

**PREDIKSI TINGKAT PEMAHAMAN SISWA DALAM
MATERI PELAJARAN BAHASA INDONESIA MENGGUNAKAN
NAÏVE BAYES DENGAN SELEKSI FITUR *INFORMATION GAIN***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Siti Utami Fhlayli

NIM: 155150201111296



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PREDIKSI TINGKAT PEMAHAMAN SISWA DALAM MATERI PELAJARAN BAHASA
INDONESIA MENGGUNAKAN *NAÏVE BAYES* DENGAN SELEKSI FITUR
INFORMATION GAIN

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Siti Utami Fhylayli
NIM: 155150201111296

Skripsi ini telah diuji dan dinyatakan lulus pada
27 Desember 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Budi Darma Setiawan, S.Kom, M.Cs
NIP: 19841015 201404 1 002

Ir. Sutrisno, M.T
NIP: 19570325 198701 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

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

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

Malang, 12 Desember 2018

Siti Utami Fhylayli

NIM: 155150201111296



PRAKATA

Segala puji dan syukur ke hadirat Allah SWT yang telah melimpahkan rahmat, hidayah, serta karunia-Nya, sehingga penulis dapat menyelesaikan skripsi “Prediksi Tingkat Pemahaman Siswa Dalam Materi Pelajaran Bahasa Indonesia Menggunakan *Naïve Bayes* Dengan Seleksi Fitur *Information Gain*”. Penulis ingin mengucapkan terimakasih kepada semua pihak yang telah memberikan bantuan, motivasi dan dukungan kepada penulis dalam menyelesaikan skripsi ini, yang diantaranya:

1. Kedua orang tua, adik dan keluarga besar penulis yang telah memberikan do’a, motivasi, semangat dan dukungan yang sangat bermanfaat bagi penulis selama menempuh Pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya dan selama penyelesaian skripsi ini.
2. Seluruh dosen Fakultas Ilmu Komputer yang telah memberikan wawasan, bantuan, serta ilmu yang bermanfaat bagi penulis selama menempuh Pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya sehingga mampu menyelesaikan skripsi ini.
3. Budi Darma Setiawan, S.Kom, M.Cs selaku dosen pembimbing pertama yang telah memberikan bimbingan, arahan, saran, kemudahan, serta motivasi kepada penulis selama penyusunan skripsi ini.
4. Ir. Sutrisno, M.T selaku dosen pembimbing kedua yang telah membimbing, memberikan saran dan motivasi kepada penulis selama penyusunan skripsi ini.
5. Seluruh civitas akademika dan staff Universitas Brawijaya yang telah memberikan fasilitas, pelayanan yang baik, sarana dan prasarana yang sangat membantu dalam penyelesaian skripsi ini.
6. Keluarga besar Asisten Laboratorium Sistem Komputer dan Robotika Fakultas Ilmu Komputer yang setiap saat mengingatkan progress pengerjaan skripsi, info mengenai tahapan penyelesaian skripsi, memberikan pembelajaran dan ilmu yang bermanfaat serta dukungan tak terkira kepada penulis.
7. Teman-teman Anonyminus dan Abang yang membantu dalam pengerjaan skripsi yang selalu memberikan berbagai bentuk dukungan, hiburan, serta motivasi kepada penulis agar penulis segera menyelesaikan progres dan segera pulang ke keampung halaman.
8. Pihak-pihak lainnya yang membantu penulis hingga terselesaikannya skripsi ini.

Penulis menyadari dalam penyusunan skripsi ini masih banyak terdapat kesalahan dan kekurangan. Sebagai pedoman dalam perbaikan, penulis dengan terbuka menerima kritik dan saran yang bersifat membangun sehingga laporan skripsi ini dapat memberikan manfaat lebih bagi penulis serta pembacanya.

Malang, 12 Desember 2018

Penulis

utamifhylayli@student.ub.ac.id



ABSTRAK

Siti Utami Fhlayli, Prediksi Tingkat Pemahaman Siswa Dalam Materi Pelajaran Bahasa Indonesia Menggunakan *Naïve Bayes* Dengan Seleksi Fitur *Information Gain*

Pembimbing: Budi Darma Setiawan, S.Kom, M.Cs dan Ir. Sutrisno, M.T.

Mata pelajaran Bahasa Indonesia secara umum dianggap sebagai pelajaran yang mudah dan tidak perlu dipelajari secara serius oleh kebanyakan siswa dan masyarakat. Mengakibatkan kemampuan setiap siswa dalam memahami pelajaran tersebut berbeda-beda, sehingga proses belajar siswa pun akan berbeda pula. Hal ini menyebabkan pengajar memiliki keterbatasan untuk mengukur tingkat pemahaman siswa, maka diperlukan sistem untuk untuk memprediksi tingkat pemahaman siswa. Prediksi ini menggunakan metode klasifikasi dengan algoritme *Naïve Bayes*. Kelas yang akan digunakan pada penelitian ini diantaranya siswa sangat paham, cukup paham dan kurang paham. Fitur dari klasifikasi berupa faktor personal, sosial, lingkungan, psikologis. Selain bertujuan mendapatkan nilai akurasi yang baik juga bertujuan mendapatkan model fitur dengan cara menerapkan seleksi fitur. Pada penelitian ini penulis menggunakan seleksi fitur *Information Gain* (IG). Tiap-tiap Fitur dari data set dihitung nilai *gain*nya dan dipilih fitur mana saja yang memiliki nilai *gain* sesuai threshold. Fitur yang terpilih akan diproses dengan algoritme klasifikasi naive bayes kemudian dilihat akurasi jika belum maksimal maka dilakukan kembali proses seleksi fitur tadi hingga mendapatkan akurasi yang diinginkan. Dari pengujian yang sudah dilakukan, didapatkan hasil bahwa fitur yang memiliki nilai *Gain* lebih dari 0.2 memiliki akurasi terbesar yaitu mencapai 90%. Fitur yang terpilih yaitu fitur jumlah anggota keluarga, status tempat tinggal, pekerjaan ibu, pengasuh, dukungan keluarga, ikut ekstrakurikuler, mengulang pelajaran di rumah, lama belajar di rumah, jenis bacaan di rumah, lama membaca di rumah.

Kata kunci: klasifikasi, naive bayes, seleksi fitur, information gain, pemahaman, bahasa Indonesia.

ABSTRACT

Siti Utami Fhylayli, Prediction of Level of Understanding of Students in Indonesian Language Materials Using Naïve Bayes by Selection of Information Gain Features

Supervisors: Budi Darma Setiawan, S.Kom, M.Cs dan Ir. Sutrisno, M.T.

Indonesian Language Subjects are generally considered to be easy lessons and do not need to be learned by most students and societies. Resulting in the ability of each student to understand the different lessons, the process of the student dropping will also be different. This causes the instructor to have limitations in measuring the level of understanding of students, so a system is needed to predict the level of understanding of students. This prediction uses the classification method with the Naïve Bayes algorithm. The class that will be used in this study is very understanding, quite understanding and lacking in understanding. Features of various personal, social, environmental, psychological factors. In addition, you can also use feature features by running the feature. At this time the author uses the Information Gain (IG) feature flow. Each feature of the data set is calculated and taken which features have the gain value according to the threshold. The features that will be used with the Naive Bayes classification algorithm next to its Accuracy if not maximal, then done. From the credit that has been done, get the results of features that have a Gain value of more than 0.2 having the greatest accuracy, reaching 90%. The features used are features of family members, residence status, mother's work, caregivers, family, taking extracurricular activities, repeating lessons at home, length of study at home, reading at home, reading time at home.

Keywords: classification, naive bayes, feature selection, information gain, understanding, Indonesian language.

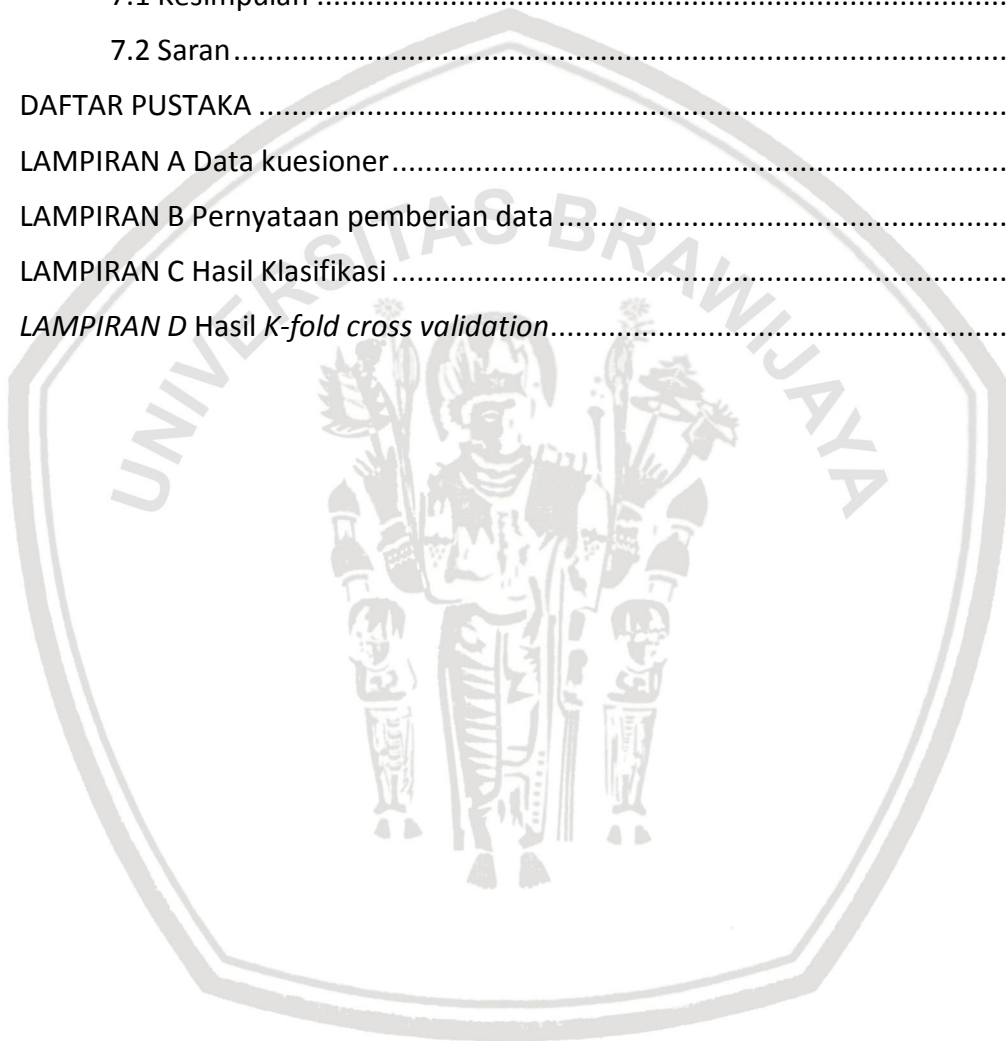
DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS.....	iii
PRAKATA.....	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xiii
DAFTAR KODE PROGRAM.....	xiv
DAFTAR LAMPIRAN.....	xv
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 Landasan teori.....	7
2.2.1 Prediksi	7
2.2.2 Pemahaman.....	8
2.2.3 Klasifikasi	9
2.2.4 <i>Naïve Bayes</i>	9
2.2.5 Seleksi Fitur	10
2.2.6 <i>Information Gain</i>	10
2.2.7 Siswa	11
BAB 3 METODOLOGI	12
3.1 Tipe Penelitian	12

3.2 Strategi Penelitian	12
3.2.1 Peralatan yang digunakan.....	12
3.2.2 Objek Penelitian.....	13
3.2.3 Prosedur Pengumpulan Data	13
3.2.4 Implementasi.....	13
3.2.5 Gambaran Umum Sistem	13
3.2.6 Penjelasan Umum Algoritme.....	14
3.2.7 Proses Algoritme.....	15
3.2.8 Pengolahan Data.....	16
3.2.9 Pengujian dan analisis.....	17
BAB 4 PERANCANGAN.....	20
4.1 Formulasi Permasalahan	20
4.2 Diagram alir.....	20
4.2.1 Seleksi Fitur <i>Information Gain</i>	21
4.2.2 Metode Naïve Bayes	32
4.2.3 Menghitung akurasi dengan <i>Confusion Matrix</i>	39
4.3 Manualisasi	40
4.3.1 Seleksi Fitur <i>Information Gain</i>	44
4.3.2 Metode Naïve Bayes	45
4.3.3 Pengujian akurasi.....	47
BAB 5 IMPLEMENTASI.....	49
5.1 Kode Program	49
5.1.1 Seleksi Fitur <i>Information Gain</i>	49
5.1.2 Klasifikasi <i>Naïve bayes</i>	53
5.2 <i>Screen Shot</i> Program	56
BAB 6 PENGUJIAN DAN ANALISIS	58
6.1 <i>K-Fold Cross Validation</i>	58
6.2 Uji Coba Nilai <i>Gain</i> ≥ 0	59
6.3 Uji Coba Nilai <i>Gain</i> ≥ 0.1	60
6.4 Uji Coba Nilai <i>Gain</i> ≥ 0.2	60
6.5 Uji Coba Nilai <i>Gain</i> ≥ 0.3	61
6.6 Uji Coba Nilai <i>Gain</i> ≥ 0.4	62



6.7 Uji Coba Nilai <i>Gain</i> ≥ 0.5	63
6.8 Uji Coba Nilai <i>Gain</i> $\geq 0.6, 0.7, 0.8, 0.9, 1$	64
6.9 Uji Coba Nilai <i>Gain</i> $\geq 1.1, 1.2, 1.3, 1.4$	65
6.10 Uji Coba Nilai <i>Gain</i> ≥ 1.5	66
6.11 Analisis Sistem	67
BAB 7 PENUTUP	69
7.1 Kesimpulan	69
7.2 Saran	69
DAFTAR PUSTAKA	70
LAMPIRAN A Data kuesioner	73
LAMPIRAN B Pernyataan pemberian data	77
LAMPIRAN C Hasil Klasifikasi	78
LAMPIRAN D Hasil <i>K-fold cross validation</i>	87



DAFTAR TABEL

Tabel 3.1 Fitur dan Kategori yang digunakan.....	16
Tabel 3.2 <i>Pengujian K-fold cross validation</i>	18
Tabel 3.3 <i>Confusion Matrix</i>	18
Tabel 4.1 Data <i>Training</i>	40
Tabel 4.2 Data <i>Testing</i>	42
Tabel 4.3 <i>Information Gain</i>	44
Tabel 4.4 Probabilitas P(H)	44
Tabel 4.5 Probabilitas P(H)	46
Tabel 4.6 Probabilitas P(X H) dengan <i>Smoothing</i>	46
Tabel 4.7 Hasil Klasifikasi.....	47
Tabel 4.8 Akurasi.....	47
Tabel 4.9 <i>Confusion Matrix</i>	48
Tabel 6.1 <i>K-fold Cross Validation</i>	58
Tabel 6.2 <i>Confusion Matrix Gain</i> ≥ 0	59
Tabel 6.3 Akurasi <i>Gain</i> ≥ 0	59
Tabel 6.4 <i>Confusion Matrix</i> ≥ 0.1	60
Tabel 6.5 Akurasi <i>Gain</i> ≥ 0.1	60
Tabel 6.6 <i>Confusion Matrix Gain</i> ≥ 0.2	61
Tabel 6.7 Akurasi <i>Gain</i> ≥ 0.2	61
Tabel 6.8 <i>Confusion Matrix Gain</i> ≥ 0.3	62
Tabel 6.9 Akurasi <i>Gain</i> ≥ 0.3	62
Tabel 6.10 <i>Confusion Matrix Gain</i> ≥ 0.4	63
Tabel 6.11 Akurasi <i>Gain</i> ≥ 0.4	63
Tabel 6.12 <i>Confusion Matrix Gain</i> ≥ 0.5	64
Tabel 6.13 Akurasi <i>Gain</i> ≥ 0.5	64
Tabel 6.14 <i>Confusion Matrix Gain</i> $\geq 0.6, 0.7, 0.8, 0.9, 1$	65
Tabel 6.15 Akurasi <i>Gain</i> $\geq 0.6, 0.7, 0.8, 0.9, 1$	65
Tabel 6.16 <i>Confusion Matrix Gain</i> $\geq 1.1, 1.2, 1.3, 1.4$	66
Tabel 6.17 Akurasi <i>Gain</i> $\geq 1.1, 1.2, 1.3, 1.4$	66
Tabel 6.18 <i>Confusion Matrix Gain</i> ≥ 1.5	67

Tabel 6.19 Akurasi *Gain* ≥ 1.5 67
Tabel 7.1 Data kuesioner 73

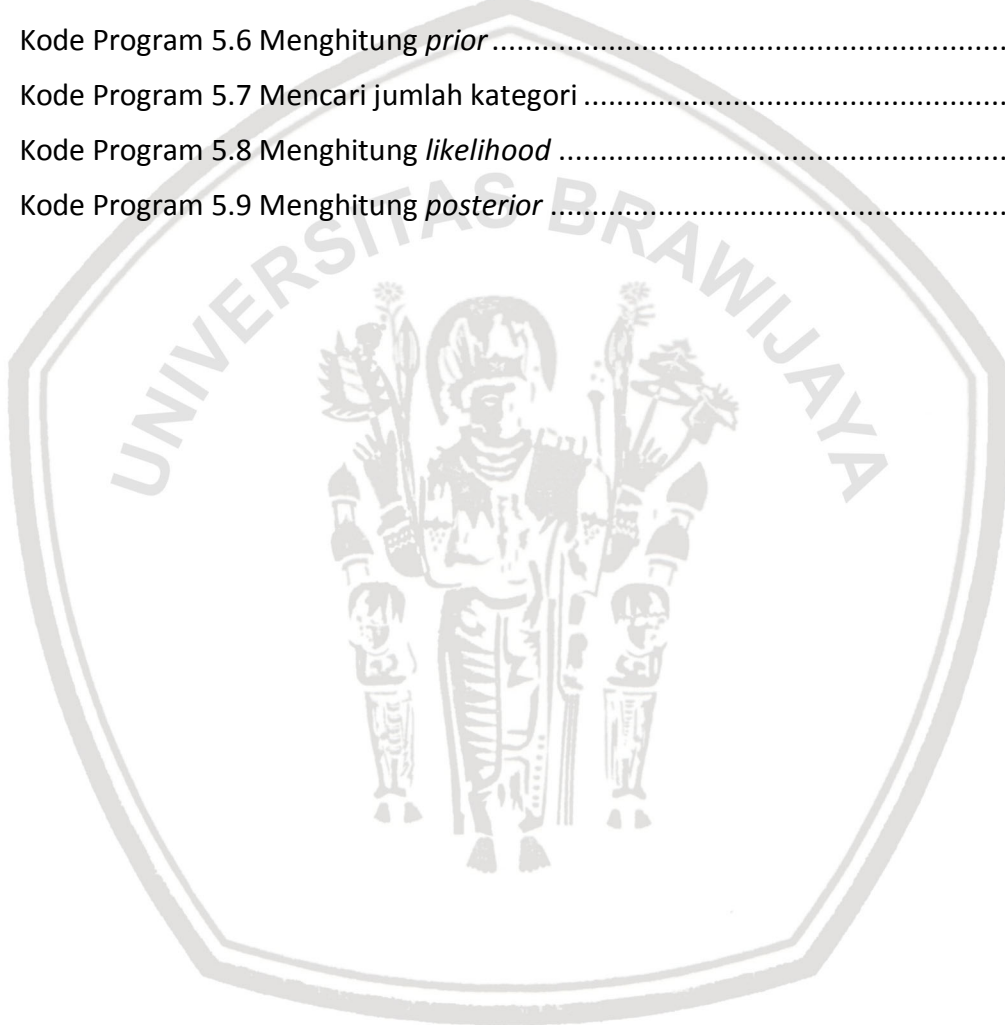


DAFTAR GAMBAR

Gambar 3.1 Diagram alir Keseluruhan	14
Gambar 4.1 Diagram alir keseluruhan	20
Gambar 4.2 Diagram alir Seleksi <i>Information Gain</i>	21
Gambar 4.3 Diagram alir Menghitung <i>Entropy</i>	23
Gambar 4.4 Diagram alir Menghitung <i>Entropy</i> semua kelas	24
Gambar 4.5 Diagram alir Pengukuran <i>Entropy</i> semua fitur	25
Gambar 4.6 Diagram alir Mencari probabilitas fitur	26
Gambar 4.7 Diagram alir menghitung <i>likelihood</i>	27
Gambar 4.8 Diagram alir menghitung <i>entropy</i> kategori	28
Gambar 4.9 Diagram alir Menghitung nilai <i>entropy</i> fitur	29
Gambar 4.10 Diagram alir Menghitung nilai <i>Gain</i>	30
Gambar 4.11 Diagram alir menentukan <i>threshold</i>	31
Gambar 4.12 Diagram alir Metode <i>Naïve Bayes</i>	32
Gambar 4.13 Diagram alir Proses <i>Training</i>	33
Gambar 4.14 Diagram alir menghitung jumlah kelas tertentu	34
Gambar 4.15 Diagram alir Menghitung <i>Prior</i>	34
Gambar 4.17 Diagram alir Mencari kategori unik	35
Gambar 4.18 Diagram alir menghitung <i>likelihood</i> dengan <i>smoothing</i>	36
Gambar 4.19 Diagram alir Proses <i>Testing</i>	37
Gambar 4.20 Diagram alir menghitung <i>posterior</i>	39
Gambar 4.21 Diagram alir <i>Confusion Matrix</i>	39
Gambar 5.1 Fitur yang terpilih	56
Gambar 5.2 Hasil Klasifikasi	56
Gambar 5.3 <i>Confusion Matrix</i>	57
Gambar 6.1 Diagram <i>Confusion Matrix</i>	67

DAFTAR KODE PROGRAM

Kode Program 5.1 Menghitung nilai <i>entropy</i> dari semua kelas	49
Kode Program 5.2 Menghitung <i>entropy</i> dari semua fitur.....	51
Kode Program 5.3 Menghitung nilai <i>Gain</i>	52
Kode Program 5.4 Menentukan <i>threshold</i>	52
Kode Program 5.5 Menghitung jumlah kelas tertentu	53
Kode Program 5.6 Menghitung <i>prior</i>	53
Kode Program 5.7 Mencari jumlah kategori	54
Kode Program 5.8 Menghitung <i>likelihood</i>	54
Kode Program 5.9 Menghitung <i>posterior</i>	55



DAFTAR LAMPIRAN

LAMPIRAN A Data kuesioner.....	73
LAMPIRAN B Pernyataan pemberian data.....	77
LAMPIRAN C Hasil Klasifikasi.....	78
LAMPIRAN D Hasil <i>K-fold cross validation</i>	87



BAB 1 PENDAHULUAN

1.1 Latar belakang

Sekolah merupakan salah satu tempat bagi para siswa untuk mendapatkan pendidikan. Tujuan pendidikan, secara umum yaitu untuk memberikan peningkatan terhadap kecerdasan bangsa, dengan cara meningkatkan pemahaman siswa terhadap mata pelajaran yang diajarkan. Kemampuan setiap siswa dalam memahami pelajaran berbeda-beda, sehingga proses belajar siswa pun akan berbeda pula. Pengajar berperan penting dalam proses belajar. Agar pengajar dapat menentukan cara pembelajaran yang dapat di terima oleh semua siswa, maka diperlukan metode prediksi tingkat pemahaman siswa. Seorang siswa dikatakan memahami sesuatu apabila siswa tersebut telah dapat menjelaskan kembali atau mampu menguraikan suatu materi yang telah dipelajari tersebut lebih rinci menggunakan bahasanya sendiri.

Tingkat pemahaman siswa dalam memahami setiap mata pelajaran berbeda-beda sesuai dengan minat siswa tersebut. Mata pelajaran Bahasa Indonesia secara umum dianggap sebagai pelajaran yang mudah dan tidak perlu dipelajari secara serius oleh kebanyakan siswa dan masyarakat. Hal ini berbeda dengan mata pelajaran lain yang membutuhkan hafalan atau perhitungan secara logika. Berdasarkan hal tersebut, muncul berbagai masalah pembelajaran Bahasa Indonesia yang ditimbulkan oleh sikap para siswa dan masyarakat terhadap pelajaran Bahasa Indonesia. Dari sikap seperti ini, kemudian timbul berbagai masalah yang melibatkan pengajar, mata pelajaran Bahasa Indonesia, siswa yang menerima pelajaran, metode mengajar, sarana-prasarana, cara mengevaluasi, serta tujuan pelajaran Bahasa Indonesia (Moeljono, 1989).

Saat ini sistem belum tercapai dengan baik untuk menganalisa kemampuan siswa memahami materi pembelajaran. Ada dua alasan mengapa hal ini terjadi, pertama karena penelitian tentang metode prediksi masih belum cukup mengidentifikasi metode yang paling tepat untuk memprediksi kemampuan pemahaman siswa. Kedua karena kurangnya investigasi terhadap faktor-faktor yang mempengaruhi kemampuan pemahaman siswa. (Shahiria, et al., 2015)

Salah satu teknik prediksi dapat menggunakan metode klasifikasi. Klasifikasi memetakan data menjadi kelompok data yang telah ditentukan kelasnya. Kelas ditentukan sebelum memulai mengolah data. Kelas yang akan digunakan pada penelitian ini diantaranya siswa sangat paham, cukup paham dan kurang paham. Model prediksi tersebut mencakup seluruh faktor personal, sosial, lingkungan, psikologis yang digunakan agar prediksi tersebut efektif bagi kemampuan memahami materi pelajaran Bahasa Indonesia. Klasifikasi dilakukan dengan menggunakan algoritme *Naïve Bayes*. *Naïve Bayes* merupakan salah satu metode

machine learning yang menggunakan konsep dasar dari sebuah teori bernama Teorema Bayes, teori ini melakukan klasifikasi dengan menghitung nilai probabilitas (Trisedya & Jais, 2009). *Naïve Bayes* dalam beberapa penelitian secara empiris terbukti sangat mudah di implementasikan ke dalam berbagai studi kasus. Selain itu algoritme ini juga memiliki performa pengklasifikasian yang cukup tinggi.

Pada penelitian kali ini selain bertujuan mendapatkan nilai akurasi yang baik juga bertujuan mendapatkan model fitur dengan cara menerapkan seleksi fitur. Seleksi fitur adalah salah satu cara untuk menentukan fitur yang paling memiliki pengaruh di dalam dataset. Seleksi fitur berperan dengan memilih subset yang tepat dari set fitur asli, dikarenakan bahwa tidak semua fitur dapat relevan atau memiliki hubungan dengan studi kasus (Maimon & Rokach, 2010). Bahkan beberapa dari fitur tersebut dapat mengganggu dan dapat mengurangi akurasi sistem. *Noisy Features* atau fitur yang tidak terpakai tersebut harus dihapus untuk meningkatkan akurasi. Selain itu, pada pengklasifikasian memiliki fitur yang banyak akan memperlambat proses komputasi.

Terdapat banyak metode seleksi fitur, pada penelitian ini penulis menggunakan seleksi fitur *Information Gain* (IG). Metode ini mengukur jumlah informasi dalam bit tentang prediksi kelas, jika satu-satunya informasi yang tersedia adalah adanya fitur dan distribusi kelasnya. Konkretnya, mengukur pengurangan diharapkan dalam entropi (ketidakpastian yang berhubungan dengan fitur yang acak). *Information Gain* adalah ukuran simetris, yaitu jumlah informasi yang diperoleh X setelah mengamati Y adalah sama dengan jumlah informasi yang diperoleh Y setelah mengamati X *Simetrical* yang diinginkan untuk mengukur fitur-fitur yang saling berkorelasi. (Larose, 2005)

Penelitian terkait dengan penelitian ini adalah penilitan yang ditulis oleh Ade Ricky Rozzaqi pada tahun 2015 berjudul *Naïve Bayes* dan Filtering Feature Selection *Information Gain* untuk Prediksi Ketepatan Kelulusan Mahasiswa. Dalam penelitian ini dilakukan menggunakan dua metode yaitu metode yang hanya menggunakan algoritme klasifikasi *Naïve Bayes*, serta metode yang menggabungkan antara dua algoritme yaitu algoritme *Naïve Bayes* dan algoritme seleksi fitur *Information Gain*. Hasil penelitian menunjukkan bahwa nilai akurasi tertinggi diperoleh dengan metode yang menggabungkan antara algoritme *Naïve Bayes* dan algoritme seleksi Fitur *Information Gain* yaitu mencapai nilai hingga 89,79 % fitur yang terpilih adalah sebanyak 3 fitur, dan peningkatan AUC meningkat dengan 3 fitur tersebut. (Rozzaqi, 2015)

Berdasarkan uraian diatas, penulis melakukan penelitian untuk memprediksi kemampuan siswa dalam memahami materi pelajaran Bahasa Indonesia dengan judul “Prediksi kemampuan siswa dalam memahami materi pelajaran bahasa indonesia menggunakan klasifikasi *naïve bayes* dengan seleksi fitur *information gain*”. Prediksi ini dapat digunakan untuk mengidentifikasi fitur

penting dalam data siswa karena mengoptimalkan algoritme *Naïve Bayes* dengan menambah fitur yang sudah digunakan sebelumnya.

1.2 Rumusan masalah

Berdasarkan uraian latar belakang diatas, maka rumusan masalah untuk prediksi tingkat pemahaman siswa dalam materi pelajaran bahasa indonesia menggunakan *naïve bayes* dengan seleksi fitur *information gain* adalah sebagai berikut:

1. Fitur apa saja yang memberikan pengaruh terhadap prediksi kemampuan siswa dalam memahami materi pelajaran Bahasa Indonesia?
2. Bagaimana tingkat akurasi prediksi kemampuan siswa memahami materi pelajaran Bahasa Indonesia menggunakan metode klasifikasi *Naïve Bayes* menggunakan seleksi fitur *Information Gain*?

1.3 Tujuan

Tujuan dari penulisan penelitian ini sebagai berikut:

1. Mendapatkan fitur yang memberikan pengaruh terhadap prediksi kemampuan siswa dalam memahami materi pelajaran Bahasa Indonesia
2. Menguji tingkat akurasi terhadap prediksi kemampuan siswa memahami materi pelajaran Bahasa Indonesia dengan algoritme menggunakan metode klasifikasi *Naïve Bayes* menggunakan seleksi fitur *Information gain*

1.4 Manfaat

Manfaat penelitian ini bagi lingkungan pendidikan yaitu:

1. Siswa dapat mengetahui tingkat pemahamannya terhadap pelajaran Bahasa Indonesia yang dia dapatkan
2. Guru dapat meninjau siswa yang memiliki tingkat pemahaman yang kurang serta dapat memberikan perhatian lebih terhadap siswa tersebut

1.5 Batasan masalah

Agar tidak menyimpang dari maksud dan tujuan yang sebenarnya, maka penulis membatasi masalah pada penulisan ini menjadi sebagai berikut :

1. Data yang digunakan adalah data siswa pada SMA *Brawijaya Smart School*
2. Data nilai bersifat pribadi dan didapatkan pada tanggal 25 Oktober 2018 dan diberikan langsung oleh guru mata pelajaran Bahasa Indonesia SMA *BSS* berupa nilai ualngan harian sampai bulan ketiga

3. Data didapatkan dari pengisian kuesioner yang diisi oleh siswa SMA BSS kemudian di cocokan dengan nilai yang didapatkan dari guru berdasarkan nomor absen
4. Mata pelajaran yang akan dijadikan adalah mata pelajaran Bahasa Indonesia

1.6 Sistematika pembahasan

Gambaran secara garis besar pada laporan skripsi ini dibagi menjadi beberapa sub bab dengan sistematika penyampaian sebagai berikut :

1. BAB I PENDAHULUAN

Berisi tentang latar belakang tentang mengapa penulis mengambil masalah ini, rumusan masalah, tujuan menerapkan algoritme *Naïve Bayes* dan seleksi fitur *Information gain*, manfaat bagi bidang pendidikan, batasan masalah, dan sistematika pembahasan.

2. BAB II LANDASAN KEPUSTAKAAN

Bab ini berisi teori berupa pengertian dan definisi tentang metode dan entitas apa saja yang digunakan yang diambil dari kutipan sumber yang berkaitan dengan penulisan laporan skripsi serta beberapa literatur review yang memiliki hubungan dengan prediksi siswa menggunakan algoritme *Naïve Bayes* dan seleksi fitur *Information Gain*.

3. BAB III METODOLOGI PENELITIAN

Bab ini memaparkan tentang metode klasifikasi *Naïve Bayes* dan seleksi fitur *Information Gain* sebagai solusi penyelesaian untuk memprediksi tingkat pemahan siswa dalam memahami materi pelajaran Bahasa Indonesia

4. BAB IV PERANCANGAN

Bab ini menjelaskan formulasi permasalahan, perancangan Diagram alir metode dan langkah-langkah penyelesaian serta perhitungan manual metode seleksi dan klasifikasi.

5. BAB V IMPLEMENTASI

Bab ini berisi kode program serta penjelasan yang telah dibuat yaitu algoritme naive bayes dan seleksi fitur *information gain*.

6. BAB VI PENGUJIAN DAN ANALISIS

Bab ini menjelaskan pengujian sistem guna mendapatkan hasil yang menjadi tujuan utama dari penelitian ini dan kemudian dianalisis dan dibandingkan hasilnya

7. BAB VI PENUTUP

Bab ini berisi penjelasan tentang saran dan kesimpulan algoritme klasifikasi *Naïve Bayes* dan seleksi fitur *Information Gain*

BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepustakaan berisi kajian pustaka yaitu penelitian terdahulu yang dapat digunakan sebagai studi kepustakaan serta terdapat landasan teori yaitu uraian dan pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah, yang berkaitan dengan tema, masalah.

2.1 Kajian Pustaka

Siti Qomariah dan Andi Yushika Rangan melakukan penelitian yang berjudul “Penerapan Metode C4.5 dan *Naïve Bayes* untuk Klasifikasi Tipe Belajar Siswa Di Smkn 4 Samarinda” dalam penelitian tersebut penulis mendapatkan pengetahuan bahwa belajar yaitu proses yang ditandai dengan perubahan seperti pemahaman, pengetahuan, tingkah laku dan sikap, keterampilan, kecakapan dan lain sebagainya. Keberhasilan proses belajar dapat diindikasikan dengan berhasilnya peserta didik dalam memahami pelajaran yang telah diberikan. Dalam penelitian ini penulisnya melakukan klasifikasi tipe belajar menggunakan penerapan metode C4.5 dan *Naïve Bayes*. Dari hasil klasifikasi didapat hasil bahwa algoritme *Naïve Bayes* lebih tinggi hasil akurasinya dari metode C4.5 yaitu sebesar 4,41 %

Pada Mei 2016 Fathur Rahman dan Muhammad Iqbal Firdaus menulis penelitian yang berjudul “Penerapan *Data Mining* Metode *Naïve Bayes* untuk Prediksi Hasil Belajar Siswa Sekolah Menengah Pertama (Smp)” penelitian tersebut memberikan informasi bahwa klasifikasi merupakan metode yang paling sering digunakan untuk menyelesaikan masalah-masalah di dunia nyata. Sebagai contoh yang paling populer yaitu sistem dapat pada sistem yang menggunakan teknik ‘*machine-learning*’, klasifikasi mempelajari pola-pola dari data fitur-fitur terhadap berbagai karakteristik kelompok-kelompok yang sudah diberi label sebelumnya dengan tujuan untuk menempatkan objek baru yang tidak memiliki kelas kedalam kelas yang tertentu.

Penelitian tersebut menyatakan faktor-faktor dari dalam dan dari luar memberikan pengaruh terhadap hasil belajar siswa merupakan hal yang penting agar manfaat dari proses belajar dapat tercapai dengan optimal diantaranya faktor personal, sosial, psikologi dan variabel siswa lainnya. Fitur yang dipakai diantaranya, jumlah status tempat tinggal, pendidikan terakhir Ibu, pendidikan terakhir ayah, pekerjaan ibu, pekerjaan ayah, pengasuh, jarak rumah, dukungan keluarga, ikut ekstrakurikuler, mengulang pelajaran dirumah, posisi baris (bangku) duduk di kelas, internet dirumah, mengikuti bimbingan belajar. Faktor-faktor itulah yang akan penulis jadikan fitur untuk memprediksi kemampuan siswa dalam memahami materi pembelajaran.

Penelitian yang berjudul *Designing Scaffolding System in a Problem-Posing Learning Environment* yang ditulis oleh Ahmad Afif Supianto, Yusuke Hayashi dan Tsukasa Hirashima pada tahun 2017 melalui *3rd International Conference on Science in Information Technology (ICSITech)*. Meneliti tentang desain *e-learning* agar latihan-latihan soal yang diberikan oleh pengajar dapat sesuai dengan kemampuan siswa yang seperti kita ketahui siswa memiliki perbedaan dalam kemampuan belajarnya. Sistem yang diusulkan dilakukan dengan membentuk proses kegiatan yang menimbulkan masalah bagi pelajar, menentukan keadaan bottleneck dan menghasilkan tugas yang baru sesuai dengan kemampuan siswa. Kemudian akan diimplementasikan di kelas dan hasilnya akan di analisis. (Supianto, et al., 2017)

Penelitian dengan judul *Model-based analysis of thinking in problem posing as sentence integration focused on violation of the constraints* yang ditulis oleh Yusuke Hayashi dan Tsukasa Hirashima pada tahun 2017 melalui *Research and Practice in Technology Enhanced Learning*. Meneliti tentang bagaimana analisis berfikir siswa dengan melakukannya berdasarkan model pembelajaran yang diimplementasikan pada tool *e-learning* yang sudah dibuat. Penelitian ini melibatkan siswa sekolah dasar kelas 1 dan menganalisis kesulitan apa saja yang dialami oleh peserta didik melalui perspektif model. Selain itu terdapat penelitian serupa namun disini perbedaannya terdapat pada data yang digunakan yaitu dengan data yang jumlahnya besar (*big data*) dengan judul *Model-based approach for educational big data analysis of learners thinking with process data* yang ditulis oleh penulis yang sama. (Supianto, et al., 2017)

Penelitian selanjutnya adalah penelitian yang dibuat oleh Ade Ricky Rozzaqi pada tahun 2015 yang berjudul "*Naïve Bayes dan Filtering Feature Selection Information Gain untuk Prediksi Ketepatan Kelulusan Mahasiswa*". Dalam penelitian ini dilakukan menggunakan dua metode yaitu metode yang hanya menggunakan algoritme klasifikasi *Naïve Bayes*, serta metode yang menggabungkan antara dua algoritme yaitu algoritme *Naïve Bayes* dan algoritme seleksi fitur *Information Gain*. Hasil penelitian menunjukkan bahwa nilai akurasi tertinggi diperoleh dengan metode yang menggabungkan antara algoritme *Naïve Bayes* dan algoritme seleksi Fitur *Information Gain* yaitu mencapai nilai hingga 89,79% fitur yang terpilih adalah sebanyak 3 fitur, dan peningkatan AUC meningkat dengan 3 fitur tersebut. (Rozzaqi, 2015)

Berdasarkan kajian pustaka diatas penulis ingin memberikan hasil akurasi yang lebih tinggi dengan memilih fitur-fitur yang berbeda dan penulis akan melakukan penyeleksian fitur-fitur tersebut yang nantinya tidak semua akan digunakan. Penulis akan menghapus fitur-fitur yang dirasa tidak memiliki pengaruh bagi prediksi ini.

2.2 Landasan teori

2.2.1 Prediksi

Prediksi yaitu sebuah proses yang mengira-ngira tetapi dengan cara yang sistematis mengenai suatu hal yang sangat dapat terjadi di masa yang akan datang yang didasari oleh informasi dari masa sebelumnya dengan masa yang dimiliki sekarang, agar perkiraan dengan kenyataan yang ada memiliki selisih yang kecil. Dalam prediksi jawaban yang diberikan tidak harus selalu merupakan jawaban yang benar atau secara pasti kejadian tersebut akan benar-benar terjadi, tetapi sebisa mungkin mencari jawaban terdekat yang akan terjadi (Herdianto, 2013)).

Prediksi dapat disebut perkiraan atau ramalan. Menurut KBBI, pengertian prediksi itu sendiri hasil yang didapatkan dari kegiatan memprediksi atau melakukan peramalan atau memberikan perkiraan nilai pada masa depan menggunakan data pada masa sebelumnya. Prediksi sendiri menjelaskan apa yang bakal terjadi di satu kondisi yang merupakan masukan untuk cara pengambilan keputusan dan perencanaan. Prediksi dapat didasarkan pada metode ilmiah maupun hanya metode subjektif. Sebagai contoh, prediksi curah hujan seringkali didasarkan pada data dan informasi paling baru dari pengamatan menggunakan satelit. Contoh lain adalah prediksi gunung meletus, gempa, ataupun bencana lain. Prediksi yang termasuk prediksi secara subjektif atau dengan sudut pandang tersendiri untuk memprediksi yaitu seperti prediksi pemenang permainan sepakbola, atau olahraga lainnya.

Forecasting atau peramalan adalah suatu tatacara untuk menghasilkan suatu informasi secara sebenarnya mengenai kondisi sosial dimasa depan berdasarkan pada informasi yang sudah ada sebelumnya. Dalam peramalan ada tiga bentuk utama yaitu prediksi, perkiraan dan proyeksi.

1. Prediksi adalah sebuah ramalan yang berdasarkan pada anggapan yang tegas secara teoritis. Anggapan ini berupa hukum secara teori (misalnya hukum berkurangnya nilai uang), proposisi teoritis (misalnya proposisi bahwa perpecahan dari masyarakat sipil disebabkan oleh adanya ketidaksesuaian antara harapan dengan kemampuan), atau secara analogi (misalnya analogi antara pertumbuhan organisasi pemerintah dengan pertumbuhan organisme biologis).
2. Perkiraan (*conjecture*) merupakan peramalan berdasarkan dari pemberian nilai yang memberikan informasi atau nilai suatu pakar mengenai keadaan siswa di masa depan. Peramalan kebijakan memiliki tujuan untuk mendapatkan informasi tentang berubahnya sesuatu pada masa depan yang dapat memberikan pengaruh kepada penerapan kebijakan dan akibatnya.
3. Proyeksi yaitu ramalan berdasarkan pada kecenderungan dimasa lampau ataupun dimasa sekarang ke masa yang akan datang. Proyeksi membuat

pertanyaan yang tegas berdasarkan argument yang diperoleh dari metode tertentu dan kasus yang paralel.

2.2.2 Pemahaman

Nana Sudjana mengatakan bahwa pemahaman adalah hasil belajar, sebagai contoh siswa mampu memberikan penjelasan ulang menggunakan kalimatnya sendiri berdasarkan apa yang di lihat maupun didengarnya. Selain itu dapat memberi contoh lain selain yang diberikan oleh gurunya. Sedangkan Benjamin S. Bloom berpendapat bahwa pemahaman (*comprehension*) yaitu keahlian dalam memahami atau mengerti suatu hal setelah mengetahui dan mengingat. Sedangkan definisi pemahaman menurut Anas Sudijono yaitu keahlian seseorang dapat melihat sesuatu dari berbagai sisi dan mengerti serta mengetahuinya. Kesimpulannya adalah siswa dikatakan paham sesuatu jika siswa tersebut dapat menyajikan uraian atau penjelasan secara lebih rinci menggunakan kata-katanya sendiri. Pemahaman yaitu kemampuan berpikir yang lebih tinggi tingkatannya dibandingkan dengan hafalan dan ingatan. Gampangya, memahami adalah dapat melihat sesuatu dari berbagai segi dan mengerti mengenai sesuatu tersebut.

2.2.2.1 Tingkat pemahaman

Setiap siswa secara umum biasanya memiliki tingkat memahami sesuatu dengan berbeda-beda terhadap masing-masing materi pelajaran yang disampaikan oleh pengajar dalam suatu aktifitas belajar mengajar terutama pada pelajaran Bahasa Indonesia. Tingkat pemahaman siswa sangat perlu digali lebih dalam untuk menentukan cara belajar mengajar yang cocok untuk masa depan. Pemahaman yaitu tingkatan kemampuan yang meminta harapan siswa mampu memahami konsep atau arti, situasi serta fakta-fakta yang diketahui. (Purwanto, 1994). Dari penjelasan tentang pemahaman tadi maka kesimpulannya adalah tingkat kemampuan dalam memahami siswa adalah berapa besar siswa dapat memahami fakta yang diketahuinya dan menjelaskannya kembali secara sistematis. Tingkat pemahaman siswa biasanya berbeda-beda tergantung beberapa faktor internal dari dalam diri seorang siswa itu sendiri contohnya kemampuan berfikir secara kritis dan teoritis, IQ siswa serta cara belajar dia sendiri. Selain faktor internal, faktor dari luar juga memberikan pengaruh terhadap pemahaman siswa, seperti proses belajar, metode mengajar guru serta faktor-faktor dari lingkungan siswa tersebut baik di sekolah maupun dilingkungan rumah.

Penyebab rendahnya tingkat pemahaman siswa terhadap mata pelajaran Bahasa Indonesia adalah metode dikarenakan mata pelajaran Bahasa Indonesia sendiri yang tergolong cukup mudah, sehingga siswa merasa tidak perlu banyak belajar materi pelajaran Bahasa Indonesia.

2.2.3 Klasifikasi

Klasifikasi berasal dari bahasa Latin “classis” yang artinya adalah proses pengelompokan, yaitu mengumpulkan suatu entitas yang memiliki ciri-ciri yang mendekati atau sama serta memisahkan entitas yang tidak memiliki kesamaan satu sama lain. (Sulistyo-Basuki, 1993). Klasifikasi yaitu membuat satu kelompokan yang dilakukan secara sistematis dari suatu objek atau benda-benda lain ke dalam suatu kelas atau golongan tertentu berdasarkan ciri-ciri yang mendekati sama (Hamakonda & Tairas, 1995) sedangkan klasifikasi adalah suatu proses memilih dan mengelompokkan suatu entitas atas suatu dasar tertentu serta meletakkannya secara berbarengan di suatu tempat yang sudah dimiliki (Bafadal, 2009),

2.2.4 Naïve Bayes

Naïve Bayes yaitu salah satu metode klasifikasi yang bersifat probalistik dengan melakukan perhitungan dari kumpulan probabilitas melalui penjumlahan antara kombinasi dan frekuensi nilai dari dataset yang tersedia. Algoritme menggunakan teorema Bayes ini memberikan asumsi bahwa seluruh fitur adalah bersifat tidak saling ketergantungan atau independen yang diberikan melalui nilai pada variabel kelas. *Naïve Bayes* juga didefinisikan sebagai pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu dengan melakukan prediksi peluang di masa yang akan datang berdasarkan yang sudah dialami pada masa sebelumnya. (Patil & Sherekar, 2013)

Berdasarkan teorema Bayes, *naïve bayes* merupakan yang mengalami perkembangan sebagai pendekatan dalam melakukan klasifikasi terhadap kelas untuk suatu dokumen. Jika pada teorema Bayes fitur-fitur yang ada terkait satu sama lain, maka pada *naïve bayes* ini memiliki asumsi bahwa setiap fitur yang ada adalah tidak memiliki kaitan satu sama lain. Walau pada kenyataannya mungkin terdapat kaitan antara fitur-fitur ini. Berikut ini merupakan gambaran dari *naïve bayes* dalam melakukan klasifikasi:

$$p(c|d) = \frac{p(f_j|c) \times p(c)}{p(d)} \quad (2.1)$$

Probabilitas $p(d|c)$ digantikan dengan perkalian probabilitas $p(f_j|c)$ dari $f|$ buah fitur independen yang merepresentasikan d . Proses pembelajaran untuk topik yang telah didapatkan dari hasil pemodelan permasalahan, adalah dengan menghitung nilai $p(f_j|c)$ yang diperoleh dari data training.

Proses klasifikasi dokumen dilakukan dengan menentukan nilai a yang akan memberikan nilai $p(a|b)$ yang paling besar dan dinyatakan sebagai berikut:

$$c^* = \arg \max_{c \in A} p(c|d) = \arg \max_{c \in A} \prod p(f_j|d) \times p(c) \quad (2.2)$$

Kelas c^* merupakan kelas yang memiliki nilai $p(c|d)$ terbesar. Nilai $p(d)$ dapat diabaikan karena nilai $p(d)$ akan bernilai sama untuk semua kelas sehingga tidak akan memberikan pengaruh apapun dalam proses perbandingan $p(c|d)$. (Mitchell, 1997)

2.2.5 Seleksi Fitur

Seleksi fitur adalah salah satu proses untuk mendapatkan fitur yang sangat memberikan pengaruh dari dalam data set. Seleksi fitur memiliki peran menentukan subset mana yang paling benar dari set fitur yang sebenarnya, dikarenakan tidak semua fitur adalah fitur yang berhubungan dengan masalah. Bahkan beberapa dari fitur atau fitur tersebut mengganggu dan dapat mengurangi akurasi. Bahkan beberapa dari fitur tersebut dapat mengganggu dan dapat mengurangi akurasi sistem. *Noisy Features* atau fitur yang tidak terpakai tersebut harus dihapus untuk meningkatkan akurasi. Selain itu, pada pengklasifikasian memiliki fitur yang banyak akan memperlambat proses komputasi. (Maimon & Rokach, 2010)

2.2.6 Information Gain

Information Gain digunakan sebagai fitur pemilih ukuran. Fitur dengan *information Gain* tertinggi dipilih sebagai fitur pemisah untuk node N . Fitur ini meminimalisasi informasi yang dibutuhkan untuk mengklasifikasi tuple dalam memberikan hasil pembelahan dan menrefleksikan nilai acak yang paling sedikit "kesalahannya" pada kelas partisi tersebut.

Perhitungan *Entropy* dapat dilakukan seperti pada rumus berikut:

$$info(D) = -\sum p_i \log_2(p_i) \quad (2.3)$$

Keterangan:

D : Himpunan kasus

M : Jumlah partisi D

p_i : Proporsi dari D_i terhadap D

Fungsi log dalam hal ini digunakan log berbasis 2 karena informasi dikodekan berbasis bit

Melakukan perhitungan nilai setelah pemisahan dapat dilakukan dengan menggunakan rumus berikut:

$$info_A(D) = -\sum_{j=1}^v \frac{|D_j|}{|D|} \times I(D_j) \quad (2.4)$$

Keterangan:

D : Himpunan kasus

A : Fitur

v : Jumlah partisi fitur A

$|D_j|$: Jumlah kasus pada partisi ke j

$|D|$: Jumlah kasus dalam D

$I(D_j)$: Total *entropy* dalam partisi

Terakhir menetapkan *information Gain* untuk fitur A dengan rumus:

$$Gain(A) = I(D) - I(A) \quad (2.5)$$

$Gain(A)$: *Information* fitur A

$I(D)$: Total *entropy*

$I(A)$: *entropy* A

2.2.7 Siswa

Siswa dilihat sebagai seseorang “subjek didik” yang mana nilai kemanusiaan sebagai individu, sebagai makhluk sosial yang mempunyai identitas moral, harus dikembangkan untuk mencapai tingkatan optimal dan kriteria kehidupan sebagai manusia warga negara yang diharapkan. (Muhaimin, 2005)

Siswa merupakan orang yang datang ke suatu lembaga untuk memperoleh atau mempelajari beberapa tipe pendidikan, selanjutnya orang ini disebut pelajar atau orang yang mempelajari ilmu pengetahuan berapapun usianya, dari manapun, siapa pun, dalam bentuk apapun, dengan biaya apapun untuk meningkatkan pengetahuan dan moral pelaku belajar. (Ali Khan, 2005)

BAB 3 METODOLOGI

Bab ini menggambarkan metodologi penelitian, terdapat tipe penelitian apa yang diteliti, strategi penelitian berisi tentang objek penelitian, pengumpulan data, implementasi, peralatan pendukung yang digunakan, gambaran umum sistem, penjelasan umum, proses algoritme, kemudian ada pengolahan data dan terakhir pengujian dan analisis.

3.1 Tipe Penelitian

Penelitian ini menggunakan metode analitik. Dimana penelitian ini memberikan penjelasan derajat relasi antar elemen dalam objek yang akan diteliti dengan fenomena atau suatu situasi yang diteliti. Kasus yang diambil adalah kemampuan siswa dalam memahami materi pelajaran Bahasa Indonesia. Metode Analitik mempunyai ciri-ciri sebagai berikut :

- a. Berfokus untuk menyelesaikan masalah pada masa kini, dan pada masalah yang benar terjadi.
- b. Data yang sudah dikumpulkan terlebih dulu disusun dan dijelaskan serta dianalisa dikarenakan metode ini bertipe analitik.

3.2 Strategi Penelitian

Metode yang digunakan dalam penelitian ini adalah Algoritme *Naïve Bayes Classification* dan seleksi fitur *Information Gain*. Perhitungan akurasi dalam penelitian ini menggunakan perhitungan akurasi, *recall* dan *precision*.

3.2.1 Peralatan yang digunakan

Peralatan yang digunakan pada penelitian ini meliputi kebutuhan perangkat lunak dan kebutuhan perangkat keras diantaranya:

Kebutuhan perangkat lunak :

- a. *Microsoft Office Word Professional Plus 2007*
Software ini digunakan untuk mengolah laporan hasil penelitian.
- b. *Microsoft Office Excel Professional Plus 2007*
Software ini digunakan sebagai media penyimpanan data set
- c. Sistem operasi *Microsoft Windows 7*
Sistem Operasi yang digunakan dalam notebook penulis.
- d. *Netbeans* dengan bahasa *java*
Software ini digunakan untuk membuat program

3.2.2 Objek Penelitian

Dari permasalahan dan urgensi berdasarkan pendahuluan, penulis menjadikan siswa sekolah menengah akhir sebagai objek penelitian yang studi kasusnya sekolah – sekolah di sekitar Malang untuk mengetahui tingkat pemahaman siswanya terhadap mata pelajaran Bahasa Indonesia.

3.2.3 Prosedur Pengumpulan Data

3.2.3.1 Data Primer

Penelitian ini menggunakan data primer, data primer yaitu data yang cara mendapatkannya secara langsung melalui sumber data. Melalui pengisian kuisisioner berupa faktor-faktor yang memberikan pengaruh terhadap pemahaman siswa dalam mempelajari pelajaran Bahasa Indonesia yaitu sesuai dengan fitur yang telah ditentukan. Kemudian untuk mendapatkan sampel kelas, penulis menggunakan nilai pada mata pelajaran Bahasa Indonesia yang diberikan secara langsung oleh guru Bahasa Indonesia dari pihak SMA BSS untuk mengukur tingkat kephahaman siswa tersebut.

3.2.3.2 Data Sekunder

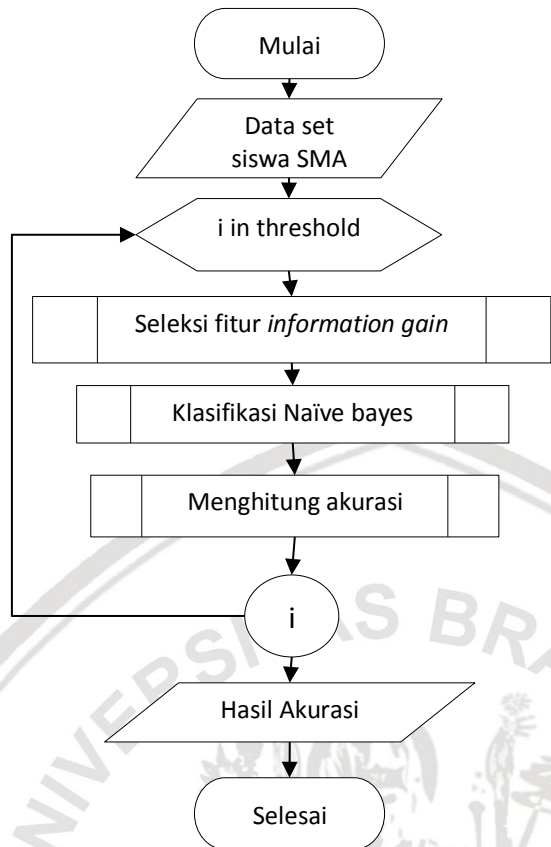
Sementara itu dalam membantu penyusunan penelitian ini diperlukan beberapa studi pustaka untuk mencari referensi untuk mendapatkan landasan penelitian dan suatu metode klasifikasi secara teoritis untuk membantu penulis dalam melakukan penelitian sehingga didapatkan konsep, dasar teori yang menjadi pendukung dalam penelitian. Tujuan studi pustaka ini adalah untuk memperoleh informasi sebanyak mungkin yang digunakan untuk menjadi landasan pada penelitian yang akan dilakukan.

3.2.4 Implementasi

Sistem diimplementasikan berdasarkan perancangan yang telah dibuat. Implementasi menggunakan bahasa pemrograman *Java*, *software* NetBeans, database *MySQL* dan beberapa *software* pendukung lainnya. Metode yang digunakan adalah klasifikasi *Naïve Bayes* dan seleksi fitur *Information Gain*.

3.2.5 Gambaran Umum Sistem

Gambaran umum sistem yang akan diimplementasikan adalah Sistem menerima masukan berupa data set siswa SMA yang akan diolah. Seleksi fitur dengan algoritme *Information Gain* lalu melakukan *training* dan *testing* dengan algoritme *Naïve Bayes* setelah itu melakukan perhitungan akurasi dengan menghitung nilai presisi, *recall* dan akurasi. Jika sudah mencapai hasil optimal maka akan memberikan keluaran hasil akurasi.



Gambar 3.1 Diagram alir Keseluruhan

3.2.6 Penjelasan Umum Algoritme

Sebelum data diklasifikasikan, fitur-fitur dari data latih yang ada, mengalami proses seleksi fitur untuk menentukan fitur mana yang memiliki pengaruh paling besar terhadap klasifikasi, dan membuang fitur yang dianggap mengganggu dan mengurangi akurasi. Metode seleksi fitur yang digunakan adalah metode *Information Gain*.

Information Gain merupakan salah satu metode seleksi fitur yang banyak dipakai oleh peneliti untuk menentukan batas dari kepentingan sebuah fitur. Nilai *information Gain* diperoleh dari nilai *entropy* sebelum pemisahan dikurangi dengan nilai *entropy* setelah pemisahan. Pengukuran nilai ini hanya digunakan sebagai tahap awal untuk penentuan fitur yang nantinya akan digunakan atau dibuang. Fitur yang memenuhi kriteria pembobotan yang nantinya akan digunakan dalam proses klasifikasi sebuah algoritme (Larose, 2005)

Metode *Naïve Bayes* pada penelitian ini diusulkan untuk klasifikasi tingkat pemahaman siswa dalam memahami materi pelajaran Bahasa Indonesia. Algoritme *Naïve Bayes* merupakan teknik prediksi berbasis *probabilistic* sederhana yang berdasar pada teorema Bayes dengan asumsi independensi (tidak ketergantungan) dengan kata lain *Naïve Bayes* merupakan model yang

menggunakan “model feature independen”. Maksudnya adalah bahwa sebuah fitur sebuah data tidak berkaitan dengan ada atau tidaknya fitur lain dalam data yang sama

3.2.7 Proses Algoritme

3.2.7.1 Algoritme Information Gain

Untuk menghitung nilai Information Gain dapat menggunakan persamaan dibawah:

1. Melakukan perhitungan *entropy* dari semua kelas

$$info(D) = -\sum p_i \log_2(p_i) \quad (3.1)$$

Keterangan:

D : Himpunan kasus

p_i : Proporsi dari D_i terhadap D

Fungsi log dalam hal ini digunakan log berbasis 2 karena informasi dikodekan berbasis bit

2. Menghitung nilai *entropy* dari semua fitur dapat dilakukan dengan menggunakan rumus berikut:

$$info_A(D) = -\sum_{j=1}^v \frac{|D_j|}{|D|} \times I(D_j) \quad (3.2)$$

Keterangan:

D : Himpunan kasus

A : Fitur

v : Jumlah partisi fitur A

$|D_j|$: Jumlah kasus pada partisi ke j

$|D|$: Jumlah kasus dalam D

$I(D_j)$: Total *entropy* dalam partisi

3. Menetapkan *information Gain* untuk fitur A dengan rumus:

$$Gain(A) = I(D) - I(A) \quad (3.3)$$

$Gain(A)$: *Information* fitur A

$I(D)$: Total *entropy*

$I(A)$: *entropy* A

4. Menentukan batas (*threshold*) yang diinginkan. Hal ini akan memungkinkan fitur yang berbobot sama dengan batas atau lebih besar akan dipertahankan serta membuang fitur yang berada dibawah batas.

5. Mengeluarkan hasil data set yang sudah mengalami pengulangan fitur

3.2.7.2 Algoritme Klasifikasi Naive Bayes

Untuk menghitung klasifikasi naive bayes dapat menggunakan langkah-langkah berikut:

1. Menghitung peluang awal *prior* $P(H)$. Peluang bahwa data sampel diamati tanpa memperhatikan nilai yang lain
2. Menghitung *likelihood* peluang diamatinya data sampel X dengan mempertibangkan nilai H. Agar tidak mendapatkan nilai 0 maka disini dilakukan proses *smoothing (laplace)* dengan rumus sebagai berikut:

$$P(X_i|H) = \frac{\text{count}(X_i|H)+1}{\sum \text{count}(X|H)+|V|} \tag{3.4}$$

3. Lalu terakhir akan menghasilkan keluaran berupa hasil pelatihan
4. Ketika sudah memiliki data hasil *training*, *likelihood*, *prior* dan juga data testing langkah selanjutnya adalah menghitung *posterior* dengan rumus:

$$P(H|X) = P(X_i|H)P(H) \tag{3.5}$$

dengan kata lain : *posteriori* = *likelihood* * *prior*

5. Menghasilkan keluaran hasil klasifikasi

3.2.8 Pengolahan Data

- a. Tahap Pertama, penentuan data yang akan diolah yaitu data siswa SMA.
- b. Tahap Ketiga, menentukan kelas dan fitur yang akan digunakan dari tahap pertama. Kelas tersebut terdiri dari siswa sangat paham, cukup paham dan kurang paham. Fitur yang digunakan terdapat pada Tabel 3.1. Fitur berisi fitur apa saja yang akan digunakan untuk klasifikasi, terdiri dari 17 fitur. Kategori berisi kategori dari fitur tersebut, dalam satu fitur jumlah kategori yang ada adalah berbeda-beda. Contoh dalam fitur jumlah anggota keluarga terdapat 2 kategori yaitu >3 dan ≤ 3 .

Tabel 3.1 Fitur dan Kategori yang digunakan

Fitur	Kategori
Jumlah Anggota Keluarga	>3 , ≤ 3
Status tempat tinggal	Tinggal bersama dengan orang tua , Tinggal berpisah dengan orang tua
Pendidikan terakhir Ibu	Tidak sekolah, SD, SMP, SMA, $\geq S1$
Pendidikan terakhir Ayah	Tidak sekolah, SD, SMP, SMA, $\geq S1$



Pekerjaan Ibu	Tidak bekerja, PNS, Pegawai swasta, Pengajar, Wirausaha, Lainnya
Pekerjaan Ayah	Tidak bekerja, PNS, Pegawai swasta, Pengajar, Wirausaha, Lainnya
Pengasuh	Orang tua, Lainnya
Jarak Rumah	Sangat Dekat, Dekat, Sedang, Jauh
Dukungan Keluarga	Ya, Tidak
Ikut Ekstrakurikuler	Ya, Tidak
Mengulang Pelajaran Di rumah	Ya, Tidak
Posisi Duduk Di kelas	1,2,3,4,5
Internet di Rumah	Ada, Tidak
Mengikuti Bimbingan Belajar	Ya, Tidak
Lama Belajar di Rumah	< 1 jam, 1-2 jam , > 2 jam
Jenis bacaan di rumah	Buku pelajaran, Novel, Majalah, Koran, Lainnya
Lama membaca di rumah	< 1 jam, 1-2 jam , > 2 jam

- c. Tahap keempat, menyalin data yang telah didapatkan dari kuesioner yang telah disebarakan kepada siswa SMA BSS ke *file* csv. *File* csv ini berguna untuk data latih dan data uji yang akan diolah oleh program.
- d. Tahap Keempat, memberikan label kelas yang didapat dari *file* nilai dari guru Bahasa Indonesia SMA BSS. Guru Bahasa Indonesia memberikan *file* berisi nama dan nilai siswa selama 3 bulan, penulis menggunakan nilai rata-rata selama 3 bulan untuk dijadikan label kelas. Jika nilai lebih dari 84 maka termasuk kedalam kelas paham, nilai lebih 74 termasuk kedalam kelas kelas cukup paham dan selain itu termasuk kedalam kelas kurang paham.

3.2.9 Pengujian dan analisis

Evaluasi (Evaluation) dalam tahapan ini akan dilakukan validasi serta pengukuran keakuratan hasil yang dicapai oleh model menggunakan perhitungan *Confusion Matrix*. Sebelum diuji dengan confusion matrix, untuk menguji baliditas data maka dilakukan pengujian k-fold cross validation terlebih dahulu dicari akurasi yang memiliki nilai maksimal.

Tabel 3.2 Pengujian K-fold cross validation

K	Nilai Gain	Akurasi	Fitur yang terpilih
1			
2			
3			
4			
5			

Tabel 3.3 Confusion Matrix

		True	False
Nilai Prediksi	True	TP	FP
	False	FN	TN

Untuk melakukan perhitungan Akurasi, Precision dan recall, perlu dilakukan pencarian nilai TF (*True Positif*), TN (*True Negatif*) FP (*False Positif*) dan (*False Negatif*). TF adalah jumlah data yang bersifat benar yang ada dikelas itu sendiri. TN adalah jumlah data yang bersifat benar pada kelas lain selain kelasnya sendiri. FP adalah jumlah data yang salah pada kelasnya sendiri. Sedangkan FN adalah jumlah data salah pada kelas lain selain kelasnya sendiri. Persamaan untuk menghitung nilai Precision adalah sebagai berikut:

$$Precision = \frac{\sum_{i=1}^l TP_i}{\sum_{i=1}^l TP_i + FP_i} \quad (3.6)$$

Nilai Precision adalah untuk pengukuran kemiripan atau kecocokan antara informasi yang ingin didapatkan dengan hasil yang didapatkan. Nilai Precision menggambarkan jumlah data dengan kategori positif yang terklasifikasi dengan benar dibagi dengan seluruh jumlah data yang diklasifikasi positif. Dapat diperoleh dengan persamaan 3.1.

$$Recall = \frac{\sum_{i=1}^l TP_i}{\sum_{i=1}^l TP_i + FN_i} \quad (3.7)$$

Nilai Recall adalah seberapa banyak data dengan kategori positif yang terklasifikasikan secara benar oleh sistem. Nilai recall menggambarkan jumlah data dengan kategori positif yang terklasifikasikan dibagi dengan seluruh dokumen yang relevan. Recall dapat dihitung dengan persamaan 3.2

$$Accuracy = \sum_{i=1}^l \frac{TP+TN}{TP+TN+FP+FN} \quad (3.8)$$

Nilai akurasi menunjukkan bahwa seberapa akurat sistem dapat memberikan hasil klasifikasi dari data dengan benar. Artinya nilai akurasi ini adalah nilai yang membandingkan antara nilai hasil klasifikasi yang benar dengan seluruh hasil klasifikasi. Persamaan untuk mendapatkan nilai akurasi adalah seperti pada persamaan 3.3

Selain itu, akan dilakukan pengujian fitur menggunakan nilai threshold dari nilai *Gain* masing-masing fitur dengan merubah nilai threshold mulai dari 0 hingga 1.5 kemudian akan dilakukan perhitungan satu persatu nilai dari *Confusion Matrix*.



BAB 4 PERANCANGAN

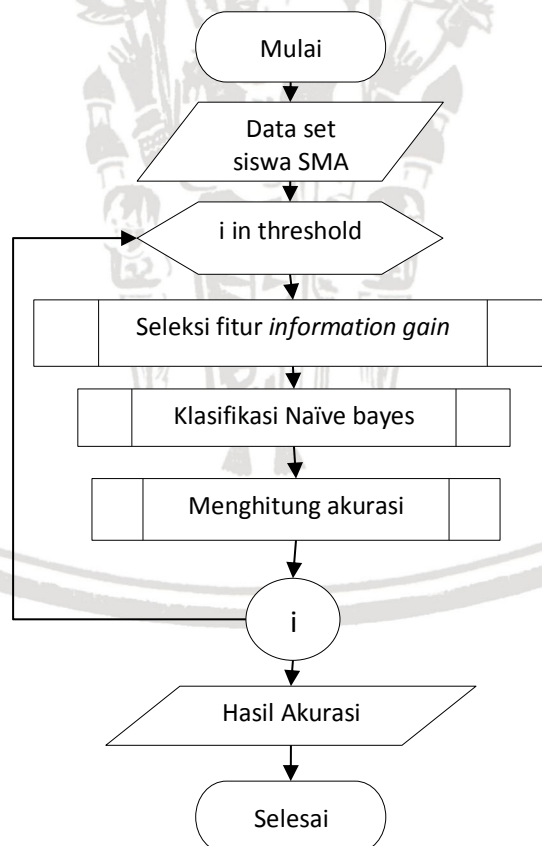
Bab ini membahas perancangan sistem yang akan dibuat meliputi formulasi permasalahan, siklus algoritme *Naïve Bayes*, seleksi fitur *Information Gain*, manualisasi dari algoritme yang digunakan dan penrancangan pengujian sistem yang dibuat.

4.1 Formulasi Permasalahan

Berdasarkan metodologi yang telah dituliskan di bab 3, permasalahan yang diselesaikan dengan Klasifikasi *Naïve Bayes* untuk memprediksi tingkat pemahaman siswa dalam materi pelajaran Bahasa Indonesia ditentukan oleh nilai masing-masing fitur yang ada. Kemudian akan dilakukan pengujian dengan *Confusion Matrix* serta menguji fitur mana saja yang memiliki tingkat relevansi yang tinggi terhadap hasil klasifikasi dengan metode seleksi fitur *Information Gain*.

4.2 Diagram alir

Berikut adalah Diagram alir dari metode yang akan digunakan dimana menggambarkan tahapan tahapan yang dilakukan dalam algoritme.

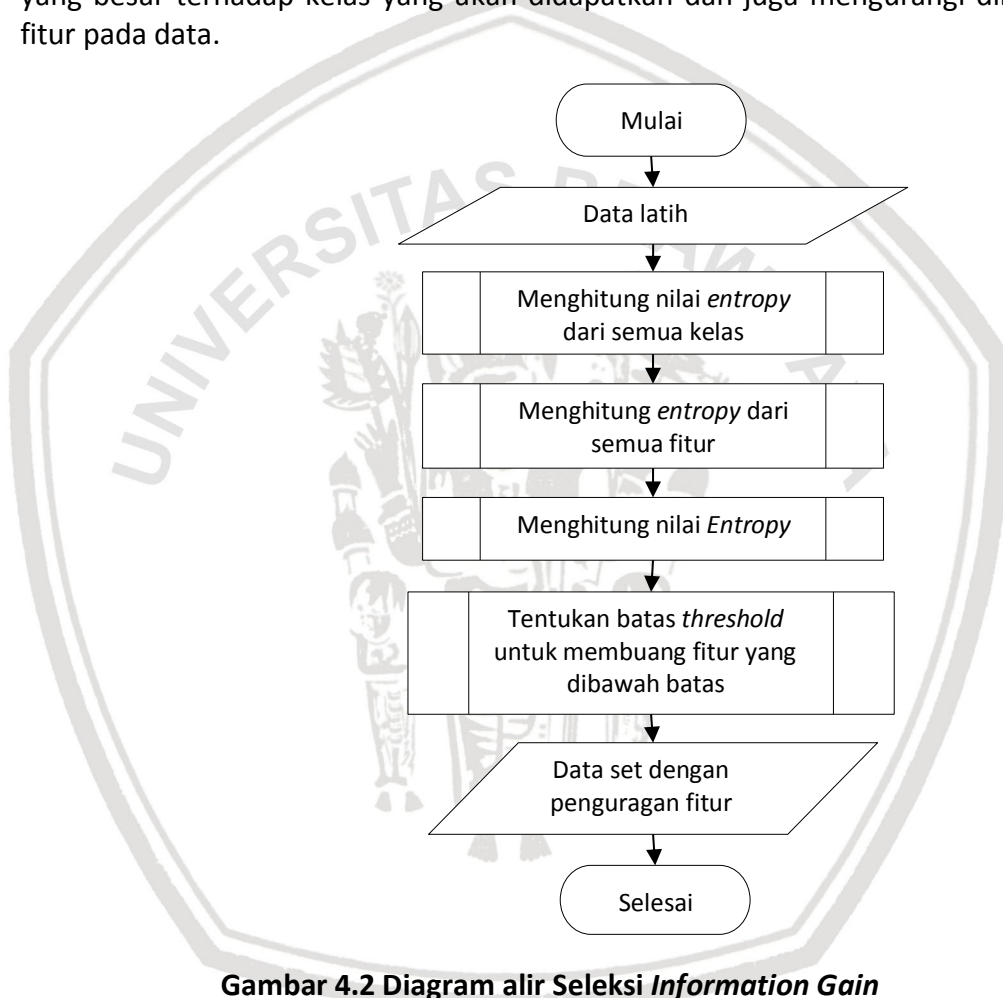


Gambar 4.1 Diagram alir keseluruhan

Berdasarkan **Gambar 4.1** secara umum sistem yang akan diimplementasikan adalah Sistem menerima masukan berupa data set siswa SMA yang akan diolah. Perulangan dilakukan saat i pada treshold kemudian dilakukan seleksi fitur dengan algoritme *Information Gain* lalu melakukan *training* dan *testing* dengan algoritme *Naïve Bayes* setelah itu melakukan perhitungan akurasi dengan menghitung nilai presisi, *recall* dan akurasi. Jika sudah memenuhi i maka akan memberikan keluaran hasil akurasi.

4.2.1 Seleksi Fitur *Information Gain*

Seleksi fitur dilakukan untuk mendapatkan fitur yang memiliki pengaruh yang besar terhadap kelas yang akan didapatkan dan juga mengurangi dimensi fitur pada data.



Gambar 4.2 Diagram alir Seleksi *Information Gain*

Berdasarkan Gambar 4.2 tahapan pada algoritme seleksi fitur *Information Gain* diantaranya:

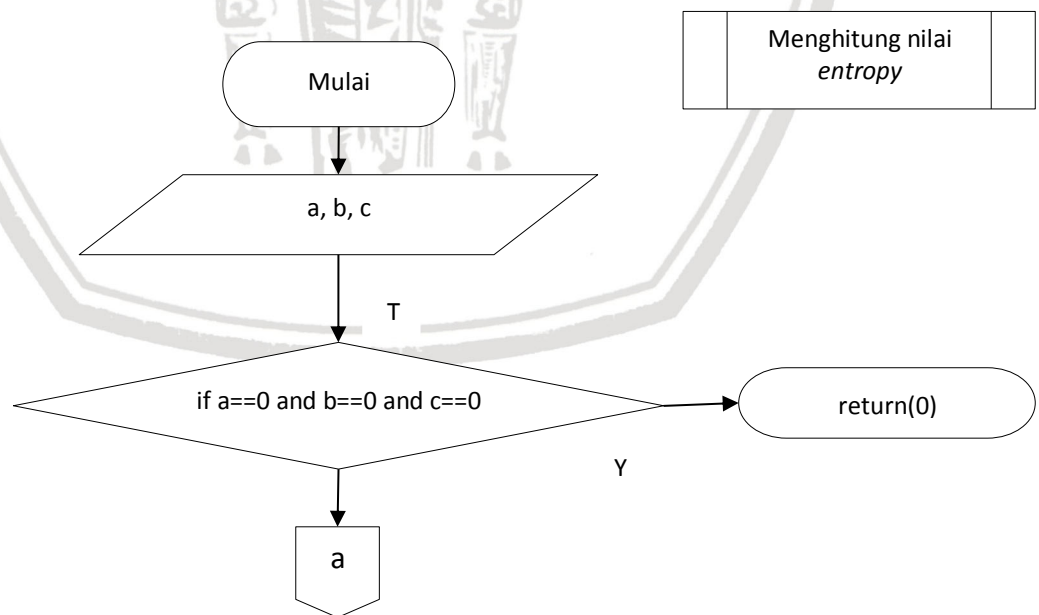
1. Sistem menerima masukan berupa data latih yang akan diolah
2. Melakukan perhitungan *entropy* dari semua kelas yaitu kelas paham, kelas cukup paham dan kelas kurang paham
3. Menghitung nilai *entropy* dari semua fitur dapat dilakukan

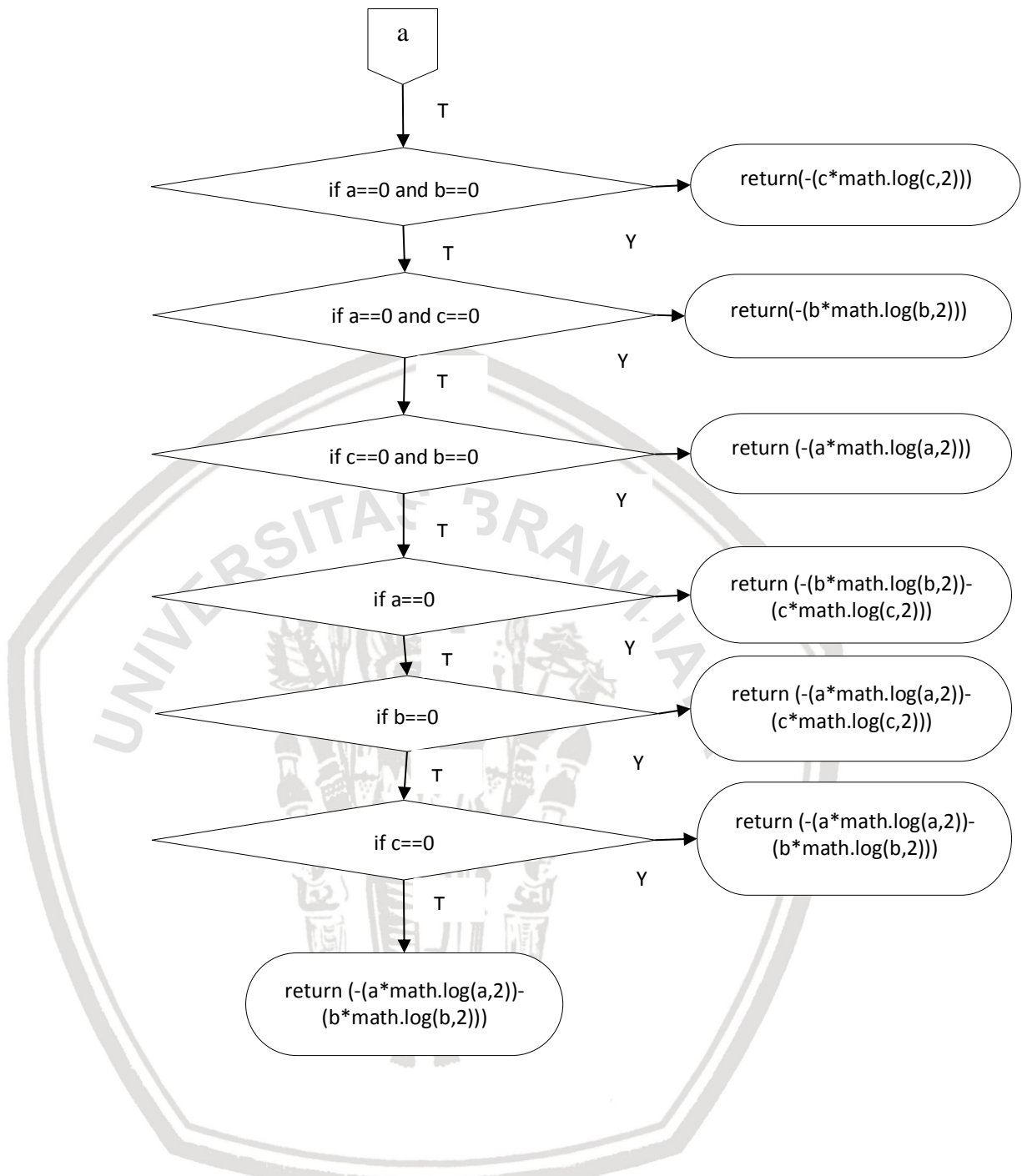
4. Menetapkan *information Gain* untuk fitur A dengan rumus:
5. Menentukan batas (*threshold*) yang diinginkan. Hal ini akan memungkinkan fitur yang berbobot sama dengan batas atau lebih besar akan dipertahankan serta membuang fitur yang berada dibawah batas.
6. Mengeluarkan hasil data set yang sudah mengalami pengulangan fitur

4.2.1.1 Menghitung entropy

Berdasarkan Gambar 4.3 Tahapan untuk menghitung *entropy* diantaranya:

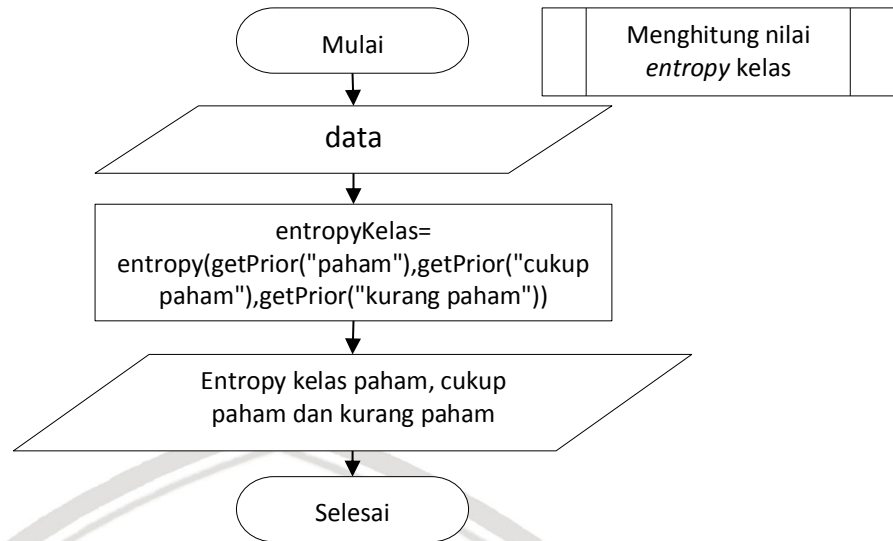
1. Masukan berupa hasil perhitungan *prior* dengan kelas paham, cukup paham dan kurang paham.
2. Dilakukan seleksi kondisi apabila $a=0$, $b=0$, dan $c=0$ jika ya maka akan mereturn hasil *entropy* menjadi 0
3. Jika tidak maka dilakukan seleksi kondisi apabila $a=0$ dan $b=0$ jika ya maka akan memproses $-(c \cdot \log_2(c))$
4. Jika tidak maka dilakukan seleksi kondisi apabila $a=0$ dan $c=0$ jika ya maka akan memproses $-(b \cdot \log_2(b))$
5. Jika tidak maka dilakukan seleksi kondisi apabila $c=0$ dan $b=0$ jika ya maka akan memproses $-(a \cdot \log_2(a))$
6. Jika tidak maka dilakukan seleksi kondisi apabila $a=0$ jika ya maka akan memproses $-(b \cdot \log_2(b)) - (c \cdot \log_2(c))$
7. Jika tidak maka dilakukan seleksi kondisi apabila $b=0$ jika ya maka akan memproses $-(a \cdot \log_2(a)) - (c \cdot \log_2(c))$
8. Jika tidak maka dilakukan seleksi kondisi apabila $c=0$ jika ya maka akan memproses $-(a \cdot \log_2(a)) - (b \cdot \log_2(b))$
9. Jika tidak maka akan memproses $-(a \cdot \log_2(a)) - (b \cdot \log_2(b)) - (c \cdot \log_2(c))$
10. Proses menghasilkan *entropy* dari kelas paham, cukup paham, kurang paham





Gambar 4.3 Diagram alir Menghitung Entropy

a. Menghitung *entropy* dari kelas tertentu



Gambar 4.4 Diagram alir Menghitung Entropy semua kelas

Berdasarkan Gambar 4.4 tahapan untuk menghitung entropy dari setiap kelas diantaranya:

1. Masukkan berupa method `getPrior` dengan parameter paham, cukup paham dan kurang paham dan method `entropy` dengan parameter `getPrior` beserta kelasnya masing-masing
2. Memanggil method `entropy` dengan parameter `getPrior` beserta kelasnya masing-masing kemudian di simpan dalam variable `entropyKelas`
3. Menghasilkan nilai `entropy` setiap kelas yang ada

4.2.1.2 Menghitung Entropy dari semua fitur

Berdasarkan Gambar 4.5 tahapan untuk menghitung `entropy` dari semua fitur diantaranya:

1. Program menerima masukan berupa data latih yang akan diolah
2. Menentukan jumlah kategori tertentu dari setiap fitur
3. Melakukan proses mencari probabilitas kategori dari setiap fitur dengan rumus:

$$p(c) = \frac{\text{jumlah kategori tertentu}}{\text{jumlah data latih}} \tag{4.1}$$

4. Menghitung `likelihood` tanpa proses `smoothing` dengan rumus:

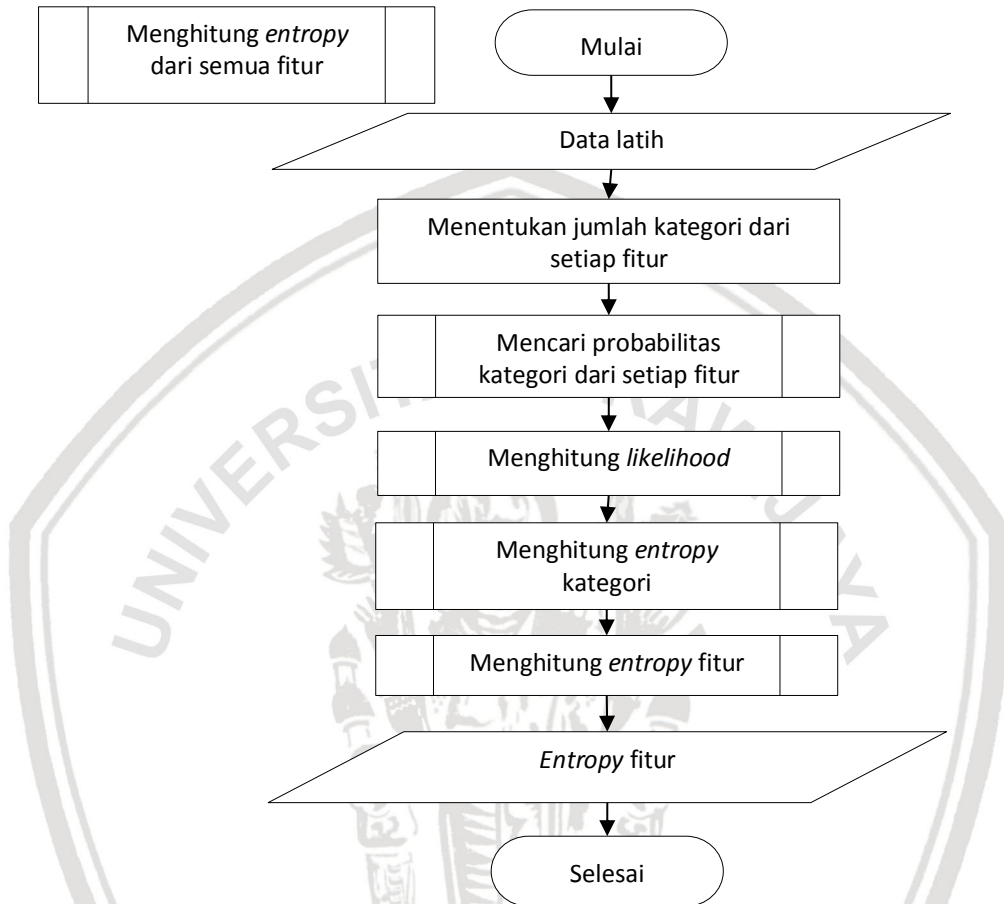
$$P(X_i|H) = \frac{\text{jumlah kategori dikelas tertentu}}{\text{jumlah kelas tertentu}} \tag{4.2}$$

5. Menghitung `entropy` kategori dengan rumus:



$$E(c) = \text{likelihood} \log_2 \text{likelihood} \quad (4.3)$$

6. Melakukan proses menghitung nilai entropi fitur dengan perkalian kemudian penjumlahan antara probabilitas kategori dan *entropy* kategori
7. Menghasilkan nilai *entropy* fitur



Gambar 4.5 Diagram alir Pengukuran *Entropy* semua fitur

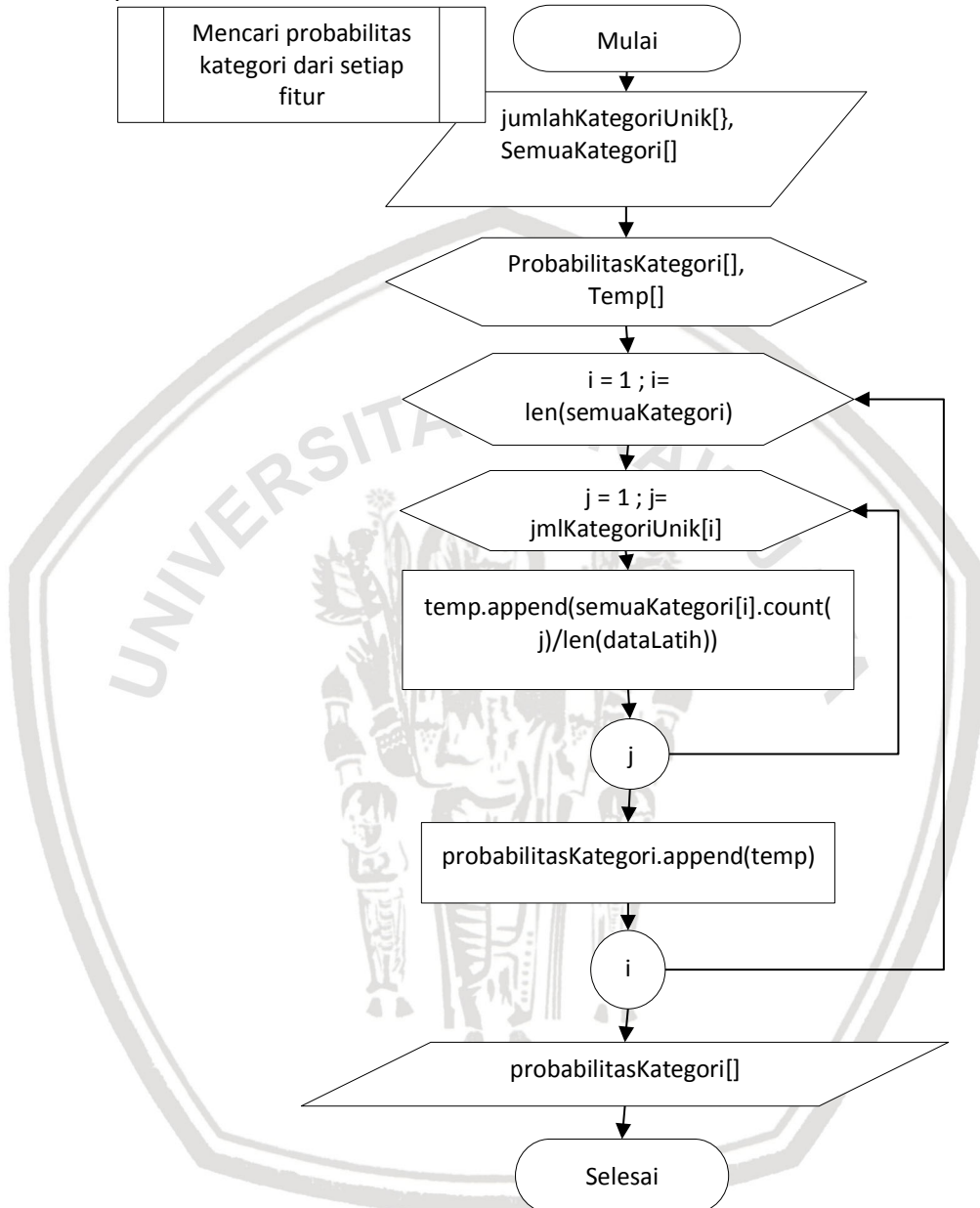
a. Mencari probabilitas kategori dari setiap fitur

Berdasarkan Gambar 4.6 tahapan untuk mencari probabilitas kategori dari setiap fitur diantaranya:

1. Program menerima masukan berupa jumlah kategori unik dari setiap fitur
2. Inisialisasi array probabilitasKategori, array ar dan semuaKategori
3. Perulangan $i = 1$ sampai $i = \text{panjang}$ dari array semuaKategori
4. Perulangan $j=1$ sampai $j = \text{array jmlKategoriUnik}$ indeks ke i
5. Menambahkan elemen dari hasil count j dari array semuaKategori indeks ke i dibagi dengan jumlah seluruh data latih
6. Jika sudah memenuhi j maka perulangan selesai



7. Kemudian melakukan proses penambahan elemen dari array `ark e` dalam array `probabilitasKategori`
8. Jika sudah memenuhi `j` perulangan maka selesai
9. Program pengashilkan keluar berupa probabilitas dari setiap kategori pada seluruh fitur



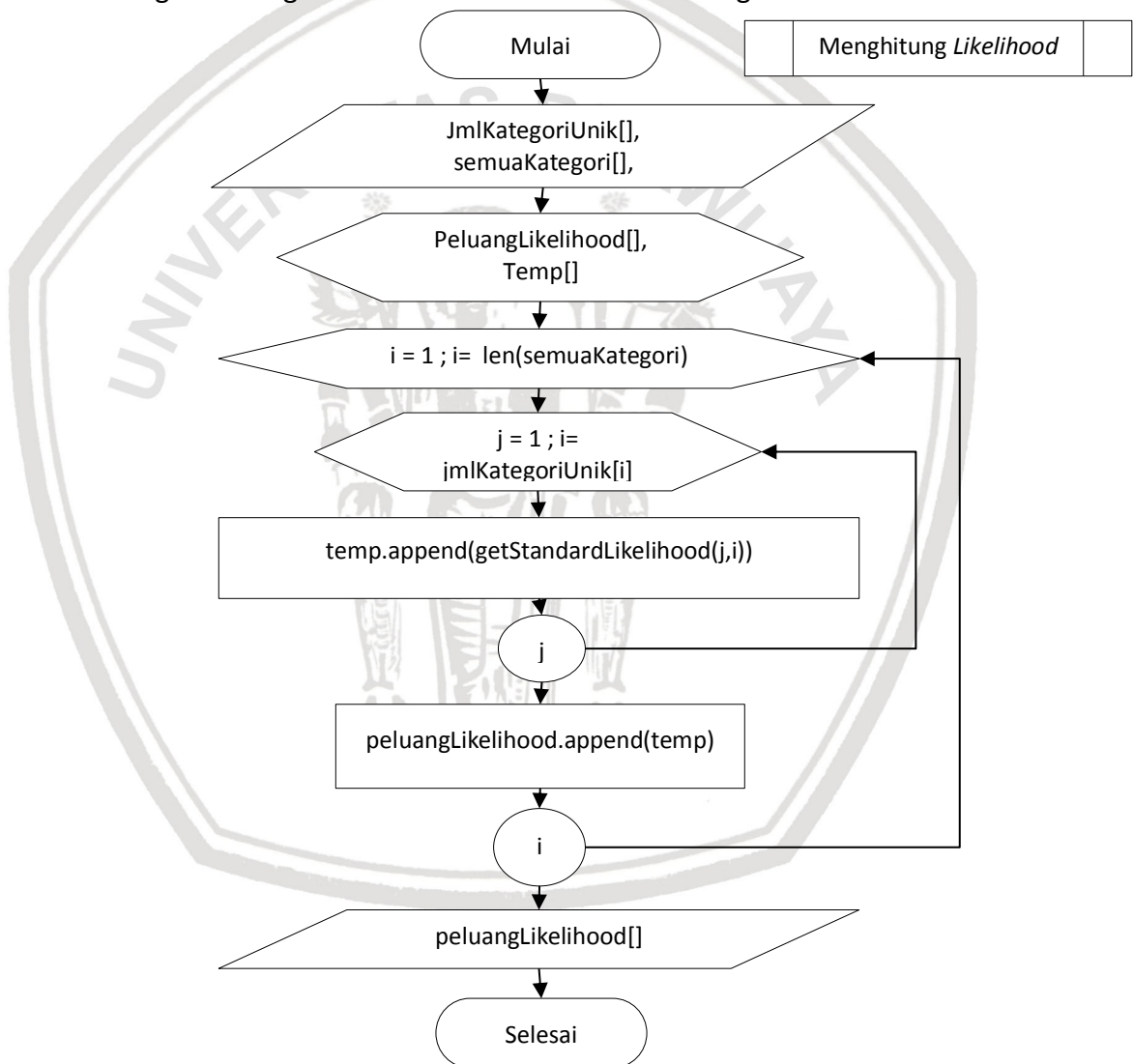
Gambar 4.6 Diagram alir Mencari probabilitas fitur

b. Menghitung *likelihood*

Berdasarkan Gambar 4.7 tahapan dalam mencari *likelihood* dari semua fitur diantaranya:

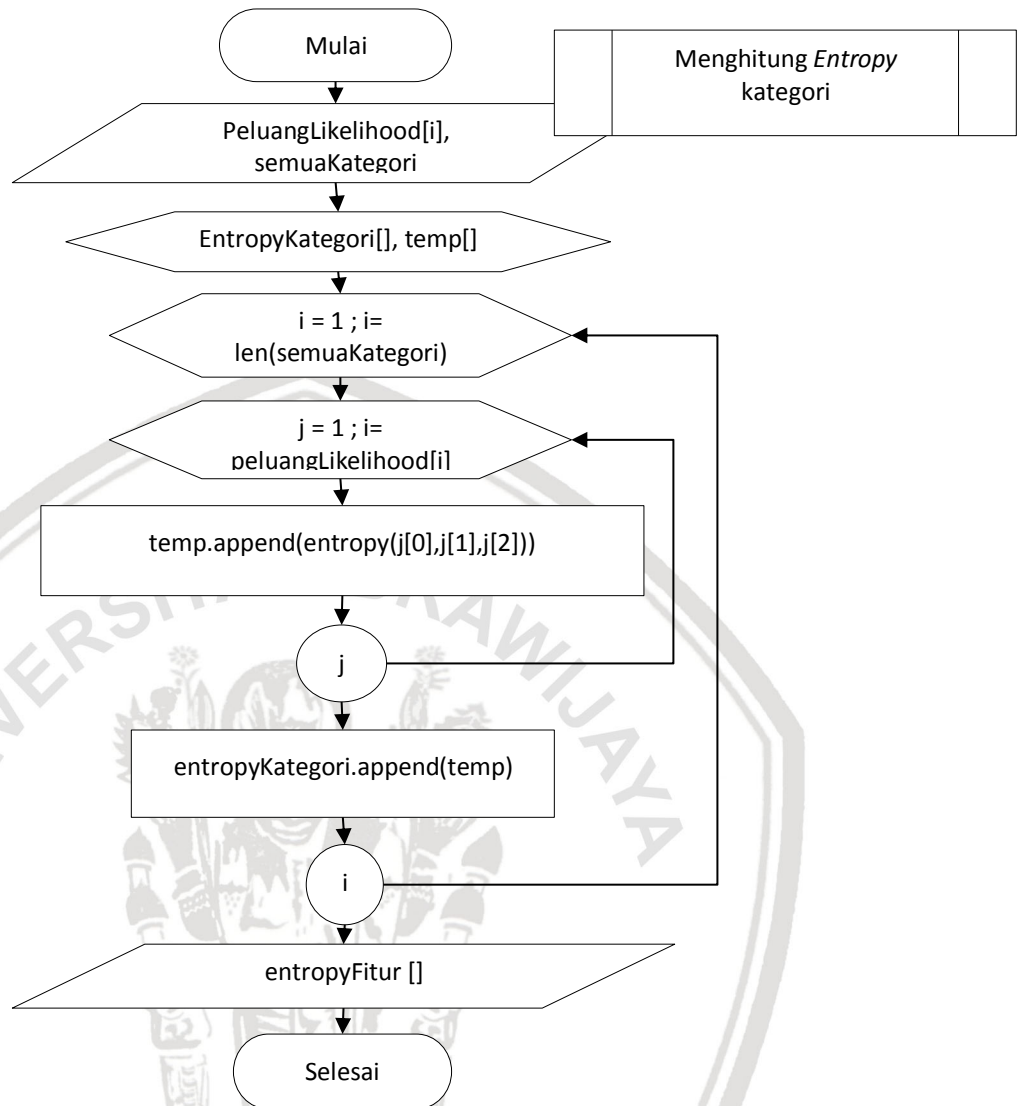


1. Masukkan berupa jumlah kategori fitur, array semuaKategori yang berisi semua kategori pada fitur dan *method* *getStandardLikelihood* yang berfungsi mencari *likelihood*
2. Inialisasi array peluangLikelihood dan array temp
3. Perulangan i = 1 sampai i = panjang dari array semuaKategori
4. Perulangan j=1 sampai j = array JmlKategoriUnik indeks ke i
5. Mengisi dan menanggil parameter dari *method* *getStandardLikelihood* dengan i dan j kemudian elemen dari *method* dimasukkan ke array temp
6. Jika sudah memenuhi j maka perulangan selesai
7. Menambahkan elemen dari temp kedalam array peluangLikelihood
8. Jika sudah memenuhi I maka perulangan selesai
9. Program menghasilkan *likelihood* dari semua kategori



Gambar 4.7 Diagram alir menghitung *likelihood*

c. Menghitung *entropy* kategori



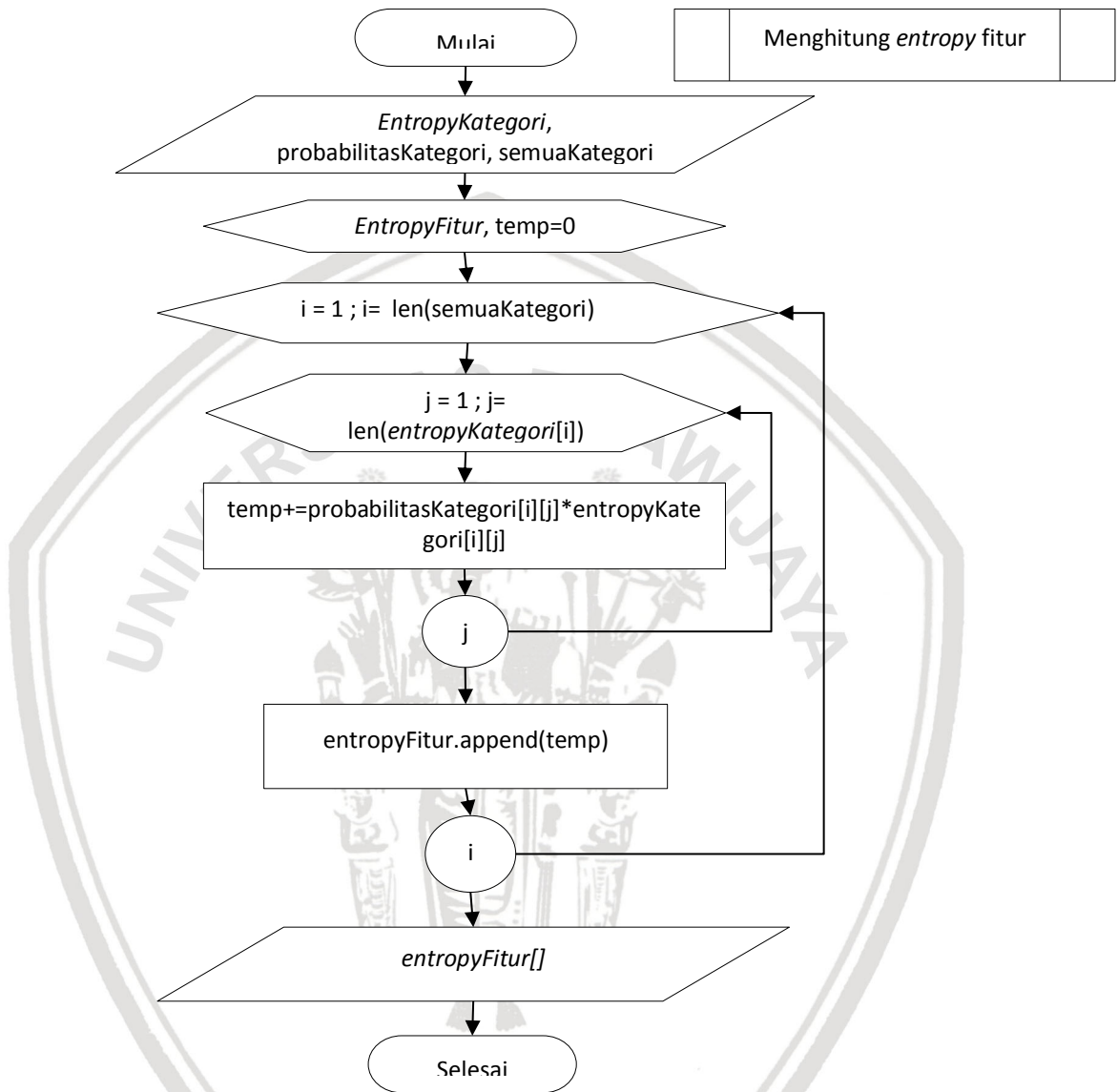
Gambar 4.8 Diagram alir menghitung entropy kategori

Berdasarkan Gambar 4.8 tahapan untuk menghitung nilai *entropy* kategori diantaranya:

1. Masukkan berupa jumlah peluangLikelihood dan array semuaKategori yang berisi semua kategori pada fitur
2. Inisialisasi array etropyKategori dan array temp
3. Perulangan i = 1 sampai i = panjang dari array semuaKategori
4. Perulangan j=1 sampai j = array peluangLikelihood indeks ke i
5. Memanggil dan mengisi parameter dari *method entropy* dengan j pada indeks ke 0, 1 dan 2 kemudian elemen dari *method* dimasukkan ke array temp
6. Jika sudah memenuhi j maka perulangan selesai

7. Menambahkan elemen dari temp kedalam array *entropyKategori*
8. Jika sudah memenuhi I maka perulangan selesai
9. Program menghasilkan nilai *entropy* dari semua fitur

d. Menghitung *entropy* fitur



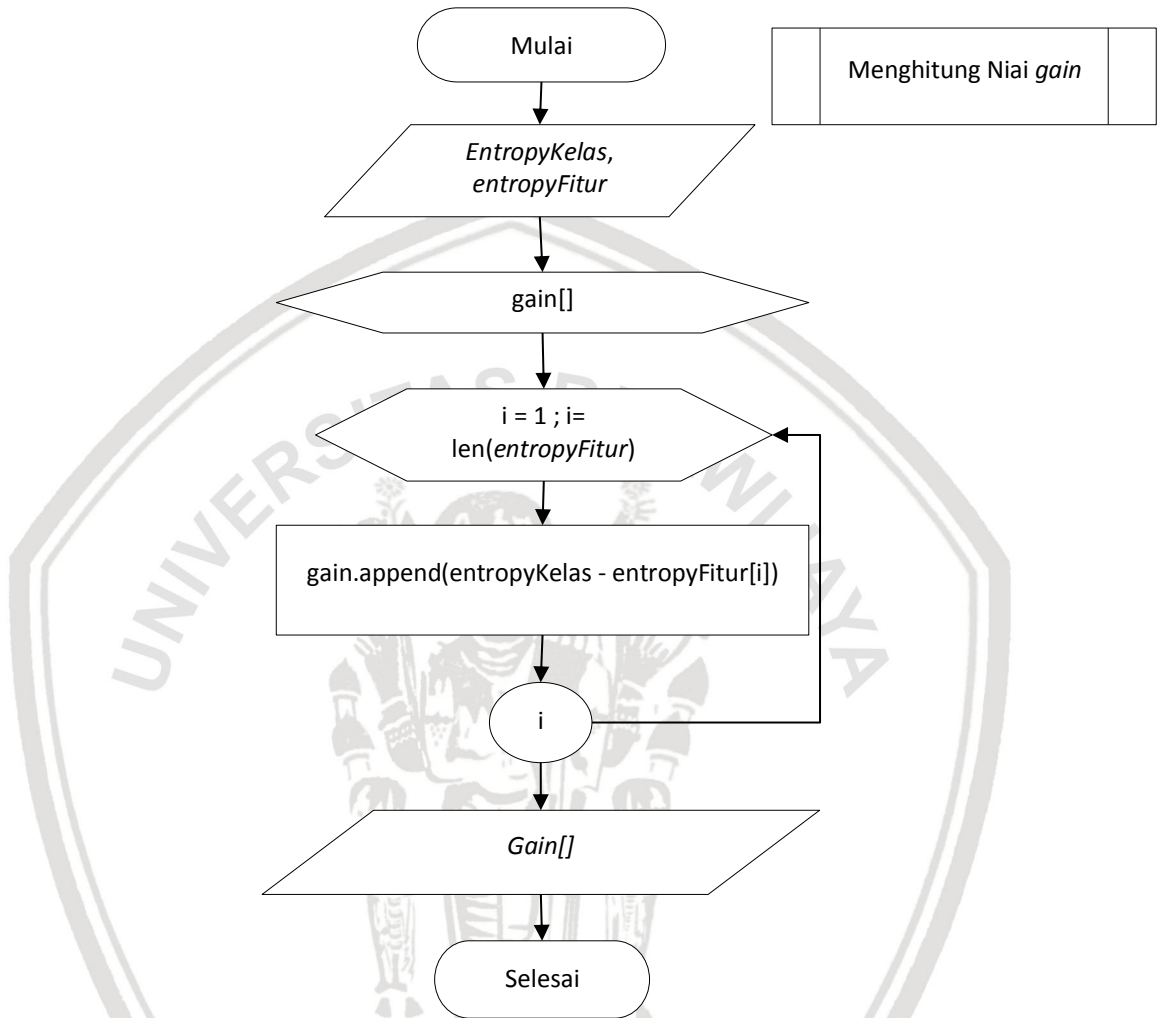
Gambar 4.9 Diagram alir Menghitung nilai *entropy* fitur

Berdasarkan Gambar 4.9 tahapan menghitung *entropy* fitur diantaranya:

1. Masukan berupa nilai dari hasil perhitungan *entropy* kategori, probabilitas kategori dan array *semuaKategori* yang berisi semua kategori pada fitur
2. Inialisasi *entropyFitur* dan inialisasi variable *temp = 0*
3. Perulangan *i = 1* sampai *i = panjang* dari array *semuaKategori*
4. Perulangan *j=1* sampai *j = panjang* dari *entropyKategori* indeks ke *i*
5. Melakukan proses sumproduct antara peluang kaategori indeks *I* dan *j* dengan *entropy* kategori index *I* dan *j*

6. Jika sudah memenuhi j maka perulangan selesai
7. Menambahkan elemen dari temp kedalam array *entropyFitur*
8. Jika sudah memenuhi i maka perulangan selesai
9. Program menghasilkan nilai *entropy* dari setiap fitur

4.2.1.3 Menghitung nilai *Gain*

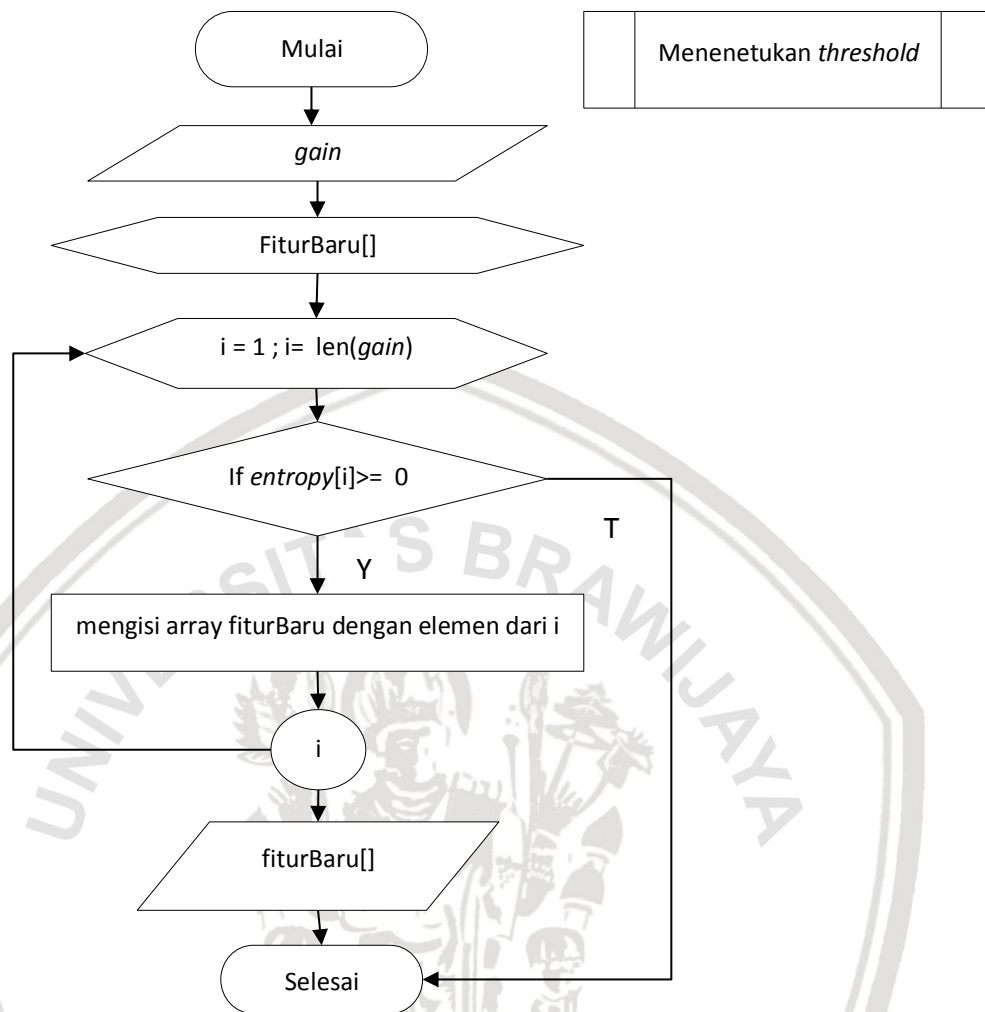


Gambar 4.10 Diagram alir Menghitung nilai *Gain*

Berdasarkan Gambar 4.10 tahapan untuk menghitung nilai *Gain* diantaranya:

1. Masukkan berupa hasil dari perhitungan *entropy* kelas dan *entropy* fitur
2. Inialisasi array *entropy* untuk menyimpan nilai *Gain*
3. Perulangan i = 1 sampai i = panjang *entropyFitur*
4. Melakukan pengurangan antara *entropyKelas* dikurang dengan *entropyFitur* index ke i kemudian elemennya disimpan pada array *Gain*
5. Jika sudah memenuhi i maka perulangan selesai
6. Program menghasilkan nilai *Gain* dari semua fitur

4.2.1.4 Menentukan *threshold*



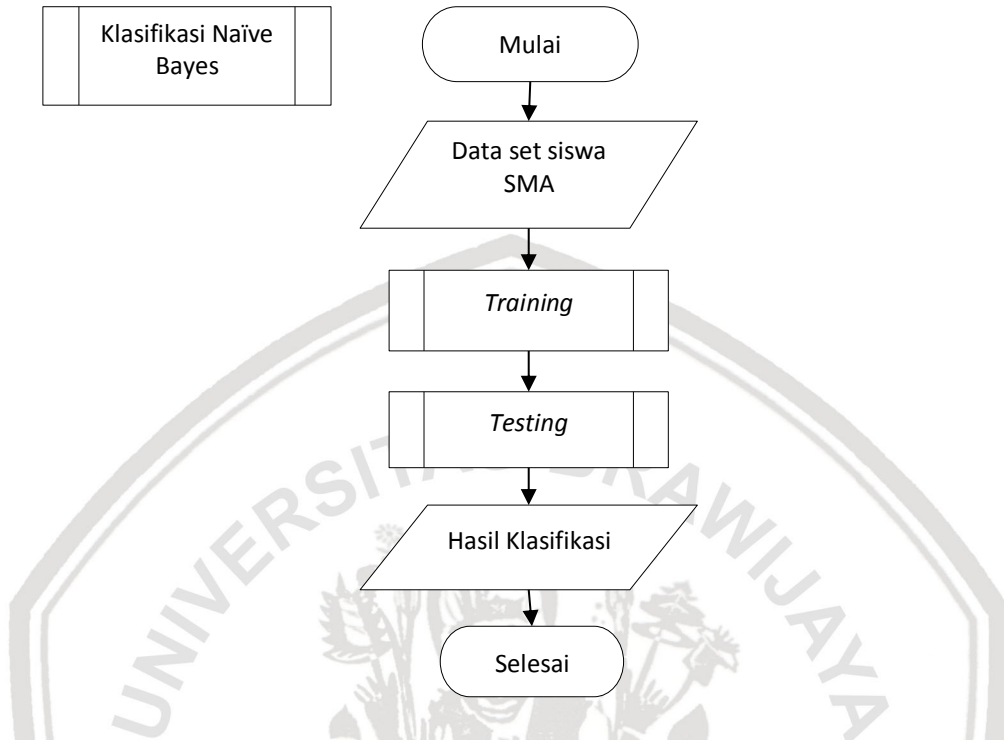
Gambar 4.11 Diagram alir menentukan *threshold*

Berdasarkan Gambar 4.11 tahapan menentukan *threshold* diantaranya:

1. Masukkan berupa nilai *Gain*
2. Inisialisasi array fiturBaru
3. Perulangan $i = 1$ sampai $i = \text{panjang Gain}$
4. Seleksi kondisi jika nilai *Gain* index ke i lebih dari atau sama dengan 0 maka
5. Menambahkan elemen i kedalam array fitur baru
6. Jika tidak maka akan terjadi penghapusan fitur
7. Jika sudah memenuhi i maka perulangan selesai
8. Program akan menghasilkan fitur baru yang sudah di seleksi

4.2.2 Metode Naïve Bayes

Metode *Naïve Bayes* adalah metode yang digunakan untuk prediksi kelas mana yang didapatkan



Gambar 4.12 Diagram alir Metode *Naïve Bayes*

Berdasarkan Gambar 4.12 tahapan pada metode *Naïve Bayes* adalah sebagai berikut:

1. Terdapat data secara keseluruhan yang terdiri dari data latih dan data uji
2. Melakukan *training* sejumlah data latih yang ada
3. Melakukan *testing* sejumlah data uji yang ada
4. Menghasilkan keluaran berupa kelas hasil klasifikasi

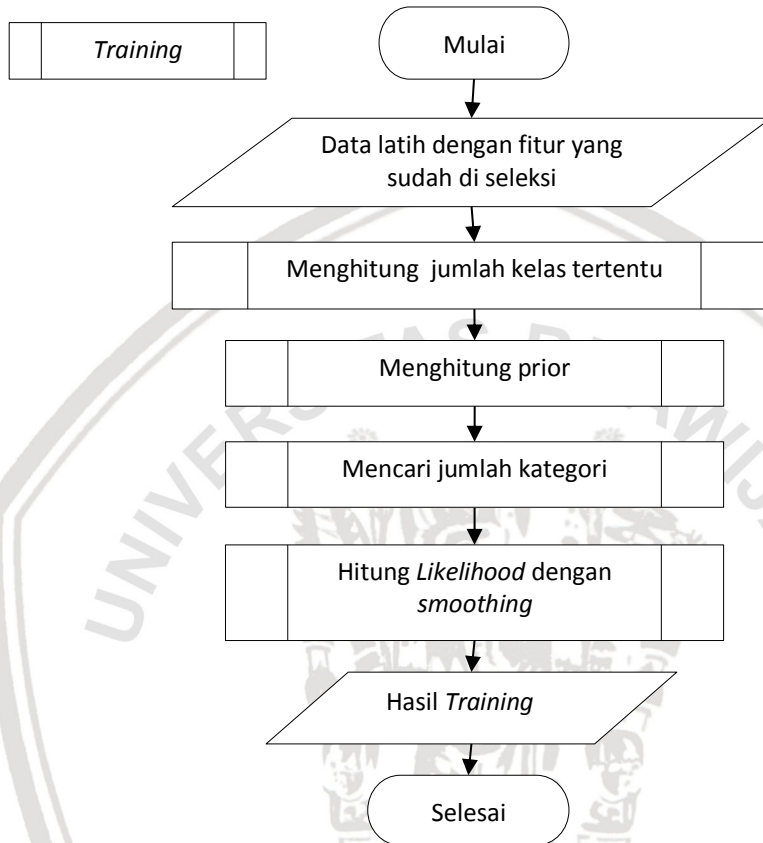
4.2.2.1 Training

Training digunakan untuk melatih data, selama proses pelatihan harus dilakukan pembelajaran probabilitas akhir $P(H|X)$ pada model untuk setiap kombinasi H dan Y berdasarkan informasi yang didapat dari data latih.

Berdasarkan Gambar 4.13 tahapan pada *training* adalah sebagai berikut:

1. Memasukkan data latih dengan fitur yang sudah diseleksi dan mengalami pengurangan
2. Menghitung jumlah kelas tertentu
3. Menghitung peluang awal *prior* $P(H)$. Peluang bahwa data sampel diamati tanpa memperhatikan nilai yang lain
4. Menghitung jumlah kategori dari setiap fitur

5. Menghitung *likelihood* peluang diamatinya data sampel X dengan mempertimbangkan nilai H. Agar tidak mendapatkan nilai 0 maka disini dilakukan proses *smoothing (laplace)* dengan rumus sebagai berikut: $P(X_i|H) = \frac{\text{count}(X_i|H)+1}{\sum \text{count}(X|H)+|V|}$
6. Lalu terakhir akan menghasilkan keluaran berupa hasil pelatihan

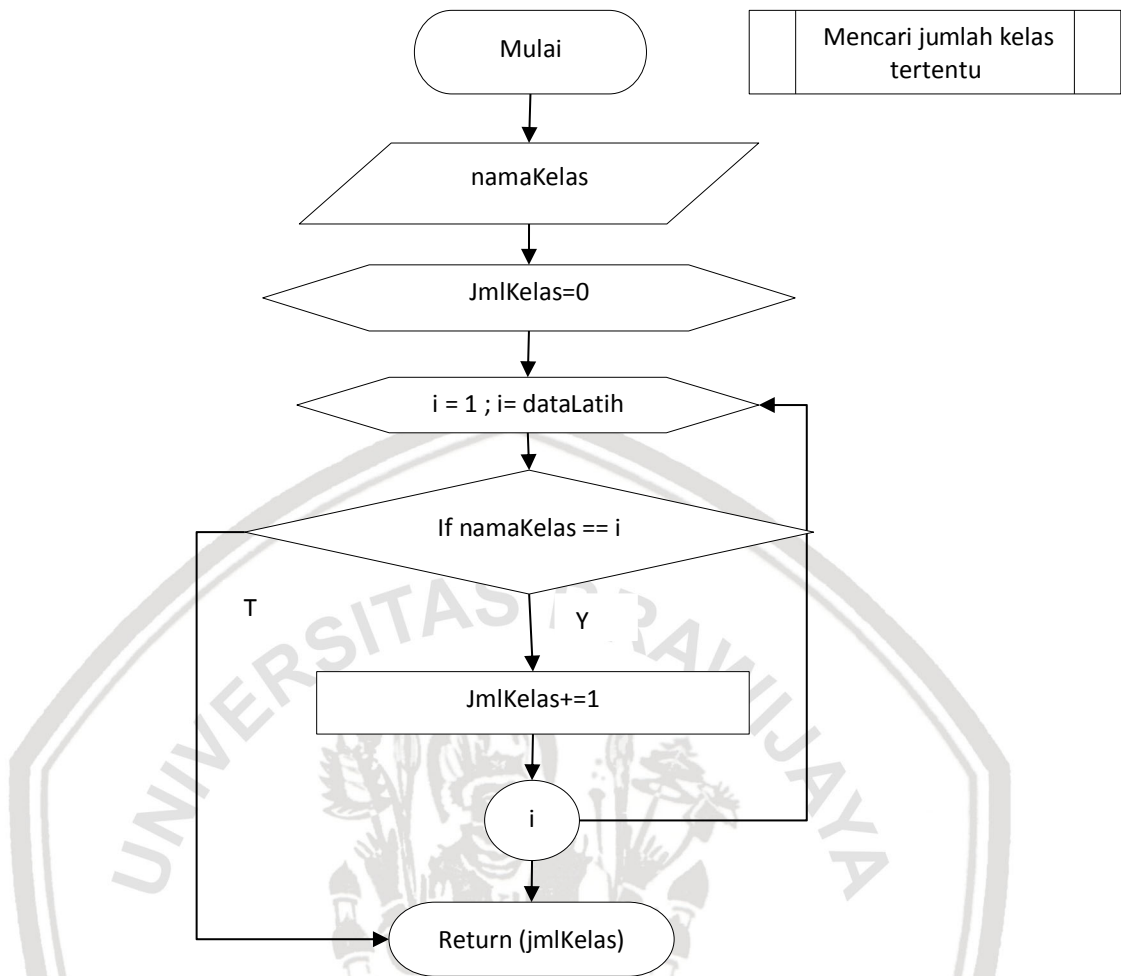


Gambar 4.13 Diagram alir Proses Training

a. Mencari jumlah kelas tertentu

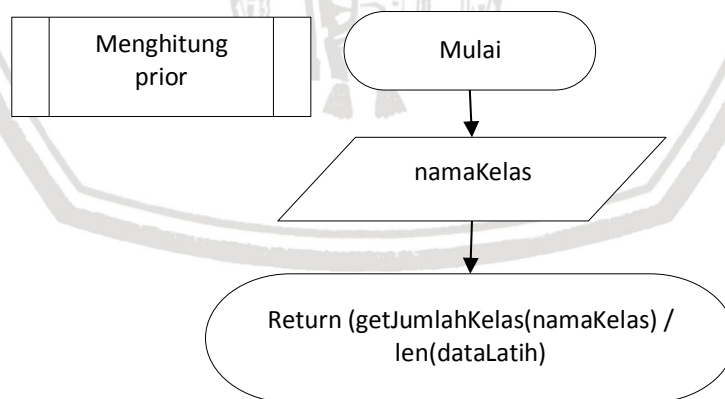
Berdasarkan Gambar 4.14 tahapan menghitung jumlah kelas tertentu diantaranya:

1. Masukan berupa data latih dan parameter namaKelas untuk menyimpan kelas
2. Inialisasi jumlah = 0 untuk menyimpan jumlah kelas
3. Perulangan i = 1 sampai i dataLatih
4. Seleksi kondisi jika namaKelas = i
5. Jika ya maka jumlah ditambah 1
6. Jika sudah memenuhi i maka perulangan selesai
7. Jika tidak maka selesai



Gambar 4.14 Diagram alir menghitung jumlah kelas tertentu

b. Menghitung *prior*

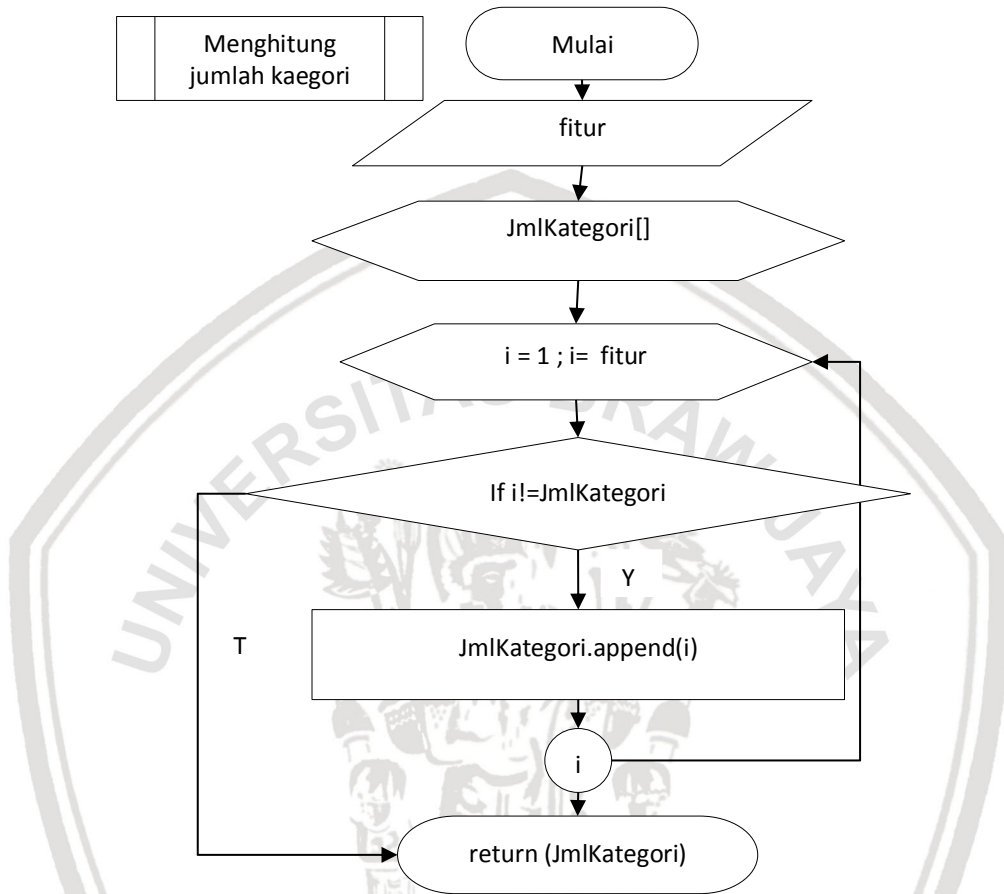


Gambar 4.15 Diagram alir Menghitung *Prior*

Berdasarkan Gambar 4.15 tahapan menghitung *prior* di antaranya:



1. Masukkan berupa parameter namaKelas
 2. Perhitungan *prior* dengan membagi jumlah kelas dengan panjang seluruh data latih akan menghasilkan nilai *prior* dari masing2 kelas saat parameter diisi dengan kelas paham, cukup paham dan kurang paham.
 3. Menghasilkan keluaran *prior* dari setiap kelas
- c. **Menghitung jumlah kategori**

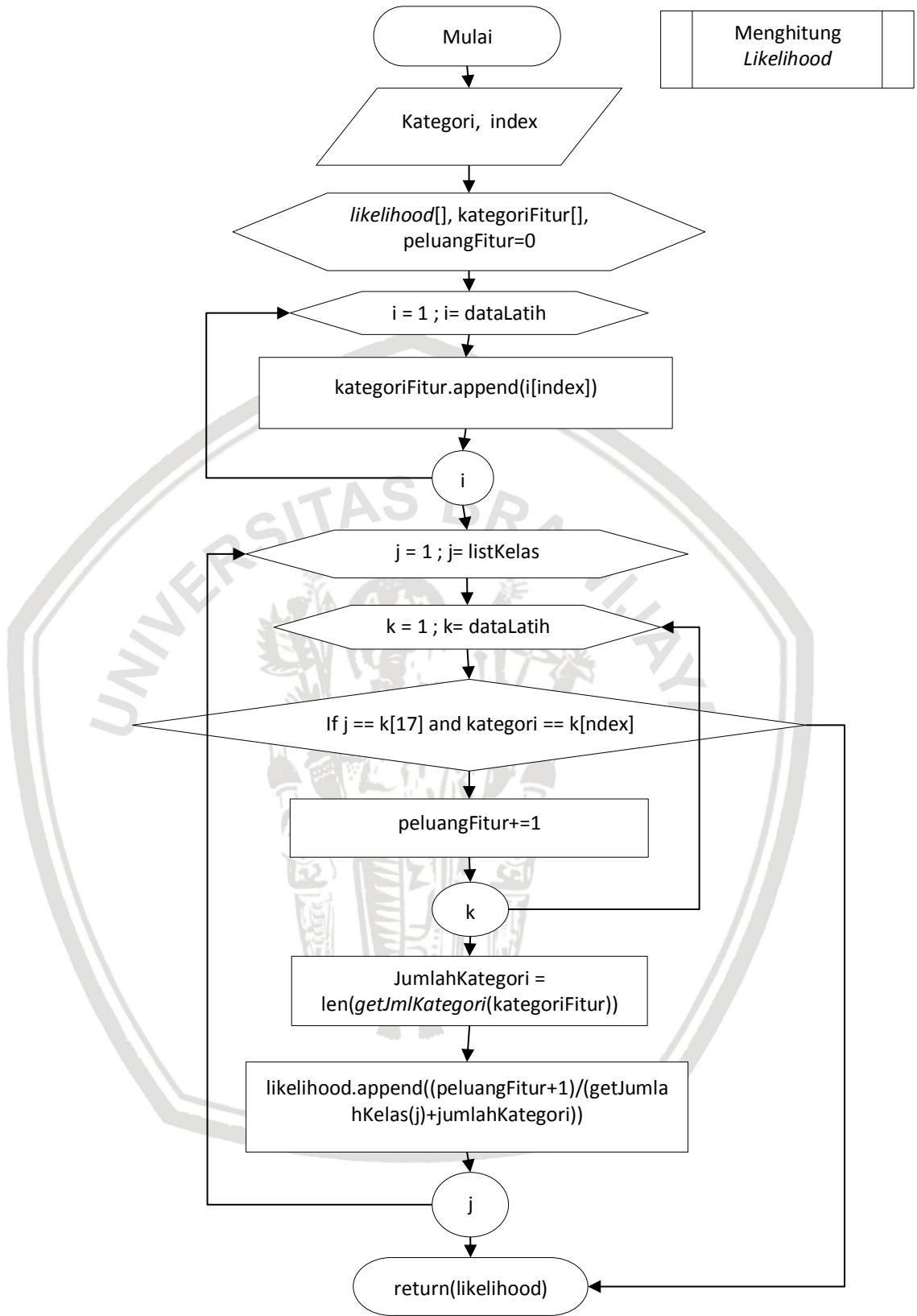


Gambar 4.17 Diagram alir Mencari kategori unik

Berdasarkan Gambar 4.17 tahapan mencari jumlah kategori diantaranya:

1. Masukan berupa parameter fitur
2. Inisialisasi array JmlKategori
3. Perulangan i = 1 sampai fitur
4. Seleksi kondisi jika jika i tidak sama dengan JmlKategori
5. Jika ya maka elemen I ditambahkan kedalam array JmlKategori
6. Jika sudah memenuhi i maka perulangan selesai
7. Menghasilkan keluaran berupa jumlah kategori

d. **Menghitung Likelihood**



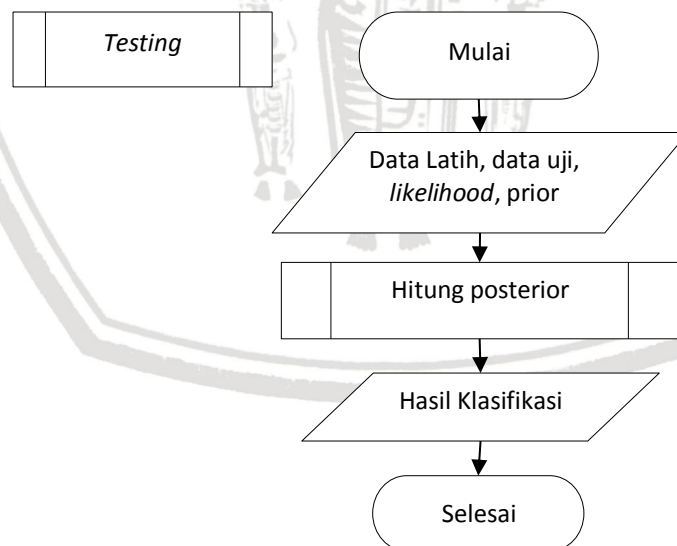
Gambar 4.18 Diagram alir menghitung *likelihood* dengan smoothing

Berdasarkan Gambar 4.18 tahapan menghitung *likelihood* dengan smoothing diantaranya:

1. Masukan berupa parameter kategori dan index
2. Inisialisasi array *likelihood*, kategoriFitur dan inisialisasi peluangFitur = 0
3. Perulangan i = 1 sampai i = dataLatih
4. Menambahkan elemen dari i dengan index index kedalam elemen kat
5. Jika sudah memenuhi i maka perulangan selesai
6. Perulangan j = 1 i = sampai i = listKelas
7. Perulangan k = 1 k = sampai i = listKelas
8. Seleksi kondisi jika j sama dengan k index ke 17 yaitu kelas dan katgeori = index
9. Jika benar maka peluangFitur ditambah 1
10. Jika sudah memenuhi k maka perulangan selesai
11. Menyimpan panjang dari *method* *getJumlahKategori* dengan parameter kategoriFiur ke dalam variabel jumlahKategori
12. Menghitung *likelihood* denga variabel peluangFitur +1 kemudian dibagi dengan *method* *getJumlahKelas* dengan parameter j + variabel jumlahKategori kemudian ditambahkan kedalam elemen *likelihood*
13. Jika sudah memenuhi j maka perulangan selesai
14. Menghasilkan keluaran *likelihood* dari setiap fitur

4.2.2.2 Testing

Dari data latih yang sudah dihitung, makan perlu data testing untuk menentukan kelas dari data tersebut.



Gambar 4.19 Diagram alir Proses Testing

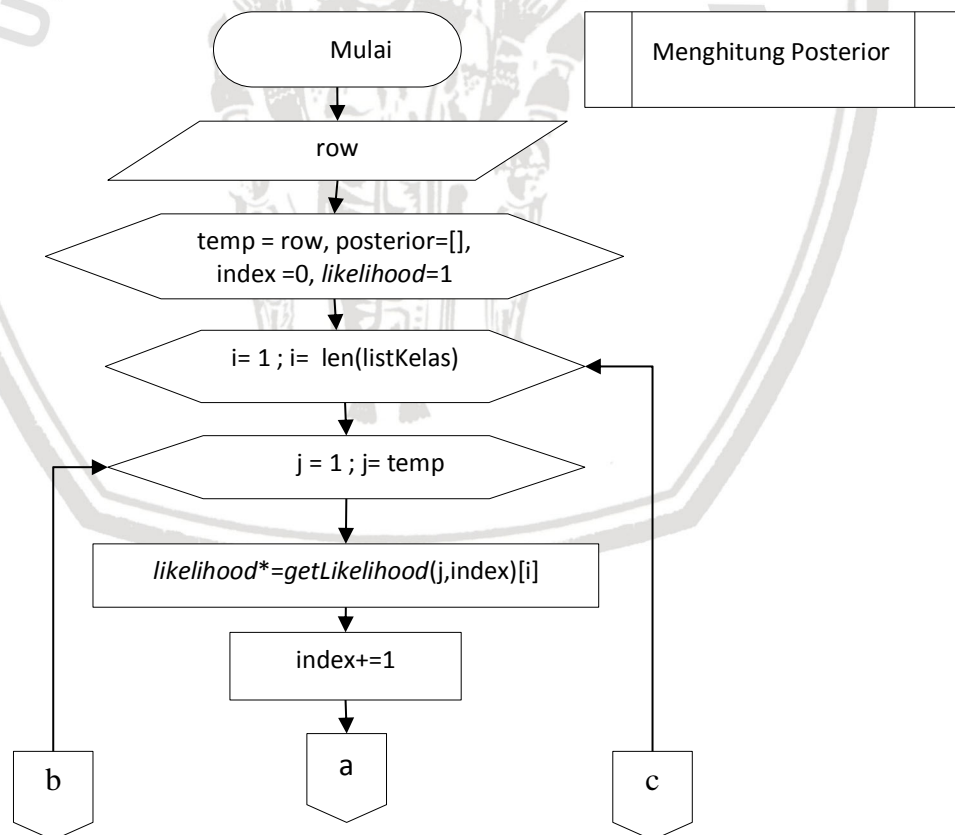
Bedasarkan Gambar 4.19 tahapan dalam testing adalah sebagai berikut:

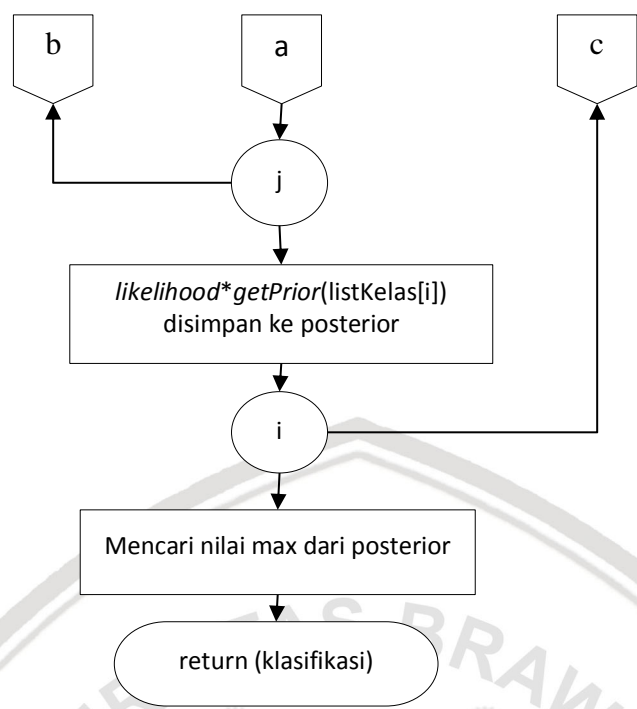
1. Pertama kita sudah memiliki data hasil *training*, *likelihood*, *prior* dan juga data testing
2. Menghitung *posterior* $P(H|X) = P(X_i|H) P(H)$ dengan kata lain *posterior* = *likelihood* * *prior*
3. Menghasilkan keluaran hasil klasifikasi

a. Menghitung *posterior*

Berdasarkan Gambar 4.20 tahapan menghitung *posterior* diantaranya:

1. Masukan berupa data uji, data latih sert hasil perhitungan *likelihood* dan *prior* serta parameter row
2. Inialisasi array *posterior*, inialisasi test=row, index=0, *likelihood*=1
3. Perulangan i = 1 sampai i = panjang dari listKelas
4. Perulangan j = 1 sampai j = test
5. Proses perhitungan *method* *getLikelihood* dengan parameter j dan index pada index i
6. Index ditambah dengan 1
7. Jika sudah memenuhi j maka perulangan selesai
8. Menghitung *posterior* dengan *likelihood* dikali *method* *getPrior* parameter listKelas index ke i
9. Mencari argument max dari *posterior*
10. Menghasilkan hasil klasifikasi

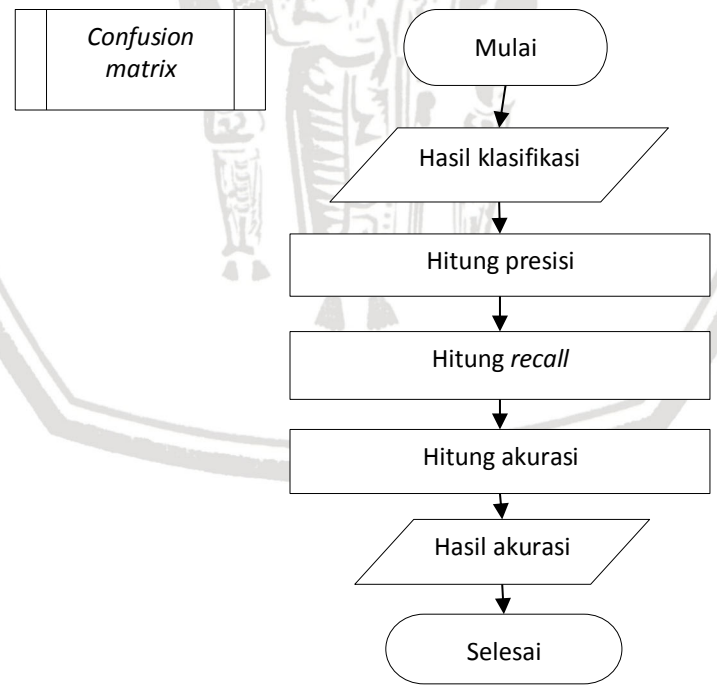




Gambar 4.20 Diagram alir menghitung *posterior*

4.2.3 Menghitung akurasi dengan *Confusion Matrix*

Confusion Matrix digunakan untuk mengukur kinerja klasifikasi dengan membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya seperti pada Gambar 4.12.



Gambar 4.21 Diagram alir *Confusion Matrix*



4.3 Manualisasi

Manualisasi untuk penelitian ini digunakan data set sebanyak 15 data siswa yang dibagi menjadi dua bagian data yaitu data latih sebanyak 15 data dan data uji sebanyak 5 data yang ditunjukkan pada tabel berikut:

Tabel 4.1 Data Training

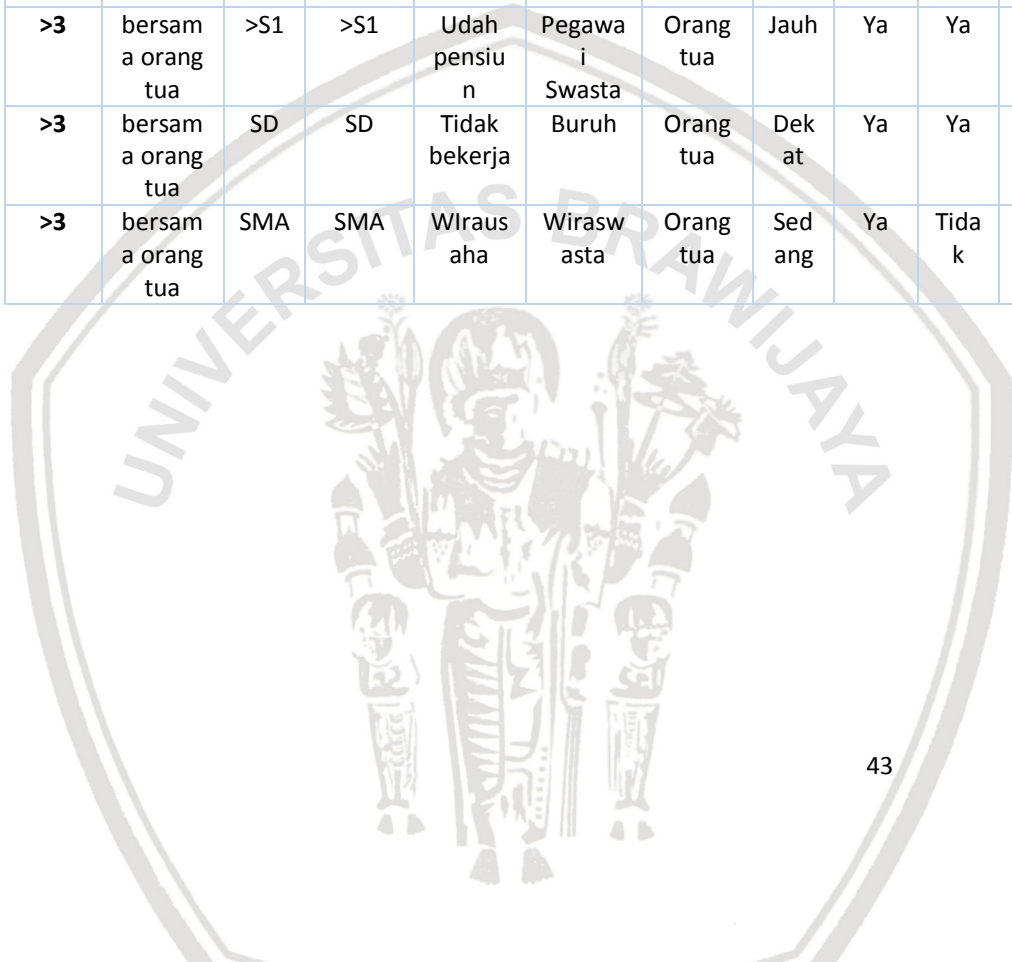
Jml anggota keluarga	Status tempat tinggal	Pendidikan terakhir ibu	Pendidikan terakhir ayah	Pekerjaan ibu	Pekerjaan ayah	Pengasuh	Jarak rumah ke sekolah	Dukungan keluarga	Ikut ekstrakurikuler	Mengulang pelajaran di rumah	Posisi duduk di kelas	Internet di rumah	Mengikuti bimbingan belajar	Lama belajar di rumah	Jenis bacaan	Lama membaca di rumah	Kelas
>3	bersama orang tua	SD	SMA	Tidak bekerja	PNS	Orang tua	Sedang	Ya	Ya	Ya	2	Tidak ada	Ya	1-2 jam	Buku pelajaran	> 2 jam	paham
<=3	bersama orang tua	SD	SMA	Tidak bekerja	Tidak bekerja	Orang tua	Jauh	Ya	Ya	Tidak	2	Tidak ada	Tidak	1-2 jam	Buku pelajaran	1-2 jam	paham
>3	bersama orang tua	SD	SMA	Tidak bekerja	Pegawai Swasta	Orang tua	Jauh	Ya	Ya	Ya	2	Tidak ada	Tidak	< 1 jam	Artikel	< 1 jam	cukup paham
>3	bersama orang tua	SD	SMA	Tidak bekerja	Pegawai Swasta	Orang tua	Jauh	Ya	Ya	Ya	2	Tidak ada	Tidak	< 1 jam	Novel	1-2 jam	cukup paham
>3	bersama orang tua	>S1	>S1	PNS	PNS	Orang tua	Sedang	Ya	Ya	Ya	2	Tidak ada	Tidak	1-2 jam	Novel	< 1 jam	paham

Jml anggota keluarga	Status tempat tinggal	Pendidikan terakhir ibu	Pendidikan terakhir ayah	Pekerjaan ibu	Pekerjaan ayah	Pengasuh	Jarak rumah kesekolah	Dukungan keluarga	Ikut ekstrakurikuler	Mengulang pelajaran di rumah	Posisi duduk di kelas	Internet di rumah	Mengikuti bimbingan belajar	Lama belajar di rumah	Jenis bacaan	Lama membaca di rumah	Kelas
>3	bersama orang tua	Tidak sekolah	Tidak sekolah	Tidak bekerja	Wiraha	Orang tua	Sedang	Ya	Ya	Ya	4	Ada	Ya	< 1 jam	Buku pelajaran	< 1 jam	kurang paham
>3	bersama orang tua	>S1	>S1	PNS	PNS	Orang tua	Sedang	Ya	Ya	Ya	2	Tidak ada	Tidak	< 1 jam	Novel	< 1 jam	paham
>3	bersama orang tua	>S1	>S1	PNS	PNS	Orang tua	Sedang	Ya	Ya	Ya	2	Tidak ada	Tidak	< 1 jam	Novel	< 1 jam	paham
>3	bersama orang tua	SD	SD	Tidak bekerja	Wiraha	Orang tua	Sedang	Ya	Ya	Ya	3	Ada	Ya	< 1 jam	Buku pelajaran	< 1 jam	cukup paham
>3	bersama orang tua	SD	SMA	Tidak bekerja	Tidak bekerja	Orang tua	Jauh	Ya	Ya	Tidak	5	Ada	Ya	< 1 jam	Artikel	1-2 jam	cukup paham

Tabel 4.2 Data Testing

Jml anggota keluarga	Status tempat tinggal	Pendidikan terakhir ibu	Pendidikan terakhir ayah	Pekerjaan ibu	Pekerjaan ayah	Pengasuh	Jarak rumah ke sekolah	Dukungan keluarga	Ikut ekstrakurikuler	Mengulang pelajaran di rumah	Posisi duduk di kelas	Internet di rumah	Mengikuti bimbingan belajar	Lama belajar di rumah	Jenis bacaan	Lama membaca di rumah	Kelas
<=3	bersama orang tua	SMA	SMA	Pegawai Swasta	Wirusaha	Orang tua	Sedang	Ya	Tidak	Tidak	4	Tidak ada	Tidak	< 1 jam	Novel	< 1 jam	cukup paham
>3	bersama orang tua	SMA	SMA	Tidak bekerja	Buruh	Orang tua	Dekat	Ya	Ya	Tidak	4	Ada	Tidak	< 1 jam	Buku pelajaran	< 1 jam	cukup paham
>3	bersama orang tua	SMA	SMA	Tidak bekerja	Pegawai Swasta	Orang tua	Sedang	Ya	Ya	Tidak	1	Tidak ada	Tidak	< 1 jam	Novel	< 1 jam	cukup paham
>3	bersama orang tua	SMA	SMA	Tidak bekerja	Pegawai Swasta	Orang tua	Sedang	Ya	Ya	Ya	1	Ada	Ya	< 1 jam	Novel	1-2 jam	paham
>3	bersama orang tua	SMA	SMA	Wirusaha	Wirusaha	Orang tua	Sedang	Ya	Tidak	Tidak	3	Tidak ada	Tidak	< 1 jam	Novel	< 1 jam	cukup paham
>3	bersama orang tua	SMA	>S1	Pegawai Swasta	Pengajar	Orang tua	Sedang	Ya	Ya	Ya	3	Ada	Tidak	< 1 jam	Wattpad wkwk	> 2 jam	paham
>3	bersama orang tua	SMP	SMP	Buruh	Buruh	Orang tua	Jauh	Ya	Ya	Ya	1	Ada	Tidak	1-2 jam	Buku pelajaran	1-2 jam	paham

Jml anggota keluarga	Status tempat tinggal	Pendidikan terakhir ibu	Pendidikan terakhir ayah	Pekerjaan ibu	Pekerjaan ayah	Pengasuh	Jarak rumah sekolah	Dukungan keluarga	Ikut ekstrakurikuler	Mengulang pelajaran di rumah	Posisi duduk di kelas	Internet di rumah	Mengikuti bimbingan belajar	Lama belajar di rumah	Jenis bacaan	Lama membaca di rumah	Kelas
<=3	bersama orang tua	>S1	>S1	Tidak bekerja	PNS	Orang tua	Dekat	Ya	Tidak	Ya	1	Ada	Ya	> 2 jam	Buku pelajaran	1-2 jam	paham
>3	bersama orang tua	>S1	>S1	Udah pensiun	Pegawai Swasta	Orang tua	Jauh	Ya	Ya	Ya	3	Tidak ada	Tidak	< 1 jam	Novel	< 1 jam	cukup paham
>3	bersama orang tua	SD	SD	Tidak bekerja	Buruh	Orang tua	Dekat	Ya	Ya	Ya	3	Tidak ada	Ya	< 1 jam	Novel	1-2 jam	cukup paham
>3	bersama orang tua	SMA	SMA	Wirusaha	Wiraswasta	Orang tua	Sedang	Ya	Tidak	Ya	1	Ada	Ya	< 1 jam	Buku pelajaran	< 1 jam	cukup paham



4.3.1 Seleksi Fitur Information Gain

Perhitungan seleksi fitur *Information Gain* dilakukan dengan menggunakan data latih. Contoh diambil dari fitur 1 yaitu fitur jumlah anggota keluarga. Perhitungan seleksi fitur ditunjukkan oleh tabel berikut:

Tabel 4.3 Information Gain

$I(D)$	$\frac{ D_j }{ D }$	$I(D_j)$	$-\sum_{j=1}^v \frac{ D_j }{ D } \times I(D_j)$	$Gain(A) = I(D) - I(A)$
1.5219	0.85	0.311278	0.264586	1.257342
	0.15	0		

Hitung probabilitas kemunculan setiap kelas dari kelas tersebut $P(H)$. dari 10 data, terdiri dari 4 kelas paham, 4 kelas cukup paham dan 2 kelas kurang paham. Hasil probabilitas ditunjukkan pada tabel berikut:

Tabel 4.4 Probabilitas P(H)

Paham	cukup paham	kurang paham
0.4	0.4	0.2

$$paham = \frac{4}{10}$$

$$cukup paham = \frac{4}{10}$$

$$kurang paham = \frac{2}{10}$$

Setelah mendapatkan probabilitas dari kelas tersebut, hitung nilai fitur *entropy* awalnya dengan menjumlahkan seluruh probabilitas yang sudah dikali dengan log berbasis 2 karena informasi dikodekan berbasis bit.

$$= -\left(\frac{4}{10} \log_2 \frac{4}{10}\right) - \left(\frac{4}{10} \log_2 \frac{4}{10}\right) - \left(\frac{2}{10} \log_2 \frac{2}{10}\right) = 1.5219$$

D merupakanh impunan kasus, m merupakan Jumlah partisi D dan pi adalah proporsi dari i terhadap D.

Selanjutnya menghitung probabilitas dari tiap fitur yang sudah dipisahkan kemudian dikalikan dengan *entropy* nya.



D merupakan himpunan kasus A adalah fitur yang sedang dihitung v adalah jumlah partisi dari fitur A kemudian $|D_j|$ adalah jumlah kasus pada fitur $|D|$ jumlah seluruh kasus yang ada pada fitur.

$$\frac{|D|_{>3}}{|D|} = \frac{9}{10}$$

$$-\left(\frac{3}{4} \log_2 \frac{3}{4}\right) - \left(\frac{4}{4} \log_2 \frac{4}{4}\right) - \left(\frac{2}{2} \log_2 \frac{2}{2}\right) = 0.3113$$

Pada fitur jumlah anggota keluarga dengan kategori >3 didapatkan hasil probabilitasnya 0.85 dan hasil *entropy* sebesar 0.62877.

$$\frac{|D|_{\leq 3}}{|D|} = \frac{1}{10}$$

$$-\left(\frac{1}{4} \log_2 \frac{1}{4}\right) - (0) - (0) = 0$$

Pada fitur jumlah anggota keluarga dengan kategori ≤ 3 didapatkan hasil probabilitasnya 0.85 dan hasil *entropy* sebesar 0

Lalu hasil *entropy* dikalikan dengan probabilitas dari fitur tersebut sehingga didapatkan hasil:

$$I(A) = \frac{9}{10} (0.3113) + \frac{1}{10} 0 = 0.28015$$

Selanjutnya kita menghitung nilai *Gain* nya yaitu dengan mengurangi total *entropy* dengan *entropy* dari A:

$$\text{Gain} = 1.521928 - 0.28015 = 1.241778$$

Dilanjutkan hingga fitur terakhir yaitu fitur ke 17. *Gain* (A) merupakan reduksi yang diharapkan di dalam *entropy* yang disebabkan oleh pengenalan nilai fitur dari A. Fitur yang memiliki nilai *information Gain* terbesar selanjutnya dipilih sebagai uji fitur himpunan S. Lalu suatu simpul dibuat dan diberi label sesuai dengan fitur tersebut, kemudian cabang-cabang dibuat untuk masing-masing nilai fitur yang lain.

4.3.2 Metode Naïve Bayes

4.3.2.1 Training

Peluang kelas terhadap kelas itu sendiri

Tabel 4.5 Probabilitas P(H)

P(H)	P(paham)	P(cukup paham)	P (kurang paham)
	0.4	0.4	0.2

P(X|H) Peluang data sampel fitur dengan mempertimbangkan kelas H. Salah satunya fitur jumlah anggota keluarga. Hasil dibawah ini sudah dilakukan *smoothing*.

Tabel 4.6 Probabilitas P(X|H) dengan Smoothing

Jumlah anggota keluarga	paham	cukup paham	kurang paham
>3	0.666666667	0.833333333	0.75
<=3	0.333333333	0.166666667	0.25

$$P(> 3|paham) = 0.666666667$$

$$P(\leq 3|paham) = 0.333333333$$

$$P(> 3|cukup paham) = 0.833333333$$

$$P(\leq 3|cukup paham) = 0.166666667$$

$$P(> 3|kurang paham) = 0.75$$

$$P(\leq 3|kurang paham) = 0.25$$

Semua fitur dicari peluang terhadap kelasnya.

4.3.2.2 Testing

Hasil peluang dari setiap fitur terhadap kelas akan digunakan untuk perhitungan data uji. Testing dilakukan dengan mengalikan semua probabilitas dari fitur terhadap kelas dengan *prior* atau probabilitas kelas terhadap kelas itu sendiri.

$$=P(\leq 3|paham) * P(Tinggal bersama dengan orang tua|paham) * P(SMA|paham) * P(SMA|paham) * P(Pegawai Swasta|paham) * P(Wirusaha|paham) * P(Orang tua|paham) * P(Sedang|paham) * P(Ya|paham) * P(Tidak|paham) * P(Tidak|paham) * P(4|paham) * P(Tidak ada|paham) * P(Tidak|paham) * P(< 1 jam|paham) * P(Novel|paham) * P(< 1 jam|paham) * P(paham)$$

Berdasarkan data uji yang ada pada tabel terdapat 5 data uji. Dan hasilnya adalah sebagai berikut:



Tabel 4.7 Hasil Klasifikasi

Data ke-	Paham	cukup paham	kurang paham	Pemenang (nilai tertinggi)	Kelas	Kelas sebenarnya
1	0.00037037	0.000493827	0.000334821	0.000493827	cukup paham	cukup paham
2	0.000246914	0.000308642	0.000502232	0.000502232	kurang paham	kurang paham
3	0.000740741	0.003703704	0.000502232	0.003703704	cukup paham	cukup paham
4	0.000266667	0.002222222	0.000251116	0.002222222	cukup paham	paham
5	0.000740741	0.000617284	0.000502232	0.000740741	paham	paham

4.3.3 Pengujian akurasi

Menghitung akurasi dengan *Confusion Matrix* yaitu dengan menghitung nilai presisi, *recall* dan akurasi. Pengujian dilakukan dengan hasil klasifikasi yang sudah dilakukan seleksi fitur. Sebagai contoh, dipilih sebanyak 5 fitur yang memiliki nilai *Gain* paling tinggi.

4.3.3.1 Akurasi menggunakan 5 fitur

Setelah dilakukan seleksi fitur didapatkan hasil sebanyak 4 data benar dan 1 data yang salah atau tidak sesuai. Dapat dilihat melalui tabel di bawah:

Tabel 4.8 Akurasi

Data ke-	Hasil Kelas	Kelas sebenarnya	Hasil
1	cukup paham	cukup paham	benar
2	kurang paham	kurang paham	benar
3	cukup paham	cukup paham	benar
4	cukup paham	paham	salah
5	paham	paham	benar

Confusion Matrix dari data uji yang belum dilakukan seleksi fitur dapat dilihat dari **Tabel 4.9**. baris pertama kesamping adalah merupakan hasil

klasifikasi dari sistem dan kolom pertama adalah hasil yang seharusnya atau yang sesuai dengan dataset

Tabel 4.9 Confusion Matrix

	paham	cukup paham	kurang paham
paham	1	0	0
cukup paham	1	2	0
kurang paham	0	0	1

$$\text{Presisi paham} = \frac{1}{0 + 1} = 1$$

$$\text{Presisi cukup paham} = \frac{2}{1 + 2} = 0.6667$$

$$\text{Presisi kurang paham} = \frac{1}{0 + 1} = 1$$

$$\text{Presisi} = \frac{1 + 0.6667 + 1}{3} = 0.889$$

$$\text{Recall paham} = \frac{1}{1 + 1} = 0.5$$

$$\text{Recall cukup paham} = \frac{2}{2 + 0} = 1$$

$$\text{Recall kurang paham} = \frac{1}{1 + 1} = 0.5$$

$$\text{Recall} = \frac{0.5 + 1 + 0.5}{3} = 0.6667$$

$$\text{Akurasi paham} = \frac{1 + 3}{1 + 3 + 0 + 1} = 0.8$$

$$\text{Akurasi cukup paham} = \frac{2 + 2}{2 + 2 + 1 + 0} = 0.8$$

$$\text{Akurasi kurang paham} = \frac{1 + 3}{1 + 3 + 0 + 1} = 0.8$$

$$\text{Akurasi} = \frac{0.8 + 0.8 + 0.8}{3} = 0.6$$

BAB 5 IMPLEMENTASI

5.1 Kode Program

5.1.1 Seleksi Fitur *Information Gain*

Seleksi fitur ini dilakukan dengan beberapa tahap yaitu yang pertama adalah menghitung nilai *entropy* dari setiap kelas yang ada yaitu *entropy* dari kelas paham, cukup paham dan kurang paham. Setelah itu menghitung *entropy* dari setiap fitur yang terdiri dari 17 fitur. Kemudian setelah mendapatkan nilai *entropy* kelas dan *entropy* fitur langkah selanjutnya adalah menghitung nilai *Gain* dengan mengurangkannya. Setelah itu tentukan batas *threshold* nilai *entropy* yang ingin di seleksi. Proses seleksi fitur dapat dilihat melalui kode program berikut

```
Menghitung nilai entropy dari semua kelas
1  def entropy(a,b,c):
2      if a==0 and b==0 and c==0:
3          return(0)
4      if a==0 and b==0:
5          return(-(c*math.log(c,2)))
6      if a==0 and c==0:
7          return(-(b*math.log(b,2)))
8      if c==0 and b==0:
9          return(-(a*math.log(a,2)))
10     if a==0:
11         return(-(b*math.log(b,2))-(c*math.log(c,2)))
12     if b==0:
13         return(-(a*math.log(a,2))-(c*math.log(c,2)))
14     if c==0:
15         return(0-(a*math.log(a,2))-(b*math.log(b,2)))
16     return(-(a*math.log(a,2))-(b*math.log(b,2))
17         (c*math.log(c,2)))
18 def informationGain(data):
19     #menghitung nilai entropy dari masing2 kelas
20     entropyKelas=entropy(getPrior("paham"),getPrior("cukup
21     paham"),getPrior("kurang paham"))
```

Kode Program 5.1 Menghitung nilai *entropy* dari semua kelas

1. Baris 1 terdapat *method entropy* yang berisi perhitungan *entropy*, memiliki parameter a, b dan c.

2. Baris 2 melakukan seleksi kondisi apabila a=0, b=0, dan c=0 jika ya maka akan memproses hasil *entropy* menjadi 0
3. Baris 4 seleksi kondisi apabila a=0 dan b=0 jika ya maka akan memproses - (c*log₂(c))
4. Baris 6 seleksi kondisi apabila a=0 dan c=0 jika ya maka akan memproses - (b*log₂(b))
5. Baris 8 seleksi kondisi apabila c=0 dan b=0 jika ya maka akan memproses - (a*log₂(a))
6. Baris 10 seleksi kondisi apabila a=0 jika ya maka akan memproses -(b*log₂(b))- (c*log₂(c))
7. Baris 12 seleksi kondisi apabila b=0 jika ya maka akan memproses -(a*log₂(a))- (c*log₂(c))
8. Baris 14 seleksi kondisi apabila c=0 jika ya maka akan memproses -(a*log₂(a))- (b*log₂(b))
9. Baris 16 jika tidak memenuhi seleksi diatas maka akan memproses -(a*log₂(a))- (b*log₂(b))- (c*log₂(c))
10. Baris 18 *method informationGain* dengan parameter data
11. Baris 20 variabel *entropyKelas* memanggil *method entropy* dengan mengisi nilai a b c dengan kelas paham, cukup paham dan kurang paham, yang akan menghasilkan hasil *entropy* dari setiap kelas tersebut.

```
Menghitung nilai entropy dari semua fitur
1
2     #pertama mencari kategori apa saja yang ada pada fitur
3     jmlKategoriUnik = []
4     semuaKategori = []
5     for i in range(len(data[0])):
6         kategori=[]
7         for j in range(len(data)):
8             kategori.append(data[j][i])
9             semuaKategori.append(kategori)
10            jmlKategoriUnik.append(getJmlKategori(kategori))
11
12            #kedua mencari probabilitas kategori
13            probabilitasKategori=[]
14            for i in range(len(semuaKategori)):
15                temp=[]
16                for j in jmlKategoriUnik[i]:
17                    temp.append(semuaKat[i].count(j)/len(dataLatih))
18                probabilitasKategori.append(temp)
19
```



```

20     #ketiga mencari likelihood
21     peluangLikelihood=[]
22     for i in range(len(semuaKategori)):
23         temp=[]
24         for j in jmlKategoriUnik[i]:
25             temp.append(getStandardLikelihood(j,i))
26         peluangLikelihood.append(temp)
27
28     #keempat hitung entropy dengan likelihood
29     entropyKategori=[]
30     for i in range(len(semuaKategori)):
31         temp=[]
32         for j in peluangLikelihood[i]:
33             temp.append(entropy(j[0],j[1],j[2]))
34         entropyKategori.append(temp)
35
36     #kelima melakukan sumproduct
37     entropyFitur=[]
38     for i in range(len(semuaKategori)):
39         temp=0
40         for j in range(len(entropyKategori[i])):
41             temp+=probabilitasKategori[i][j]*entropyKategori[i][j]
42         entropyFitur.append(temp)
43

```

Kode Program 5.2 Menghitung *entropy* dari semua fitur

1. Baris 1 *method* *informationGain* dengan parameter data
2. Baris 3 dan 4 inialisasi array *JmlKategoriUnik* dan array *arr*
3. Baris 5 perulangan *i* sampai panjang data index ke 0
4. Baris 7-8 perulangan *j* sampai panjang data kemudian menyimpan elemen yang ada pada data index ke *j* dan *i* kedalam array kategori
5. Baris 9 *i* telah mencapai panjang data maka melakukan operasi penyimpanan elemen yang ada dalam kategori ke array *arr*
6. Baris 10 memanggil *method* *getJumlahKategori* dan mengisinya dengan parameter kategori yang telah dibuat, kemudian disimpan pada array *jmlKategoriUnik* tadi.
7. Baris 13 inialisasi array *probabilitasKategori*
8. Baris 14 perulangan *i* sampai panjang array *arr*
9. Baris 16 perulangan *j* dalam *jmlKategoriUnik* index *i*
10. Baris 17 menambahkan elemen dari hasil count *j* dari array *semuaKategori* index *i* dibagi dengan panjang seluruh data latih

11. Baris 18 melakukan proses penambahan elemen dari array *ar* ke dalam array *probabilitasKategori*
12. Baris 21 inialisasi array *peluangLikelihood*
13. Baris 22 perulangan *l* sampai panjang dari array *arr*
14. Baris 24 perulangan *j* dalam *jmlKategoriUnik* index *i*
15. Baris 25 memanggil *method* *getStandardLikelihood* dengan mengisi parameter *j* dan *l* kemudian disimpan kedalam elemen array *temp*
16. Baris 26 elemen *temp* disimpan dalam array *peluangLikelihood*
17. Baris 29 inialisasi array *entropyKategori*
18. Baris 30 perulangan *i* sampai panjang dari array *arr*
19. Baris 32 perulangan *j* dalam *peluangLikelihood* index *i*
20. Baris 33 memanggil *method* *entropy* dan mengisinya dengan array *j* index 0 index 1 dan index 2 kemudian elemennya disimpan dalam array *temp*
21. Baris 34 menyimpan elemen dari array *temp* edalam array *entropyKategori*
22. Baris 37 inialisasi *entropyFitur*
23. Baris 38 Perulangan *i* sampai panjang dari array *semuaKategori*
24. Baris 40 perulangan *j* dalam panjang dari *entropyKategori* index ke *i*
25. Baris 42 melakukan proses *sumproduct* antara peluang kaategori index *l* dan *j* dengan *entropy* kategori index *l* dan *j*
26. Baris 43 menambahkan elemen dari *temp* kedalam array *entropyFitur*

Menghitung nilai <i>entropy</i>	
2	<i>Gain</i> =[]
3	for <i>i</i> in range(len(<i>entropyFitur</i>)):
4	<i>Gain</i> .append(<i>entropyKelas</i> - <i>entropyFitur</i> [<i>i</i>])

Kode Program 5.3 Menghitung nilai *Gain*

1. Baris 2 inialisasi array *Gain*
2. Baris 3 perulangan *i* sampai panjang dari array *entropyFitur*
3. Baris 4 memanggil variabel *entropyKelas* dikurang dengan *entropyFitur* index *i* dan menyimpannya dalam array *Gain* yang sudah diinisialisasi

Menentukan nilai <i>threshold</i>	
2	<i>fiturBaru</i> =[]
3	for <i>i</i> in range(len(<i>Gain</i>)):
4	if <i>Gain</i> [<i>i</i>]>=0 :
5	<i>fiturBaru</i> .append(<i>i</i>)
6	return(<i>fiturBaru</i>)

Kode Program 5.4 Menentukan *threshold*

1. Baris 2 inialisasi array *fiturBaru*
2. Baris 3 perulangan *i* sampai panjang dari array *Gain*



3. Baris 4 seleksi kondisi apabila nilai *Gain* lebih dari 0(contoh *threshold* yang dipakai)
4. Baris 5 mengembalikan nilai fitur baru dengan membuang fitur-fitur yang memiliki nilai *Gain* dibawah *threshold*

5.1.2 Klasifikasi *Naïve bayes*

Klasifikasi *naïve bayes* memiliki beberapa tahap dalam proses perhitungannya yang pertama adalah menghitung jumlah kelas tertentu yang akan berguna untuk program berikutnya yaitu mendapatkan nilai prior. Setelah itu mencari jumlah kategori dari setiap fitur guna melakukan perhitungan likelihood yang menggunakan proses *smoothing*. Setelah mendapatkan nilai prior dan likelihood maka langkah selanjutnya adalah menghitung nilai posterior lalu dan dicari nilai maksimalnya dan akan mendapatkan hasil klasifikasinya.

Menghitung jumlah kelas tertentu	
1 2 3 4 5 6	<pre>def getJumlahKelas(namaKelas): jmlKelas = 0 for i in dataLatih: if namaKelas in i: jmlKelas +=1 return (jmlKelas)</pre>

Kode Program 5.5 Menghitung jumlah kelas tertentu

1. Baris 1 membuat *method* *getJumlahKelas* dengan parameter *namaKelas*
2. Baris 2 inisialisasi variable *jmlKelas* = 0
3. Baris 3 perulangan *i* sebanyak *dataLatih*
4. Baris 4 seleksi kondisi jika variable *namaKelas* dalam *i*
5. Baris 5 melakukan proses penambahan variable *jmlKelas* dengan angka 1
6. Baris 6 mengembalikan variable *jmlKelas*

Menghitung <i>prior</i>	
1 2	<pre>def getPrior(namaKelas): return (getJumlahKelas(namaKelas)/len(dataLatih))</pre>

Kode Program 5.6 Menghitung *prior*

1. Baris 1 membuat *method* *getPrior* dengan parameter *namaKelas*
2. Baris 2 memanggil *method* *getJumlahKelas* kemudian dibagi dengan panjang dari *dataLatih* untuk mendapatkan nilai *prior* dari setiap kelas

Mencari jumlah kategori	
1 2 3	<pre>def getJmlKategori(fitur): JmlKategori=[] for i in fitur:</pre>



```

4         if i not in JmlKategori:
5             JmlKategori.append(i)
6         return JmlKategori
    
```

Kode Program 5.7 Mencari jumlah kategori

1. Baris 1 membuat *method* *getJmlKategori* dengan parameter fitur
2. Baris 2 inialisasi array *JmlKategori*
3. Baris 3 perulangan *i* dalam fitur
4. Baris 4 dan 5 seleksi kondisi apabila *i* tidak ada dalam array *JmlKategori* maka elemen dari *I* masuk ke array *JmlKategori*
5. Baris 6 mengembalikan nilai *JmlKategori*

```

Menghitung likelihood
1  def getLikelihood(kategori, index) :
2      likelihood = []
3      kategoriFitur = []
4      for i in dataLatih:
5          kategoriFitur.append(i[index])
6      for j in listKelas:
7          peluangFitur = 0
8          for k in dataLatih:
9              if j == k[17] and kategori == k[index]:
10                 peluangFitur+=1
11                 jumlahKategori = len(getJmlKategori(kat))
12
13 likelihood.append((peluangFitur+1)/(getJumlahKelas(j)+jumlahKategori))
14     return(likelihood)
    
```

Kode Program 5.8 Menghitung *likelihood*

1. Baris 1 membuat *method* *getLikelihood* dengan parameter kategori dan indexnya
2. Baris 2-3 inialisasi array *likelihood* dan array kategoriFitur
3. Baris 4 perulangan *I* dalam dataLatih
4. Baris 5 memasukan elemen dari *i* dengan index *index* kedalam array kategoriFitur
5. Baris 6 perulangan *j* dalam listKelas
6. Baris 7 inialisasi peluangFitur =0
7. Baris 8 perulangan *k* dalam dataLatih
8. Baris 9 seleksi kondisi jika *j* = *k* index ke 17 dan kategori = *k* ke index maka variable peluangFitur ditambahkan dengan angka 1
9. Baris 11 memanggil *method* *getJmlKategori* dengan parameter kategoriFitur
10. Baris ke 13 melakukan perhitungan *likelihood* dengan variable peluangFitur ditambah 1 dan dibagi dengan *method* *getJumlahKelas* parameter *j* ditambah



jumlahKategori karena perhitungan *likelihood* ini dilakukan dengan *laplace* atau *smoothing*

Menghitung <i>posterior</i>	
1	<code>def getPosterior(row):</code>
2	<code> temp = row</code>
3	<code> posterior=[]</code>
4	<code> for i in range(len(listKelas)):</code>
5	<code> index=0</code>
6	<code> likelihood=1</code>
7	<code> for j in temp:</code>
8	<code> likelihood*=getLikelihood(j,index)[i]</code>
9	<code> index+=1</code>
10	<code> posterior.append(likelihood*getPrior(listKelas[i]))</code>
11	<code> klasifikasi=listKelas[posterior.index(max(posterior))]</code>
12	<code> print(klasifikasi)</code>
13	<code> return klasifikasi</code>

Kode Program 5.9 Menghitung *posterior*

1. Baris 1 membuat *method* `getPosterior`
2. Baris 2 -3 nisialisasi array *posterior*, inisialisasi `temp=row`, `index=0`, `likelihood=1`
3. Baris 4 perulangan `i` sampai panjang dari `listKelas`
4. Baris 7 perulangan `j` dalam `temp`
5. Baris 8 proses perhitungan meyhod `getLikelihood` dengan parameter `j` dan `index` pada indeks ke `i`
6. Baris 9 variabel indeks ditambah dengan angka 1
7. Baris 10 proses menghitung *posterior* dengan *likelihood* dikali dengan *method* `getPrior` dengan parameter `listKelas` indeks ke `i`
8. Baris 11 mencari argument `max` dari *posterior*

5.2 Screen Shot Program

```
bismillah x
C:\Users\Dell\PycharmProjects\skripsi\venv\Scripts\python.exe "D:
Fitur yang terpilih adalah:
[0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16]
fitur index ke 0 : jumlah anggota keluarga
fitur index ke 1 : Status tempat tinggal
fitur index ke 2 : Pendidikan terakhir Ibu
fitur index ke 3 : Pendidikan terakhir Ayah
fitur index ke 4 : Pekerjaan Ibu
fitur index ke 5 : Pekerjaan Ayah
fitur index ke 6 : Pengasuh
fitur index ke 8 : Dukungan Keluarga
fitur index ke 9 : Ikut ekstrakurikuler
fitur index ke 10 : Mengulang pelajaran dirumah
fitur index ke 11 : Posisi baris (bangku) duduk di kelas
fitur index ke 12 : Internet dirumah
fitur index ke 13 : Mengikuti bimbingan belajar
fitur index ke 14 : Lama belajar di rumah
fitur index ke 15 : Jenis bacaan dirumah
fitur index ke 16 : Lama membaca di rumah
```

Gambar 5.1 Fitur yang terpilih

```
Hasil Klasifikasi:
P
kp
P
P
cp
P
cp
kp
P
kp
P
P
cp
cp
P
P
cp
P
kp
P
```

Gambar 5.2 Hasil Klasifikasi



Confusion Matrix:

Predicted \ Actual	cp	kp	p	__all__
cp	4	0	0	4
kp	1	3	1	5
p	0	1	10	11
__all__	5	4	11	20

Overall Statistics:

Accuracy: 0.85
 95% CI: (0.621073173454686, 0.9679290628145363)
 No Information Rate: ToDo
 P-Value [Acc > NIR]: 0.004933407986020243
 Kappa: 0.7489539748953975
 McNemar's Test P-Value: ToDo

Class Statistics:

Classes	cp	kp	p
Population	20	20	20
P: Condition positive	15	16	11
N: Condition negative	5	4	9
Test outcome positive	16	15	11
Test outcome negative	4	5	9
TP: True Positive	14	15	10
TN: True Negative	3	4	8
FP: False Positive	2	0	1
FN: False Negative	1	1	1
TPR: (Sensitivity, hit rate, recall)	0.933333	0.9375	0.909091
TNR=SPC: (Specificity)	0.6	1	0.888889
PPV: Pos Pred Value (Precision)	0.875	1	0.909091
NPV: Neg Pred Value	0.75	0.8	0.888889
FPR: False-out	0.4	0	0.111111
FDR: False Discovery Rate	0.125	0	0.0909091
FNR: Miss Rate	0.0666667	0.0625	0.0909091
ACC: Accuracy	0.85	0.95	0.9

Gambar 5.3 Confusion Matrix



BAB 6 PENGUJIAN DAN ANALISIS

6.1 K-Fold Cross Validation

Pengujian *k-fold cross validation* dilakukan untuk menguji validitas data, disini dilakukan dengan menggunakan k sebanyak 5 dari 100 data set dibagi menjadi 80 data latih dan 20 data uji. Pada Tabel 6.1 Terdapat 5 k yang telah diuji akurasi tertingginya. Untuk dilakukan analisis penulis menggunakan k ke-5 dengan akurasi sebesar 0.9 dan nilai gain ≥ 0.2 .

Tabel 6.1 K-fold Cross Validation

K	Nilai Gain	Akurasi	Fitur yang terpilih
1	≥ -0.4	0.9	jumlah anggota keluarga, status tempat tinggal, pendidikan terakhir ibu, pendidikan terakhir ayah, pekerjaan ibu, pekerjaan ayah, pengasuh, dukungan keluarga, ikut ekstrakurikuler, mengulang pelajaran di rumah, posisi baris (bangku) duduk di kelas, mengikuti bimbingan belajar, lama belajar di rumah, jenis bacaan di rumah, lama membaca di rumah
2	≥ -0.5	0.85	jumlah anggota keluarga, status tempat tinggal, pendidikan terakhir ibu, pendidikan terakhir ayah, pekerjaan ibu, pekerjaan ayah, pengasuh, dukungan keluarga, ikut ekstrakurikuler, mengulang pelajaran di rumah, posisi baris (bangku) duduk di kelas, mengikuti bimbingan belajar, lama belajar di rumah, jenis bacaan di rumah, lama membaca di rumah
3	≥ 0	0.9	jumlah anggota keluarga, status tempat tinggal, pekerjaan ibu, pengasuh, dukungan keluarga, ikut ekstrakurikuler, lama belajar di rumah, ama membaca di rumah
4	≥ 0	0.85	status tempat tinggal, pekerjaan ibu, pengasuh, dukungan keluarga, ikut ekstrakurikuler, lama belajar di rumah, lama membaca di rumah
5	≥ 0.2	0.9	jumlah anggota keluarga, status tempat tinggal, pekerjaan ibu, pengasuh, dukungan keluarga, ikut ekstrakurikuler, mengulang pelajaran di rumah, lama belajar di rumah, jenis bacaan di rumah, lama membaca di rumah

6.2 Uji Coba Nilai *Gain* ≥ 0

Uji coba pertama adalah pada saat nilai *Gain* lebih dari atau sama dengan 0 maka fitur yang terpilih adalah fitur ke 0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16 yaitu fitur jumlah anggota keluarga, status tempat tinggal, pendidikan terakhir ibu, pendidikan terakhir ayah, pekerjaan ibu, pekerjaan ayah, pengasuh, dukungan keluarga, ikut ekstrakurikuler, mengulang pelajaran di rumah, posisi baris (bangku) duduk di kelas, internet di rumah, mengikuti bimbingan belajar, lama belajar di rumah, peringkat di kelas, jenis bacaan di rumah, lama membaca di rumah. Fitur yang tidak terpilih adalah fitur Jarak rumah. Tabel 6.2 adalah hasil perhitungan *Confusion Matrix Gain* yang memiliki nilai lebih dari sama dengan 0

Tabel 6.2 *Confusion Matrix Gain* ≥ 0

<i>Predicted</i>	cukup paham	kurang paham	paham	Jumlah
<i>Actual</i>				
cukup paham	4	0	0	4
kurang paham	1	3	1	5
paham	0	1	10	11
jumlah	5	4	11	20

Dari Tabel 6.2 diperoleh nilai *recall*, *precision* dan akurasi dari setiap kelas dapat dilihat pada Tabel 6.3. Hasil tersebut diperoleh melalui perhitungan seperti pada fungsi 3.1, 3.2 dan 3.3 yaitu menggunakan nilai TF (*True Positif*), TN (*True Negatif*) FP (*False Positif*) dan (*False Negatif*) rata-rata *recall* sebesar 0.926641, rata-rata *precision* sebesar 0.92803 dan rata-rata akurasi adalah sebesar 0.9. Nilai akurasi keseluruhan yang diperoleh adalah sebesar 0.85.

Tabel 6.3 Akurasi *Gain* ≥ 0

	cukup paham	kurang paham	paham	rata-rata
<i>Recall</i>	0.933333	0.9375	0.909091	0.926641
<i>Precision</i>	0.875	1	0.909091	0.92803
Akurasi	0.85	0.95	0.9	0.9
Akurasi keseluruhan	0.85			

6.3 Uji Coba Nilai *Gain* ≥ 0.1

Uji coba kedua adalah pada saat nilai *Gain* lebih dari atau sama dengan 0.1 maka fitur yang terpilih adalah fitur ke 0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 13, 14, 15, 16 yaitu fitur jumlah anggota keluarga, status tempat tinggal, pendidikan terakhir ibu, pendidikan terakhir ayah, pekerjaan ibu, pekerjaan ayah, pengasuh, dukungan keluarga, ikut ekstrakurikuler, mengulang pelajaran di rumah, posisi baris (bangku) duduk di kelas, mengikuti bimbingan belajar, lama belajar di rumah, peringkat di kelas, jenis bacaan di rumah, lama membaca di rumah. Fitur yang tidak terpilih adalah 7 dan 12 fitur Jarak rumah dan adanya internet di rumah. Tabel 6.4 adalah hasil perhitungan *Confusion Matrix Gain* yang memiliki nilai lebih dari sama dengan 0.1.

Tabel 6.4 *Confusion Matrix* ≥ 0.1

<i>Predicted</i> <i>Actual</i>	cukup paham	kurang paham	paham	Jumlah
cukup paham	4	0	0	4
kurang paham	1	3	1	5
paham	0	1	10	11
Jumlah	5	4	11	20

Dari Tabel 6.4 diperoleh nilai *recall*, *precision* dan akurasi dari setiap kelas dapat dilihat pada Tabel 6.5. Hasil tersebut diperoleh melalui perhitungan seperti pada fungsi 3.1, 3.2 dan 3. yaitu menggunakan nilai TF (*True Positif*), TN (*True Negatif*) FP (*False Positif*) dan (*False Negatif*) rata-rata *recall* sebesar 0.92664133, rata-rata *precision* sebesar 0.92803033 dan rata-rata akurasi adalah sebesar 0.9. Nilai akurasi keseluruhan yang diperoleh adalah sebesar 0.85.

Tabel 6.5 Akurasi *Gain* ≥ 0.1

	cukup paham	kurang paham	Paham	rata-rata
<i>Recall</i>	0.933333	0.9375	0.909091	0.92664133
<i>Precision</i>	0.875	1	0.909091	0.92803033
Akurasi	0.85	0.95	0.9	0.9
Akurasi keseluruhan	0.85			

6.4 Uji Coba Nilai *Gain* ≥ 0.2

Uji coba ketiga adalah pada saat nilai *Gain* lebih dari atau sama dengan 0.2 maka fitur yang terpilih adalah fitur ke 0, 1, 4, 6, 8, 9, 10, 14, 15, 16 yaitu fitur

jumlah anggota keluarga, status tempat tinggal, pekerjaan ibu, pengasuh, dukungan keluarga, ikut ekstrakurikuler, mengulang pelajaran di rumah, mengikuti bimbingan belajar, lama belajar di rumah, jenis bacaan di rumah, lama membaca di rumah. Fitur yang tidak terpilih adalah 2, 3, 5, 7, 11, 12, 13 fitur pendidikan terakhir ibu, pendidikan terakhir ayah, pekerjaan ayah, jarak rumah, posisi duduk dikelas, internet di rumah, dan mengikuti bimbingan belajar. Tabel 6.4 adalah hasil perhitungan *Confusion Matrix Gain* yang memiliki nilai lebih dari sama dengan 0.3.

Tabel 6.6 Confusion Matrix Gain ≥ 0.2

<i>Predicted</i>	cukup paham	kurang paham	paham	jumlah
<i>Actual</i>				
cukup paham	3	0	1	4
kurang paham	0	5	0	5
paham	0	1	10	11
jumlah	3	6	11	20

Dari Tabel 6.6 diperoleh nilai *recall*, *precision* dan akurasi dari setiap kelas dapat dilihat pada Tabel 6.7. Hasil tersebut diperoleh melalui perhitungan seperti pada fungsi 3.1, 3.2 dan 3.3 yaitu menggunakan nilai TF (*True Positif*), TN (*True Negatif*) FP (*False Positif*) dan (*False Negatif*) rata-rata *recall* sebesar 0.891919, rata-rata *precision* sebesar 0.914141 dan rata-rata akurasi adalah sebesar 0.916667. Nilai akurasi keseluruhan yang diperoleh adalah sebesar 0.9.

Tabel 6.7 Akurasi Gain ≥ 0.2

	cukup paham	kurang paham	Paham	rata-rata
<i>Recall</i>	0.933333	0.833333	0.909091	0.891919
<i>Precision</i>	1	0.833333	0.909091	0.914141
Akurasi	0.95	0.9	0.9	0.916667
Akurasi keseluruhan			0.9	

6.5 Uji Coba Nilai *Gain* ≥ 0.3

Uji coba keempat adalah pada saat nilai *Gain* lebih dari atau sama dengan 0.3 maka fitur yang terpilih adalah fitur ke 0, 1, 4, 6, 8, 9, 10, 14, 16 yaitu fitur jumlah anggota keluarga, status tempat tinggal, pekerjaan ibu, pengasuh, dukungan keluarga, ikut ekstrakurikuler, mengulang pelajaran di rumah, lama belajar di rumah, lama membaca di rumah. Fitur yang tidak terpilih adalah 2, 3,

5, 7, 11, 12, 13 dan 15 yaitu fitur pendidikan terakhir ibu, pendidikan terakhir ayah, pekerjaan ayah, jarak rumah, posisi duduk dikelas, internet di rumah, mengikuti bimbingan belajar dan jenis bacaan di rumah. Tabel 6.8 adalah hasil perhitungan *Confusion Matrix Gain* yang memiliki nilai lebih dari sama dengan 0.3.

Tabel 6.8 Confusion Matrix Gain ≥ 0.3

<i>Predicted</i> <i>Actual</i>	cukup paham	kurang paham	paham	jumlah
cukup paham	3	0	1	4
kurang paham	0	5	0	5
paham	1	2	8	11
jumlah	4	7	9	20

Dari Tabel 6.8 diperoleh nilai *recall*, *precision* dan akurasi dari setiap kelas dapat dilihat pada Tabel 6.9. Hasil tersebut diperoleh melalui perhitungan seperti pada fungsi 3.1, 3.2 dan 3.3 yaitu menggunakan nilai TF (*True Positif*), TN (*True Negatif*) FP (*False Positif*) dan (*False Negatif*) rata-rata *recall* sebesar 0.843813333, rata-rata *precision* sebesar 0.942129667 dan rata-rata akurasi adalah sebesar 0.866666667. Nilai akurasi keseluruhan yang diperoleh adalah sebesar 0.8.

Tabel 6.9 Akurasi Gain ≥ 0.3

	cukup paham	kurang paham	paham	rata-rata
<i>Recall</i>	0.866667	0.9375	0.727273	0.843813333
<i>Precision</i>	1	0.9375	0.888889	0.942129667
Akurasi	0.9	0.9	0.8	0.866666667
Akurasi keseluruhan	0.8			

6.6 Uji Coba Nilai *Gain* ≥ 0.4

Uji coba kelima adalah pada saat nilai *Gain* lebih dari atau sama dengan 0.4 maka fitur yang terpilih adalah fitur ke 1, 4, 6, 8, 9, 14, 16 yaitu, status tempat tinggal, pekerjaan ibu, pengasuh, dukungan keluarga, ikut ekstrakurikuler, lama belajar di rumah dan lama membaca di rumah. Fitur yang tidak terpilih adalah 0, 2, 3, 5, 7, 10, 11, 12, 13 dan 15 yaitu fitur jumlah anggota keluarga, pendidikan terakhir ibu, pendidikan terakhir ayah, pekerjaan ayah, jarak rumah, mengulang pelajaran di rumah, posisi duduk dikelas, internet di rumah, mengikuti

bimbingan belajar dan jenis bacaan di rumah. Tabel 6.10 adalah hasil perhitungan *Confusion Matrix Gain* yang memiliki nilai lebih dari sama dengan 0.4.

Tabel 6.10 *Confusion Matrix Gain* ≥ 0.4

<i>Predicted</i> <i>Actual</i>	cukup paham	kurang paham	paham	jumlah
cukup paham	3	0	1	4
kurang paham	0	5	0	5
paham	1	2	8	11
jumlah	4	7	9	20

Dari Tabel 6.10 diperoleh nilai *recall*, *precision* dan akurasi dari setiap kelas dapat dilihat pada Tabel 6.11. Hasil tersebut diperoleh melalui perhitungan seperti pada fungsi 3.1, 3.2 dan 3.3 yaitu menggunakan nilai TF (*True Positif*), TN (*True Negatif*) FP (*False Positif*) dan (*False Negatif*) rata-rata *recall* sebesar 0.843813333, rata-rata *precision* sebesar 0.942129667 dan rata-rata akurasi adalah sebesar 0.866666667. Nilai akurasi keseluruhan yang diperoleh adalah sebesar 0.8.

Tabel 6.11 Akurasi *Gain* ≥ 0.4

	cukup paham	kurang paham	paham	rata-rata
<i>Recall</i>	0.866667	0.9375	0.727273	0.843813333
<i>Precision</i>	1	0.9375	0.888889	0.942129667
Akurasi	0.9	0.9	0.8	0.866666667
Akurasi keseluruhan	0.8			

6.7 Uji Coba Nilai *Gain* ≥ 0.5

Uji coba keenam adalah pada saat nilai *Gain* lebih dari atau sama dengan 0.5 maka fitur yang terpilih adalah fitur ke 1, 4, 6, 8, 9, 14 yaitu, status tempat tinggal, pekerjaan ibu, pengasuh, dukungan keluarga, ikut ekstrakurikuler, lama belajar di rumah dan. Fitur yang tidak terpilih adalah 0, 2, 3, 5, 7, 10, 11, 12, 13, 15 dan 16 yaitu fitur jumlah anggota keluarga, pendidikan terakhir ibu, pendidikan terakhir ayah, pekerjaan ayah, jarak rumah, mengulang pelajaran di rumah, posisi duduk dikelas, internet di rumah, mengikuti bimbingan belajar jenis bacaan di rumah dan lama membaca di rumah. Tabel 6.12 adalah hasil perhitungan *Confusion Matrix Gain* yang memiliki nilai lebih dari sama dengan 0.5.

Tabel 6.12 Confusion Matrix Gain ≥ 0.5

<i>Predicted</i> <i>Actual</i>	cukup paham	kurang paham	paham	jumlah
cukup paham	3	0	1	4
kurang paham	2	0	3	5
paham	3	0	8	11
jumlah	8	0	12	20

Dari Tabel 6.12 diperoleh nilai *recall*, *precision* dan akurasi dari setiap kelas dapat dilihat pada Tabel 6.13. Hasil tersebut diperoleh melalui perhitungan seperti pada fungsi 3.1, 3.2 dan 3.3 yaitu menggunakan nilai TF (*True Positif*), TN (*True Negatif*) FP (*False Positif*) dan (*False Negatif*) rata-rata *recall* sebesar 0.804924333, rata-rata *precision* sebesar 0.777778 dan rata-rata akurasi adalah sebesar 0.7. Nilai akurasi keseluruhan yang diperoleh adalah sebesar 0.55.

Tabel 6.13 Akurasi Gain ≥ 0.5

	cukup paham	kurang paham	paham	rata-rata
<i>Recall</i>	1	0.6875	0.727273	0.804924333
<i>Precision</i>	0.75	0.916667	0.666667	0.777778
Akurasi	0.75	0.7	0.65	0.7
Akurasi keseluruhan	0.55			

6.8 Uji Coba Nilai Gain $\geq 0.6, 0.7, 0.8, 0.9, 1$

Uji coba ketujuh adalah pada saat nilai *Gain* lebih dari atau sama dengan 0.6, 0.7, 0.8, 0.9, dan 1 maka fitur yang terpilih adalah fitur ke 1, 6, 8, 9 yaitu status tempat tinggal, pengasuh, dukungan keluarga, dan ikut ekstrakurikuler. Fitur yang tidak terpilih adalah 0, 2, 3, 4, 5, 7, 10, 11, 12, 13, 14 15 dan 16 yaitu fitur jumlah anggota keluarga, pendidikan terakhir ibu, pendidikan terakhir ayah, pekerjaan ibu pekerjaan ayah, jarak rumah, mengulang pelajaran di rumah, posisi duduk dikelas, internet di rumah, mengikuti bimbingan belajar, lama belajar di rumah jenis bacaan di rumah dan lama membaca di rumah. Tabel 6.14 adalah hasil perhitungan *Confusion Matrix Gain* yang memiliki nilai lebih dari sama dengan 0.6, 0.7, 0.8, 0.9, dan 1.

Tabel 6.14 Confusion Matrix Gain ≥ 0.6 , 0.7, 0.8, 0.9, 1

<i>Predicted</i>	cukup paham	kurang paham	paham	jumlah
<i>Actual</i>				
cukup paham	3	0	1	4
kurang paham	4	0	1	5
paham	6	0	5	11
jumlah	13	0	7	20

Dari Tabel 6.14 diperoleh nilai *recall*, *precision* dan akurasi dari setiap kelas dapat dilihat pada Tabel 6.15. Hasil tersebut diperoleh melalui perhitungan seperti pada fungsi 3.1, 3.2 dan 3.3 yaitu menggunakan nilai TF (*True Positif*), TN (*True Negatif*) FP (*False Positif*) dan (*False Negatif*) rata-rata *recall* sebesar 0.804924333, rata-rata *precision* sebesar 0.777778 dan rata-rata akurasi adalah sebesar 0.7 Nilai akurasi keseluruhan yang diperoleh adalah sebesar 0.4.

Tabel 6.15 Akurasi Gain ≥ 0.6 , 0.7, 0.8, 0.9, 1

	cukup paham	kurang paham	paham	rata-rata
<i>Recall</i>	1	0.375	0.454545	0.609848333
<i>Precision</i>	0.75	0.857143	0.714286	0.773809667
Akurasi	0.75	0.45	0.6	0.6
Akurasi keseluruhan			0.4	

6.9 Uji Coba Nilai *Gain* $\geq 1.1, 1.2, 1.3, 1.4$

Uji coba kedelapan adalah pada saat nilai *Gain* lebih dari atau sama dengan 1.1, 1.2, 1.3, dan 1.4 maka fitur yang terpilih adalah fitur ke 8, 9 yaitu fitur, dukungan keluarga, dan ikut ekstrakurikuler. Fitur yang tidak terpilih adalah 0,1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14 15 dan 16 yaitu fitur jumlah anggota keluarga, status tempat tinggal, pendidikan terakhir ibu, pendidikan terakhir ayah, pekerjaan ibu pekerjaan ayah, pengasuh, jarak rumah, mengulang pelajaran di rumah, posisi duduk dikelas, internet di rumah, mengikuti bimbingan belajar, lama belajar di rumah jenis bacaan di rumah dan lama membaca di rumah. Tabel 6.16 adalah hasil perhitungan *Confusion Matrix Gain* yang memiliki nilai lebih dari sama dengan 1.1, 1.2, 1.3, dan 1.4.

Tabel 6.16 *Confusion Matrix Gain* $\geq 1.1, 1.2, 1.3, 1.4$

<i>Predicted</i> <i>Actual</i>	cukup paham	kurang paham	paham	jumlah
cukup paham	2	0	2	4
kurang paham	0	0	5	5
paham	0	0	11	11
jumlah	2	0	18	20

Dari Tabel 6.16 diperoleh nilai *recall*, *precision* dan akurasi dari setiap kelas dapat dilihat pada Tabel 6.17 Hasil tersebut diperoleh melalui perhitungan seperti pada fungsi 3.1, 3.2 dan 3.3 yaitu menggunakan nilai TF (*True Positif*), TN (*True Negatif*) FP (*False Positif*) dan (*False Negatif*) rata-rata *recall* sebesar 1, rata-rata *precision* sebesar 0.75 dan rata-rata akurasi adalah sebesar 0.766666667. Nilai akurasi keseluruhan yang diperoleh adalah sebesar 0.65.

Tabel 6.17 Akurasi *Gain* $\geq 1.1, 1.2, 1.3, 1.4$

	cukup paham	kurang paham	paham	rata-rata
<i>Recall</i>	1	1	1	1
<i>Precision</i>	0.75	0.888889	0.611111	0.75
Akurasi	0.75	0.9	0.65	0.766666667
Akurasi keseluruhan	0.65			

6.10 Uji Coba Nilai *Gain* ≥ 1.5

Uji coba kesembilan adalah pada saat nilai *Gain* lebih dari atau sama dengan 1.5 maka fitur yang terpilih adalah fitur ke 8, yaitu fitur, dukungan keluarga. Fitur yang tidak terpilih adalah 0,1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14 15 dan 16 yaitu fitur jumlah anggota keluarga, status tempat tinggal, pendidikan terakhir ibu, pendidikan terakhir ayah, pekerjaan ibu pekerjaan ayah, pengasuh, jarak rumah,ikut ekstrakurikuler mengulang pelajaran di rumah, posisi duduk dikelas, internet di rumah, mengikuti bimbingan belajar, lama belajar di rumah jenis bacaan di rumah dan lama membaca di rumah.Tabel 6.18 adalah hasil perhitungan *Confusion Matrix Gain* yang memiliki nilai lebih dari sama dengan 1.5.

Dari Tabel 6.18 diperoleh nilai *recall*, *precision* dan akurasi dari setiap kelas dapat dilihat pada Tabel 6.19. Hasil tersebut diperoleh melalui perhitungan seperti pada fungsi 3.1, 3.2 dan 3.3 yaitu menggunakan nilai TF (*True Positif*), TN (*True Negatif*) FP (*False Positif*) dan (*False Negatif*) rata-rata *recall* sebesar 1,

rata-rata *precision* sebesar 0.7 dan rata-rata akurasi adalah sebesar 0.7. Nilai akurasi keseluruhan yang diperoleh adalah sebesar 0.55.

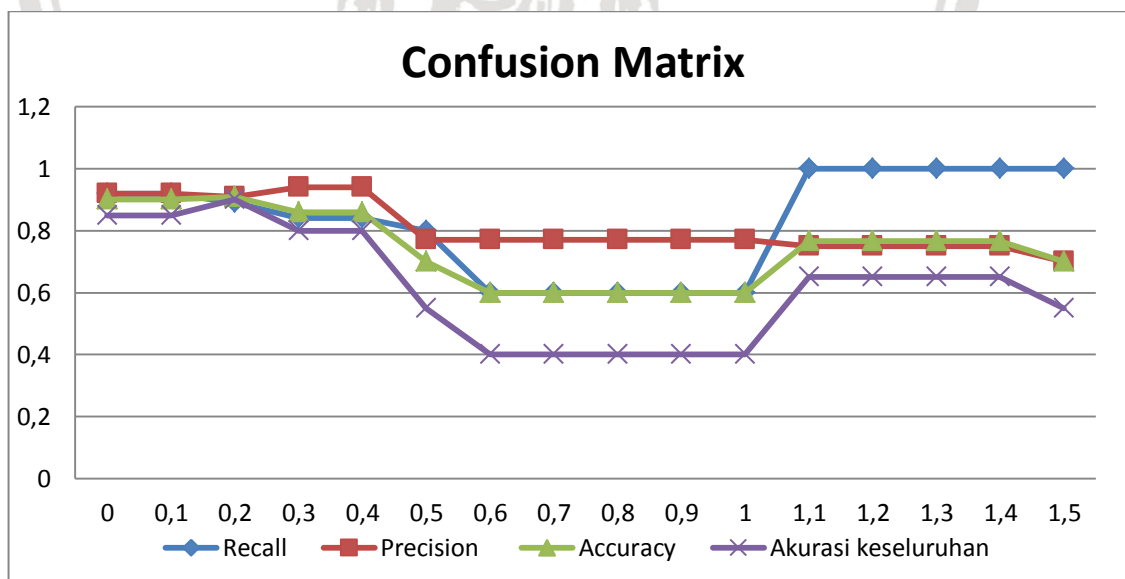
Tabel 6.18 *Confusion Matrix Gain* ≥ 1.5

<i>Predicted</i> <i>Actual</i>	cukup paham	kurang paham	paham	jumlah
cukup paham	0	0	4	4
kurang paham	0	0	0	5
paham	0	0	11	11
jumlah	0	0	20	20

Tabel 6.19 Akurasi *Gain* ≥ 1.5

	cukup paham	kurang paham	paham	rata-rata
<i>Recall</i>	1	1	1	1
<i>Precision</i>	0.75	0.8	0.55	0.7
Akurasi	0.75	0.8	0.55	0.7
Akurasi keseluruhan	0.55			

6.11 Analisis Sistem



Gambar 6.1 Diagram *Confusion Matrix*

Dari pengujian yang sudah dilakukan, didapatkan hasil bahwa akurasi secara keseluruhan tertinggi diperoleh dari nilai *Gain* lebih dari 0.2 yaitu sebesar 0.9. Fitur yang memiliki nilai *Gain* lebih dari 0.2 terpilih fitur ke 0, 1, 4, 6, 8, 9, 10, 14, 15, 16 adalah fitur jumlah anggota keluarga, status tempat tinggal, pekerjaan ibu, pengasuh, dukungan keluarga, ikut ekstrakurikuler, mengulang pelajaran di rumah, lama belajar di rumah, jenis bacaan di rumah, lama membaca di rumah. Fitur yang tidak terpilih adalah fitur pendidikan terakhir Ibu, pendidikan terakhir ayah, pekerjaan ayah, jarak rumah, posisi duduk dikelas, internet di rumah dan mengikuti bimbingan belajar.

Akurasi tertinggi diperoleh dari nilai *gain* sama seperti pada nilai *gain* lebih dari 0.2 yaitu mencapai angka 0.914141. Akurasi ini didapat dengan menghitung rata-rata dari setiap kelas.

Recall tertinggi diperoleh dari nilai *gain* yang bernilai 1.1 hingga 1.5 yaitu mencapai angka 1 dengan fitur yang terpilih adalah fitur ke 8 dan fitur 9 yaitu fitur dukungan keluarga dan ikut ekstrakurikuler. Seperti kita ketahui *Recall* merupakan kualitas seberapa lengkap hasil relevan yang ditampilkan oleh sistem pencarian maka dapat dikatakan bahwa sistem ini memberikan hasil yang relevan.

Precision tertinggi diperoleh oleh nilai *Gain* yang lebih dari atau sama dengan 0.4 yaitu mencapai 0.942129667 dengan fitur yang terpilih adalah fitur 1, 4, 6, 8, 9, 14, 16 pekerjaan ibu, pengasuh, dukungan keluarga, ikut ekstrakurikuler, lama belajar di rumah, lama membaca di rumah. *Precision* merupakan pengukuran kualitas seberapa bergunakah sistem pencarian tersebut maka dapat dikatakan bahwa sistem ini 94% berguna.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan rumusan masalah yang telah dirumuskan, kesimpulan yang diperoleh oleh penulis adalah sebagai berikut:

1. Fitur yang memberikan pengaruh terhadap hasil klasifikasi adalah fitur yang memiliki nilai *Gain* lebih dari atau sama dengan 0.2 diantaranya yaitu fitur index ke 0, 1, 4, 6, 8, 9, 10, 14, 15, 16 antarlain fitur jumlah anggota keluarga, status tempat tinggal, pekerjaan ibu, pengasuh, dukungan keluarga, ikut ekstrakurikuler, mengulang pelajaran di rumah, mengikuti bimbingan belajar, lama belajar di rumah, jenis bacaan di rumah dan lama membaca di rumah. Fitur yang tidak terpilih adalah fitur dengan index ke 2, 3, 5, 7, 11, 12, 13 fitur pendidikan terakhir ibu, pendidikan terakhir ayah, pekerjaan ayah, jarak rumah, posisi duduk dikelas, internet di rumah, dan mengikuti bimbingan belajar.
2. Akurasi keseluruhan tertinggi pada sistem ini adalah sebesar 0.9 yaitu dengan fitur yang terpilih adalah fitur yang memiliki nilai *Gain* lebih dari atau sama dengan 0.2. Rata-rata akurasi dari setiap kelas adalah sebesar 0.914 fitur dengan nilai *Gain* yang sama yaitu 0.2. Rata-rata *recall* tertinggi adalah sebesar 1 yaitu fitur dengan nilai *Gain* yang bernilai 1.1 hingga 1.5. Sedangkan untuk *precision* yang tertinggi adalah sebesar 0.942 yaitu fitur yang memiliki nilai *Gain* lebih dari atau sama dengan 0.4.

7.2 Saran

Saran dari penulis untuk penelitian selanjutnya yaitu sebagai berikut:

1. Menggunakan metode klasifikasi yang lain selain klasifikasi *naive bayes* agar dapat menjadi perbandingan metode klasifikasi apa yang memiliki kualitas dan hasil yang terbaik
2. Menggunakan metode seleksi fitur yang lain selain *information Gain* agar dapat menjadi perbandingan metode seleksi fitur apa yang memiliki kualitas dan hasil yang terbaik
3. Tambahkan skenario pengujian dengan menguji validitas data menggunakan metode *k-fold cross validation* agar data latih dan data uji dapat divalidasi.
4. Gunakan studi kasus siswa selain SMA *Smart School* Brawijaya untuk mendapatkan data yang lebih bervariasi. Dapat juga dilakukan dengan menganalisis siswa SMP atau SD.
5. Analisis pada pemahaman selain mata pelajaran Bahasa Indonesia dengan mengubah fitur-fitur dari penelitian ini dengan fitur yang memiliki keterkaitan dengan mata pelajaran yang akan diteliti

DAFTAR PUSTAKA

- Ali Khan, S., 2005. *Filsafat Pendidikan Al-Ghazali*. Bandung: Pustaka Setia. Anon., n.d. s.l.:s.n.
- Bafadal, I., 2009. *Menekemen Peningkatan Mutu Sekolah Dasar: Dari Sentralisasi Menuju Desentralisasi*. 1st ed. Jakarta: Bumi Aksara.
- Bhardwaj, B. . K. & Pal, S., 2011. Data Mining: A prediction for performance improvement using classification. (*IJCSIS*) *International Journal of Computer Science and Information Security*, 9(4).
- Choliq, A., 2011. *Pengantar Manajemen*. Semarang: Rafi Sarana Perkasa.
- Fayyad, U., Piatetsky-Shapiro, G. & Smyth , . P., 1996. From Data Mining to Knowledge Discovery in Database. *American Association for Artificial Intelligence*, 17(3), pp. 37-54.
- Hamakonda, T. P. & Tairas, J., 1995. *Pengantar Klasifikasi Persepuluhan Dewey*. 1st ed. Jakarta: Gunung Mulya.
- Handoko, T. H., 1999. *Manajemen*. Yogyakarta : BPFE Yogyakarta .
- Hasibuan, M. S., 2001. *Manajemen Sumber Daya Manusia*. Jakarta: PT. Bumikarsa.
- Herdianto, 2013. *Prediksi Kerusakan Motor Induksi Menggunakan Metode Jaringan Saraf Tiruan Backpropagation*, Medan: Universitas Sumatera Utara .
- Kusrini & Luthfi, E. T., 2009,. *Algoritma Data Mining*. 1st ed. Yogyakarta: Andi Offset.
- Larose, D. T., 2005. *Discovering Knowledge in Data An Introduction to Data Mining*. 1st ed. Canada: A John Wiley & Sons, Inc..
- Maimon, O. & Rokach, L., 2010. *Data Mining and Knowledge Discovery Handbook*. 2nd ed. Eropa: Springer.
- Mhetre, V. & Nagar, M., 2017. *Classification based data mining algorithms to predict slow, average and fast learners in educational system using Weka*. Mumbai, Proceedings of the IEEE 2017 International Conference on Computing Methodologies and Communication.
- Mitchell, T. M., 1997. *Machine Learning*.. United States of America: McGraw – Hill.
- Moeljono, S., 1989. *Bahasa Indonesia dan Problematikanya*. 1st ed. Madiun: Widya Mandala.
- Muhaimin, 2005. *Pengembangan Kurikulum PAI Islam di Sekolah, Madarasah, dan Perguruan Tinggi*. Jakarta: Raja Grafindo Persada.

- Nugroho, A. & Subanar., 2013. Klasifikasi Naive Bayes untuk Prediksi Kelahiran pada Data Ibu Hamil. *Berkala MIPA*, pp. 297-308.
- Parack, S., Zahid, Z. & Merchan, F., 2012. *Application of Data Mining in Educational Databases for Predicting Academic Trends and Patterns*, Kerala: IEEE Xplore.
- Patil, T. R. & Sherekar, M. S., 2013. Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification. *International Journal of Computer Science and Applications*, 6(2), pp. 256-261.
- Purwanto, M., 1994. *Prinsip-prinsip Dan Teknik Evaluasi Pengajaran Pendidikan*. 1st ed. Bandung: Rosda Karya.
- R., P., Tang, L. & Liu, H., 2008. *Cross Validation*, USA: Arizona State University.
- Rahman, F., 2016. *Penerapan Data Mining Metode Naive Bayes Untuk Prediksi Hasil Belajar Siswa Sekolah Menengah Pertama (Smp)*, Banjarmasin: Fakultas Teknologi Informasi Universitas Islam Kalimantan (Uniska).
- Rozzaqi, A. R., 2015. *Naive Bayes dan Filtering Feature Selection Information Gain untuk Prediksi Ketepatan Kelulusan Mahasiswa*, Semarang: Universitas PGRI Semarang.
- Sadili, S., 2006. *Manajemen Sumber Daya Manusia*. Bandung: Pustaka Setia.
- Saleh, A., 2015. Implementasi Metode Klasifikasi Naive Bayes Dalam Memprediksi Besarnya Penggunaan Listrik Rumah Tangga. *Citec Journal*, 2(3), pp. 2354-5771.
- Shahiria, A. M., Husaini, W. & Rashid, N. A., 2015. *A Review on Predicting Student's Performance using Data Mining Techniques*. Penang, Elsevier.
- Sulistyo-Basuki, 1993. *Pengantar Ilmu Perpustakaan*. 1st ed. Jakarta: Gramedia Pustaka Utama.
- Trisedya, B. & Jais, H., 2009. *Klasifikasi Dokumen Menggunakan Algoritme Naive Bayes dengan Parent Category*, Depok: Fasilkom Universitas Indonesia.
- Turban E., A. J., 2003. *Decision Support Systems and Intelligent Systems*. 1st ed. Yogyakarta: Andi Offset.
- Uliyandri, M., 2014. *Analisis Tingkat Pemahaman Siswa Kelas Xii Ipa Sma Negeri Kota Bengkulu Untuk Mata Plajaran Kimia*, Bengkulu: Universitas Bengkulu.
- Witten, H. I., Frank, E. & Hall, M. A., 2011. *Data Mining Practical Machine Learning Tools and Techniques*, USA: Elsevier Inc.
- Supianto, A. A., Hayashi, Y. & Hirashima, T., 2017. *Designing Scaffolding System*

in a Problem-Posing Learning Environment. Hiroshima, Japan, International Conference on Science in Information Technology (ICSITech).

Supianto, A. A., Hayashi, Y. & Hirashima, T., 2017. Model-based analysis of thinking in problem posing as sentence integration focused on violation of the constraints. *Research and Practice in Technology Enhanced Learning*, 1(12), p. 12.

