
177	L	6	11.6	Gizi Lebih
178	L	7	12	Gizi Lebih
179	L	8	12.9	Gizi Lebih
180	L	51	26	Gizi Lebih
181	L	52	26.3	Gizi Lebih
182	P	38	21.5	Gizi Lebih
183	L	60	27.8	Gizi Lebih
184	P	41	22.8	Gizi Lebih
185	L	29	19	Gizi Lebih
186	L	30	19.4	Gizi Lebih
187	L	27	18.2	Gizi Lebih
188	L	46	24	Gizi Lebih
189	L	0	2.9	Gizi Baik
190	L	1	5	Gizi Baik
191	L	5	7	Gizi Baik
192	L	6	8.5	Gizi Baik
193	L	7	7.2	Gizi Baik
194	L	9	9	Gizi Baik
195	L	10	9.1	Gizi Baik
196	L	11	9	Gizi Baik
197	L	12	9.5	Gizi Baik
198	L	15	11	Gizi Baik
199	L	16	10	Gizi Baik
200	L	17	11	Gizi Baik
201	L	18	12	Gizi Baik
202	L	19	10.2	Gizi Baik
203	L	23	11	Gizi Baik
204	L	24	12	Gizi Baik
205	L	25	13	Gizi Baik
206	L	26	11.7	Gizi Baik
207	L	27	13	Gizi Baik
208	L	28	12	Gizi Baik
209	L	29	13	Gizi Baik
210	L	30	14	Gizi Baik
211	L	31	14.3	Gizi Baik
212	L	32	11	Gizi Baik

213	L	33	14	Gizi Baik
214	L	37	14.5	Gizi Baik
215	L	39	15	Gizi Baik
216	L	42	15	Gizi Baik
217	L	43	16	Gizi Baik
218	L	44	15	Gizi Baik
219	L	45	14.2	Gizi Baik
220	L	47	14.2	Gizi Baik
221	L	50	16	Gizi Baik
222	L	51	15	Gizi Baik
223	L	52	15.3	Gizi Baik
224	L	54	15.6	Gizi Baik
225	L	56	16	Gizi Baik
226	L	58	16	Gizi Baik
227	L	60	16	Gizi Baik
228	P	15	10.2	Gizi Baik
229	P	16	11	Gizi Baik
230	P	18	11	Gizi Baik
231	P	19	12	Gizi Baik
232	P	21	12	Gizi Baik
233	P	27	13.2	Gizi Baik
234	P	28	14	Gizi Baik
235	P	29	13	Gizi Baik
236	P	30	12.1	Gizi Baik
237	P	31	11	Gizi Baik
238	P	32	14	Gizi Baik
239	P	34	12	Gizi Baik
240	P	35	12.6	Gizi Baik
241	P	41	15	Gizi Baik
242	P	42	15.1	Gizi Baik
243	P	44	16	Gizi Baik
244	P	47	16	Gizi Baik
245	P	51	15.3	Gizi Baik
246	P	58	15	Gizi Baik
247	P	59	16	Gizi Baik
248	L	2	8.7	Gizi Baik
249	L	2	5	Gizi Baik
250	L	3	5.6	Gizi Baik
251	L	4	6.5	Gizi Baik
252	L	20	12.1	Gizi Baik
253	L	21	11	Gizi Baik
254	L	22	12	Gizi Baik
255	P	22	10.7	Gizi Baik

256	P	23	12	Gizi Baik
257	P	48	16.4	Gizi Baik
258	L	13	10	Gizi Baik
259	L	14	10.3	Gizi Baik
260	L	34	11.8	Gizi Baik
261	L	35	13	Gizi Baik
262	L	36	15	Gizi Baik
263	P	38	15.8	Gizi Baik
264	P	36	14	Gizi Baik
265	P	37	15	Gizi Baik
266	P	24	11	Gizi Baik
267	P	25	13	Gizi Baik
268	P	33	13	Gizi Baik
269	L	29	13	Gizi Baik
270	L	30	14	Gizi Baik
271	P	19	12	Gizi Baik
272	L	8	8	Gizi Baik
273	P	26	10.1	Gizi Baik
274	L	26	8.7	Gizi Buruk
275	L	27	8.5	Gizi Buruk
276	L	29	9	Gizi Buruk
277	L	30	9.1	Gizi Buruk
278	L	31	9.4	Gizi Buruk
279	L	32	9.5	Gizi Buruk
280	L	33	8.8	Gizi Buruk
281	L	34	9	Gizi Buruk
282	L	35	9.6	Gizi Buruk
283	L	36	9.5	Gizi Buruk
284	L	49	10.8	Gizi Buruk
285	L	53	11	Gizi Buruk
286	L	54	11.5	Gizi Buruk
287	L	55	11.3	Gizi Buruk
288	L	56	11.7	Gizi Buruk
289	L	57	11	Gizi Buruk
290	L	59	11.9	Gizi Buruk
291	L	60	12	Gizi Buruk
292	P	28	8.3	Gizi Buruk
293	P	29	8.1	Gizi Buruk
294	P	30	8.5	Gizi Buruk
295	P	31	8.7	Gizi Buruk
296	P	32	7.7	Gizi Buruk
297	P	33	8.2	Gizi Buruk
298	P	34	9	Gizi Buruk

299	P	35	8.4	Gizi Buruk
300	P	36	9	Gizi Buruk
301	P	37	9	Gizi Buruk
302	P	38	9.4	Gizi Buruk
303	P	43	9.4	Gizi Buruk
304	P	45	9.6	Gizi Buruk
305	L	38	9.9	Gizi Buruk
306	L	50	10.9	Gizi Buruk
307	L	51	10.5	Gizi Buruk
308	L	52	11.5	Gizi Buruk
309	L	28	8.9	Gizi Buruk
310	P	42	9.7	Gizi Buruk
311	P	25	7.9	Gizi Buruk
312	P	26	8	Gizi Buruk
313	P	27	8	Gizi Buruk
314	P	39	9	Gizi Buruk
315	P	40	9.5	Gizi Buruk
316	L	58	12	Gizi Buruk
317	L	59	11.9	Gizi Buruk
318	L	60	12	Gizi Buruk
319	P	36	9	Gizi Buruk
320	P	37	9	Gizi Buruk
321	L	53	11	Gizi Buruk
322	L	25	8	Gizi Buruk



LAMPIRAN B TABEL STANDAR ANTROPOMETRI

Tabel 1
Standar Berat Badan Menurut umur (BB/U)
Anak laki-laki umur 0-60 bulan

Umur	3sd	2sd	1sd	-	-1sd	-2sd	-3sd
0	5.0	4.4	3.9	3.3	2.9	2.5	2.1
1	6.6	5.8	5.1	4.5	3.9	3.4	2.9
2	5.0	7.1	6.3	5.6	4.9	4.3	3.8
2	8.0	7.1	6.3	5.6	4.9	4.3	3.8
3	9.0	8.0	7.2	6.4	5.7	5.0	4.4
4	9.7	8.7	7.8	7.0	6.2	5.6	4.9
5	10.4	9.3	8.4	7.5	6.7	6.0	5.3
6	10.9	9.8	8.8	7.9	7.1	6.4	5.7
7	11.4	10.3	9.2	8.3	7.4	6.7	5.9
8	11.9	10.7	9.6	8.6	7.7	6.9	6.2
9	12.3	11.0	9.9	8.9	8.0	7.1	6.4
10	12.7	11.4	10.2	9.2	8.2	7.4	6.6
11	13.0	11.7	10.5	9.4	8.4	7.6	6.8
12	13.3	12.0	10.8	9.6	8.6	7.7	6.9
13	13.7	12.3	11.0	9.9	8.8	7.9	7.1
14	14.0	12.6	11.3	10.1	9.0	8.1	7.2
15	14.3	12.8	11.5	10.3	9.2	8.3	7.4
16	14.6	13.1	11.7	10.5	9.4	8.4	7.5
17	14.9	13.4	12.0	10.7	9.6	8.6	7.7
18	15.3	13.7	12.2	10.9	9.8	8.8	7.8
19	15.6	13.9	12.5	11.1	10.0	8.9	8.0
20	15.9	14.2	12.7	11.3	10.1	9.1	8.1
21	16.2	14.5	12.9	11.5	10.3	9.2	8.2
22	16.5	14.7	13.2	11.8	10.5	9.4	8.4



23	16.8	15.0	13.4	12.0	10.7	9.5	8.5
24	17.1	15.3	13.6	12.2	10.8	9.7	8.6
25	17.5	15.5	13.9	12.4	11.0	9.8	8.8
26	17.8	15.8	14.1	12.5	11.2	10.0	8.9
27	18.1	16.1	14.3	12.7	11.3	10.1	9.0
28	18.4	16.3	14.5	12.9	11.5	10.2	9.1
29	18.7	16.6	14.8	13.1	11.7	10.4	9.2
30	19.0	16.9	15.0	13.3	11.8	10.5	9.4
31	19.3	17.1	15.2	13.5	12.0	10.7	9.5
32	19.6	17.4	15.4	13.7	12.1	10.8	9.6
33	19.9	17.6	15.6	13.8	12.3	10.9	9.7
34	20.2	17.8	15.8	14.0	12.4	11.0	9.8
35	20.4	18.1	16.0	14.2	12.6	11.2	9.9
36	20.7	18.3	16.2	14.3	12.7	11.3	10.0
37	21.0	18.6	16.4	14.5	12.9	11.4	10.1
38	21.3	18.8	16.6	14.7	13.0	11.5	10.2
39	21.6	19.0	16.8	14.8	13.1	11.6	10.3
40	21.9	19.3	17.0	15.0	13.3	11.8	10.4
41	22.1	19.5	17.2	15.2	13.4	11.9	10.5
42	22.4	19.7	17.4	15.3	13.6	12.0	10.6
43	22.7	20.0	17.6	15.5	13.7	12.1	10.7
44	23.0	20.2	17.8	15.7	13.8	12.2	10.8
45	23.3	20.5	18.0	15.8	14.0	12.4	10.9
46	23.6	20.7	18.2	16.0	14.1	12.5	11.0
47	23.9	20.9	18.4	16.2	14.3	12.6	11.1
48	24.2	21.2	18.6	16.3	14.4	12.7	11.2
49	24.5	21.4	18.8	16.5	14.5	12.8	11.3
50	24.8	21.7	19.0	16.7	14.7	12.9	11.4

51	25.1	21.9	19.2	16.8	14.8	13.1	11.5
52	25.4	22.2	19.4	17.0	15.0	13.2	11.6
53	25.7	22.4	19.6	17.2	15.1	13.3	11.7
54	26.0	22.7	19.8	17.3	15.2	13.4	11.8
55	26.3	22.9	20.0	17.5	15.4	13.5	11.9
56	26.6	23.2	20.2	17.7	15.5	13.6	12.0
57	26.9	23.4	20.4	17.8	15.6	13.7	12.1
58	27.2	23.7	20.6	18.0	15.8	13.8	12.2
59	27.6	23.9	20.8	18.2	15.9	14.0	12.3
60	27.9	24.2	21.0	18.3	16.0	14.1	12.4

Tabel 2
Standar Berat Badan Menurut umur (BB/U)
Anak Perempuan umur 0-60 bulan

Umur	3sd	2sd	1sd	-	-1sd	-2sd	-3sd
0	4.8	4.2	3.7	3.2	2.8	2.4	2.0
1	6.2	5.5	4.8	4.2	3.6	3.2	2.7
2	7.5	6.6	5.9	5.1	4.5	3.9	3.4
3	8.5	7.5	6.6	5.8	5.2	4.5	4.0
4	9.3	8.2	7.3	6.4	5.7	5.0	4.4
5	10.0	8.8	7.8	6.9	6.1	5.4	4.8
6	10.6	9.3	8.2	7.3	6.5	5.7	5.1
7	11.1	9.8	8.6	7.6	6.8	6.0	5.3
8	11.6	10.2	9.0	7.9	7.0	6.3	5.6
9	12.0	10.5	9.3	8.2	7.3	6.5	5.8
10	12.4	10.9	9.6	8.5	7.5	6.7	5.9
11	12.8	11.2	9.9	8.7	7.7	6.9	6.1
12	13.1	11.5	10.1	8.9	7.9	7.0	6.3
13	13.5	11.8	10.4	9.2	8.1	7.2	6.4

14	13.8	12.1	10.6	9.4	8.3	7.4	6.6
15	14.1	12.4	10.9	9.6	8.4	7.6	6.7
16	14.5	12.6	11.1	9.8	8.7	7.7	6.9
17	14.8	12.9	11.4	10.0	8.9	7.9	7.0
18	15.1	13.2	11.6	10.2	9.1	8.1	7.2
19	15.4	13.5	11.8	10.4	9.2	8.2	7.3
20	15.7	13.7	12.1	10.6	9.4	8.4	7.5
21	16.0	14.0	12.3	10.9	9.6	8.6	7.6
22	16.4	14.3	12.5	11.1	9.8	8.7	7.8
23	16.7	14.6	12.8	11.3	10.0	8.9	7.9
24	17.0	14.8	13.0	11.5	10.2	9.0	8.1
25	17.3	15.1	13.3	11.7	10.3	9.2	8.2
26	17.7	15.4	13.5	11.9	10.5	9.4	8.4
27	18.0	15.7	13.7	12.1	10.7	9.5	8.5
28	18.3	16.0	14.0	12.3	10.9	9.7	8.6
29	18.7	16.2	14.2	12.5	11.1	9.8	8.8
30	19.0	16.5	14.4	12.7	11.2	10.0	8.9
31	19.3	16.8	14.7	12.9	11.4	10.1	9.0
32	19.6	17.1	14.9	13.1	11.6	10.3	9.1
33	20.0	17.3	15.1	13.3	11.7	10.4	9.3
34	20.3	17.6	15.4	13.5	11.9	10.5	9.4
35	20.6	17.9	15.6	13.7	12.0	10.7	9.5
36	20.9	18.1	15.8	13.9	12.2	10.8	9.6
37	21.3	18.4	16.0	14.0	12.4	10.9	9.7
38	21.6	18.7	16.3	14.2	12.5	11.1	9.8
39	22.0	19.0	16.5	14.4	12.7	11.2	9.9
40	22.3	19.2	16.7	14.6	12.8	11.3	10.1
41	22.7	19.5	16.9	14.8	13.0	11.5	10.2

42	23.0	19.8	17.2	15.0	13.1	11.6	10.3
43	23.4	20.1	17.4	15.2	13.3	11.7	10.4
44	23.7	20.4	17.6	15.3	13.4	11.8	10.5
45	24.1	20.7	17.8	15.5	13.6	12.0	10.6
46	24.5	20.9	18.1	15.7	13.7	12.1	10.7
47	24.8	21.2	18.3	15.9	13.9	12.2	10.8
48	25.2	21.5	18.5	16.1	14.0	12.3	10.9
49	25.5	21.8	18.8	16.3	14.2	12.4	11.0
50	25.9	22.1	19.0	16.4	14.3	12.6	11.1
51	26.3	22.4	19.2	16.6	14.5	12.7	11.2
52	26.6	22.6	19.4	16.8	14.6	12.8	11.3
53	27.0	22.9	19.7	17.0	14.8	12.9	11.4
54	27.4	23.2	19.9	17.2	14.9	13.0	11.5
55	27.7	23.5	20.1	17.3	15.1	13.2	11.6
56	28.1	23.8	20.3	17.5	15.2	13.3	11.7
57	28.5	24.1	20.6	17.7	15.3	13.4	11.8
58	28.8	24.4	20.8	17.9	15.5	13.5	11.9
59	29.2	24.6	21.0	18.0	15.6	13.6	12.0
60	29.5	24.9	21.2	18.2	15.8	13.7	12.1