

**APLIKASI PENGENALAN JENIS-JENIS IRAMA *QIRO'AH*
MENGUNAKAN METODE *NAÏVE BAYES CLASSIFIER***

SKRIPSI

Oleh:
TOTO RICKY FERNANDO
NIM. 14650012



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2020**

**APLIKASI PENGENALAN JENIS-JENIS IRAMA QIRO'AH
MENGUNAKAN METODE NAÏVE BAYES CLASSIFIER**

SKRIPSI

**Diajukan Kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN)
Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:
TOTO RICKY FERNANDO
NIM. 14650012**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK BRAHIM
MALANG
2020**

LEMBAR PERSETUJUAN

**APLIKASI PENGENALAN JENIS-JENIS IRAMA *QIRO'AH*
MENGUNAKAN METODE *NAÏVE BAYES CLASSIFIER***

SKRIPSI

Oleh:
TOTO RICKY FERNANDO
NIM. 14650012

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal: Juli 2020

Dosen Pembimbing I

Dosen Pembimbing II

Syahiduz Zaman, M.Kom
NIP. 19700502 200501 1 005

Irwan Budi Santoso, M.Kom
NIP. 19770103 201101 1 004

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

HALAMAN PENGESAHAN

APLIKASI PENGENALAN JENIS-JENIS IRAMA *QIRO'AH* MENGUNAKAN METODE *NAÏVE BAYES CLASSIFIER*

SKRIPSI

Oleh:
TOTO RICKY FERNANDO
NIM. 14650012

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: Juli 2020

Susunan Dewan Penguji	Tanda Tangan
Penguji Utama : <u>M. Ainul Yaqin, M.Kom</u> NIP. 19761013 200604 1 004	()
Ketua Penguji : <u>Fatchurrochman, M.Kom</u> NIP. 19700731 200501 1 002	()
Sekretaris Penguji : <u>Syahiduz Zaman, M.Kom</u> NIP. 19700502 200501 1 005	()
Anggota Penguji : <u>Irwan Budi Santoso, M.Kom</u> NIP. 19770103 201101 1 004	()

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

HALAMAN PERSEMBAHAN

Bismillahirrahmanirrahim

Segala puji bagi Allah atas ridho-Nya dan juga dukungan serta doa dari orang-orang tercinta akhirnya skripsi ini dapat terselesaikan dengan baik. Oleh karena itu kupersembahkan karya sederhana ini kepada:

Ayah dan Ibu tersayang yang selalu memberikan motivasi, pengorbanan dan doa yang tiada henti untuk kesuksesan saya.

Bapak Syahiduz Zaman, M.Kom dan Bapak Irwan Budi Santoso, M.Kom selaku dosen pembimbing yang selama ini tulus, sabar serta ikhlas meluangkan waktunya untuk memberikan bimbingan, selain itu juga untuk seluruh dosen Teknik Informatika yang selama ini telah mendidik dan menyalurkan segala ilmu informatika dan juga pengalaman yang sangat berarti untuk saya. Bapak dan Ibu dosenku jasmu akan selalu terpatri dihati.

Teman-teman Biner (TI 2014) Irfan, Sukran, Rico, Wanga, Ipunk, Anwar, Fathur, Filza, Galang, Adam, Nanda dst, terima kasih atas semangat, dukungan dan bantuan serta kekompakan selama ini, terima kasih atas canda tawa dan kenangan manis yang telah terukir selama ini. Sekaligus teman di saat jurusan telah sepi dari angkatan 2014. Semoga suatu hari ada moment bertemu berkumpul bersama lagi.

Seluruh teman-temanku yang senantiasa selalu memotivasi dan menjadi tempat untuk saling belajar.

Serta seluruh pihak yang tidak dapat disebutkan satu persatu, terima kasih.

KATA PENGANTAR

Assalamualaikum Wr. Wb.

Segala puji bagi Allah SWT Tuhan semesta alam, karena atas segala rahmat dan karunia-Nya sehingga penulis mampu menyelesaikan skripsi ini dengan judul “Aplikasi Pengenalan Jenis-Jenis Irama *Qiro’ah* Menggunakan Metode *Naïve Bayes Classifier*” sebagai salah satu syarat meraih gelar Strata-1 program studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.

Shalawat serta salam selalu tucurahkan kepada tauladan terbaik Nabi Muhammad SAW yang telah membimbing umatnya dari zaman kebodohan menuju Islam yang rahmatan lil alamin.

Selanjutnya penulis mengucapkan terima kasih karena dalam penyelesaian skripsi ini tidak lepas dari bantuan, bimbingan serta dukungan dari beberapa pihak. Oleh karnanya, dengan kerendahan hati penulis mengucapkan terima kasih kepada:

1. Prof. DR. H. Abdul. Haris, M.Ag, selaku rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang beserta seluruh staf. Bakti Bapak dan Ibu sekalian yang menaungi segala kegiatan di kampus Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.

2. Dr. Sri Harini, M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang beserta seluruh staf.
3. Bapak Dr. Cahyo Crysdian, selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
4. Bapak Syahiduz Zaman, M.Kom selaku dosen pembimbing I yang telah bersedia meluangkan waktu untuk membimbing, memotivasi untuk terus berkembang dan memberi arahan kepada penulis dalam pengerjaan skripsi ini hingga akhir.
5. Bapak Irwan Budi Santoso, M.Kom selaku dosen pembimbing II yang juga senantiasa memberi masukan dan nasihat serta petunjuk dalam penyusunan skripsi ini.
6. Bapak Fatchurrochman, M.Kom selaku dosen wali yang telah dan selalu memberi arahan dan bimbingan hingga saat ini.
7. Ayah, Ibu dan Adik serta keluarga besar tercinta yang selalu memberi dorongan dan doa yang senantiasa mengiringi setiap langkah penulis.
8. Seluruh Dosen Teknik Informatika yang telah memberikan keilmuan serta pengalaman yang berarti kepada penulis selama ini.
9. Teman-teman yang telah memotivasi dan membantu banyak hal selama ini.
10. Seluruh teman-teman Teknik Informatika Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang yang telah banyak berbagi ilmu,

pengalaman dan menjadi inspirasi untuk terus semangat belajar dan berdiskusi.

11. Teman-teman seperjuangan Teknik Informatika Biner 2014 yang telah berjuang bersama dan saling mendukung selama ini.

12. Para peneliti yang telah melakukan penelitian tentang klasifikasi pengolahan suara menggunakan metode *naïve bayes classiffier* yang menjadi acuan penulis dalam pembuatan skripsi ini. Serta semua pihak yang telah membantu yang tidak bisa disebutkan satu persatu, mengucapkan terima kasih banyak.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna dikarenakan terbatasnya pengalaman dan pengetahuan yang dimiliki penulis. Oleh karena itu, penulis mengharapkan segala bentuk saran dan masukan serta kritik yang membangun dari berbagai pihak. Terlepas dari berbagai kekurangan tersebut, semoga skripsi ini dapat memberikan manfaat bagi pembaca dan mendorong peneliti selanjutnya dalam menyempurnakannya. Aminn.

Wassalamualaikum Wr. Wb.

Malang, Juli 2020

Penulis

MOTTO

“MEMANG BENAR!!! KEBAHAGIAAN YANG
SESUNGGUHNYA ITU ADA DI DALAM HATI,
BUKAN DI MANA KITA DUDUK”



DAFTAR ISI

HALAMAN JUDUL.....	ii
LEMBAR PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iv
PERNYATAAN KEASLIAN TULISAN	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR	vii
MOTTO	x
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiv
DAFTAR TABEL	xvi
ABSTRAK.....	xvii
ABSTRACT.....	xviii
ملخص.....	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Tujuan Penelitian.....	6
1.4 Batasan Penelitian	6
1.5 Manfaat Penelitian.....	7
1.6 Sistematika Penulisan.....	8
BAB II KAJIAN PUSTAKA	10
2.1 Definisi-Definisi	10
2.1.1 Seni Membaca Al-Quran (qiro'ah)	10

2.1.2	Persiapan Seorang Menjadi Qori' dan Qori'ah.....	10
2.1.3	Jenis-Jenis Irama qiro'ah.....	12
2.1.4	Suara.....	13
2.1.5	Representasi Suara	14
2.1.6	Amplitudo	16
2.1.7	Similarity.....	17
2.2	Klasifikasi Naïve Bayes Classifier	17
2.2.1	Teorema Bayes.....	18
2.2.2	Penerapan Naïve Bayes Classifier untuk Klasifikasi	18
2.2.3	Kinerja Klasifikasi Metode Naïve Bayes Classifier.....	20
2.3	NetBeans IDE.....	21
2.4	Penelitian Terkait	22
BAB III METODE PENELITIAN.....		26
3.1	Prosedur Penelitian.....	26
3.2	Deskripsi Data Objek	31
3.3	Desain Penelitian.....	32
3.3.1	Desain Proses	32
3.3.2	Desain Sistem.....	34
3.4	Proses Klasifikasi	35
3.4.1	Proses Pengambilan Data Training	35
3.4.2	Data Rekaman qiro'ah.....	36
3.4.3	Flowchart Algoritma Naïve Bayes Classifier	37
3.4.4	Perhitungan Naïve Bayes Classifier.....	39
3.4.5	Proses Pengecekan Similarity	44
3.5	Perhitungan Akurasi	45

3.6	Desain Interface.....	46
BAB IV UJI COBA DAN PEMBAHASAN		47
4.1	Peralatan yang digunakan.....	47
4.1.1	Hardware	47
4.1.2	Software	48
4.2	User Interface	48
4.3	Pembahasan	53
4.4	Hasil dan Pengujian.....	64
4.5	Integrasi Penelitian terhadap Al-Quran dan Hadits.....	74
BAB V PENUTUP.....		78
5.1	Kesimpulan.....	78
5.2	Saran.....	79
DAFTAR PUSTAKA		81
LAMPIRAN-LAMPIRAN.....		84

DAFTAR GAMBAR

Gambar 2.1 Proses Sampling	15
Gambar 2.2 Visualisasi Gelombang Amplitudo	16
Gambar 2.3 Gelombang Amplitudo Berkaitan dengan Waktu	16
Gambar 3.1 Prosedur Penelitian.....	27
Gambar 3.2 Desain Proses	33
Gambar 3.3 Uce Case Diagram Fitur Aplikasi	33
Gambar 3.4 Flowchart Sistem.....	34
Gambar 3.5 Alur Proses Pengambilan Data Training.....	35
Gambar 3.6 Flowchart Algoritma Naïve Bayes Classifier	38
Gambar 3.7 Diagram Blok Perhitungan Naïve Bayes Classifier	40
Gambar 3.8 Desain Interface Aplikasi Pengenalan Jenis Irama qiro'ah.....	46
Gambar 4.1 Running Aplikasi dan Tampilan Utama.....	48
Gambar 4.2 Tampilan untuk Mengambil Audio	50
Gambar 4.3 Tampilan Aplikasi setelah Memilih Audio.wav	51
Gambar 4.4 Tampilan Probabilitas Data Testing.....	52
Gambar 4.5 Tampilan Hasil Akhir Penentuan Jenis-Jenis Irama qiro'ah.....	52
Gambar 4.6 Tampilan Nilai-Nilai pada Data Training	53
Gambar 4.7 Running Nilai AmplitudoMin, AmplitudoMax, TurunAvg, dan NaikAvg Berdasarkan Jenis Irama qiro'ah	55
Gambar 4.8 Output Perhitungan Rata-Rata dari Setiap Jenis Irama qiro'ah	56
Gambar 4.9 Output Perhitungan Distribusi Normal Tiap Jenis Irama.....	62

Gambar 4.10 Output Proses Sebelum dan Sesudah Similarity	64
Gambar 4.11 Diagram Blok Pengujian Data Testing.....	66
Gambar 4.12 Hasil Pengujian Data Testing contoh File Ba011431.wav.....	68



DAFTAR TABEL

Tabel 3. 1 Jumlah Keseluruhan Data Training Rekaman	37
Tabel 3.2 Jumlah Keseluruhan Data Testing Rekaman	37
Tabel 3.3 Data Training Rekaman dan Parameter Perhitungan Naïve Bayes.....	40
Tabel 3.4 Jumlah 70 Data Training.....	41
Tabel 3.5 Contoh Pengambilan Data Testing untuk Perhitungan Naïve Bayes....	41
Tabel 3.6 Perhitungan Rata-Rata dari Setiap Atribut Jenis Irama	42
Tabel 3.7 Perhitungan Standart Deviasi Setiap Atribut Jenis Irama.....	42
Tabel 3.8 Probabilitas Naïve Bayes Setiap Jenis Irama.....	43
Tabel 3.9 Perhitungan Distribusi Normat dari Setiap Atribut Jenis Irama	44
Tabel 3.10 Model Confusion Matric.....	45
Tabel 4.1 Pendataan Data Training Rekaman Qiro'ah Berjumlah 210 Sampel ...	53
Tabel 4.2 Hasil Pengujian Kinerja System Data Training	64
Tabel 4.3 Hasil Pengujian Kinerja System Data Testing.....	68
Tabel 4.4 Confusion Matrix	71
Tabel 4.5 Hasil Klasifikasi Akurasi Perhitungan Metode Confusion Matrix	71
Tabel 4.6 Hasil Nilai Akurasi, Presisi, dan Recall Setiap Jenis Irama qiro'ah	73
Tabel 4.7 Nilai Akhir Akurasi, Presisi, dan Recall	74

ABSTRAK

Fernando, Toto Ricky. 2020. Aplikasi Pengenalan Jenis-Jenis Irama *Qiro'ah* Menggunakan Metode *Naïve Bayes Classifier*. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing (I) Syahiduz Zaman, M.Kom. (II) Irwan Budi Santoso, M.Kom

Kata Kunci: Klasifikasi, *Naïve Bayes Classifier*, Irama *Qiro'ah*

Al-Quran merupakan kitab suci bagi umat Islam. Umat Islam mempercayai bahwa Al-Quran merupakan wahyu terakhir dari Allah SWT yang diturunkan bagi manusia. Selama ini pencapaian seseorang dalam hal membaca Al-Quran khususnya dalam seni membaca Al-Quran atau yang disebut dengan *qiro'ah* masih minim. Hal ini disebabkan adanya masalah dalam hal pemahaman seseorang yang ditentukan oleh berbagai faktor yang mempengaruhinya. Salah satunya adalah faktor dalam diri seseorang yaitu belum bisa membedakan suatu jenis irama dalam seni membaca Al-Quran. Seni membaca Al-Quran merupakan bacaan yang harus didasarkan sesuai dengan tajwid yang sudah ditetapkan, dan tidak lepas dari irama dan lagu. Sedangkan, sebuah sistem yang mampu mengklasifikasikan berdasarkan jenis irama *qiro'ah* belum banyak dilakukan secara intensif. Penelitian ini menghasilkan sebuah model klasifikasi jenis irama *qiro'ah* sebagai salah satu alternatif penyelesaian masalah pengenalan jenis-jenis irama *qiro'ah* berdasarkan suara *qori* dengan data rekaman suara jenis irama *qiro'ah* yang digunakan adalah Bayyati, Hijaz, Nahawan, Rost, Jiharkah, Shoba, dan Sikah. Dikarenakan jenis irama tersebut yang umum dibacakan oleh pecinta *qori*. Metode penelitian yang diterapkan dalam sistem ini adalah *Naïve Bayes Classifier* yang digunakan sebagai klasifikasi untuk mendapatkan nilai yang identik cocok berdasarkan kemiripan suara. Hasil penelitian ini berupa: (1). Aplikasi pengenalan jenis-jenis irama *qiro'ah* menggunakan metode *naïve bayes classifier* memanfaatkan aplikasi *Java NetBeans IDE*. (2). Pengujian dengan nilai akurasi persentase data *training* sebesar 85,7% dan data *testing* menggunakan *confusion matrix validation* didapat sebesar akurasi 57%, presisi 57%, dan *recall* 72%. (3). Berdasarkan total kinerja yang telah didapatkan, maka dapat ditarik kesimpulan dengan implementasi metode *naïve bayes classifier* dapat digunakan mengklasifikasikan pengenalan jenis irama *qiro'ah* Bayyati, Hijaz, Nahawan, Rost, Jiharkah, Shoba, dan Sikah berdasarkan suara *qori*.

ABSTRACT

Fernando, Toto Ricky. 2020. **Application of Introduction to Types of Qiro'ah Rhythms Using the Naïve Bayes Classifier Method.** Thesis. Informatics Engineering Department of the Faculty of Science and Technology of the State Islamic University of Maulana Malik Ibrahim Malang.

Supervisor (I) **Syahiduz Zaman, M.Kom.** (II) **Irwan Budi Santoso, M.Kom**

Keywords: Classification, Naïve Bayes Classifier, Rhythm of Qiro'ah

Al-Quran is a holy book for Islam. Muslims believe that the Al-Qur'an is the final revelation from Allah, revealed to humans. During this time someone's achievement in terms of reading the Al-Qur'an especially in the art of reading the Al-Qur'an or the so-called qiro'ah is still minimal. This is due to a problem in terms of one's understanding which is determined by various factors that influence it. One of them is a factor in a person that is not yet able to distinguish a type of rhythm in the art of reading the Al-Qur'an. The art of reading the Al-Qur'an is a reading that must be based in accordance with the recitation set, and can not be separated from the beat and song. Meanwhile, a system that is able to classify based on the type of qiro'ah rhythm has not been done intensively. This research resulted in a classification model of qiro'ah rhythm as an alternative to solving the problem of introducing the types of qiro'ah rhythm based on qori sound with recorded sound data of qiro'ah rhythm types used were Bayyati, Hijaz, Nahawan, Rost, Jiharkah, Shoba, and Sikah. Because these types of rhythms are commonly read by qori lovers. The research methodology applied in the system is the *Naïve Bayes Classifier* used as a classification to get matching identical values based on sound similarity. The results of this study are: (1). Application of introduction to the types of rhythm of qiro'ah using the *naïve bayes classifier* method utilizing the *Java NetBeans IDE* application. (2). Testing with an accuracy value of training data percentage of 85.7% and testing data using *confusion matrix validation* obtained an accuracy of 57%, 57% precision, and 72% *recall*. (3). Based on the total performance that has been obtained, it can be concluded with the implementation of the *naïve bayes classifier* method can be used to classify the introduction of the types of rhythm of the Bayyati qiro'ah, Hijaz, Nahawan, Rost, Jiharkah, Shoba, and Sikah based on the qori voice.

ملخص

فرناندو، توتو ريكي. ٢٠٢٠. تطبيق مقدمة على أنواع إيقاعات القروعة باستخدام الطرق Naïve Bayes Classifier. أطروحة. قسم المعلوماتية ، كلية العلوم والتكنولوجيا ، جامعة مولانا مالك إبراهيم الإسلامية الحكومية في مالانغ.

مرشد (١) شهيدوزجمان، الماجستير. (٢) إروان بودي سانتوس، الماجستير.

الكلمات الدالة : تصنيف، Naïve Bayes Classifier، إيقاع القروعة.

القرآن هو كتاب مقدس للإسلام. يعتقد المسلمون أن القرآن هو الوحي النهائي من الله ، أنزل على البشر. خلال هذا الوقت ، لا يزال إنجاز شخص ما من حيث قراءة القرآن وخاصة في فن قراءة القرآن أو ما يسمى القروعة ضئيلاً. ويرجع ذلك إلى مشكلة من حيث فهم المرء والتي تحددها العوامل المختلفة التي تؤثر عليه. أحدها عامل في شخص لا يستطيع بعد تمييز نوع من الإيقاع في فن قراءة القرآن. فن قراءة القرآن هو قراءة يجب أن تستند إلى مجموعة التلاوة ، ولا يمكن فصلها عن الإيقاع والأغنية. وفي الوقت نفسه ، لم يتم إجراء نظام قادر على التصنيف بناءً على نوع إيقاع القرعة بشكل مكثف. أسفر هذا البحث عن نموذج تصنيف لإيقاع القروع كبديل لحل مشكلة إدخال أنواع إيقاعات القروع بناءً على صوت القرع مع البيانات الصوتية المسجلة لأنواع إيقاعات القروع المستخدمة: البياتي ، الحجاز ، نهاوان ، روست ، جهارقة ، شوبا والزواج. لأن هذه الأنواع من الإيقاعات يقرأها عادةً عشاق القر. منهجية البحث المطبقة في النظام Naïve Bayes Classifier تستخدم لتصنيف للحصول على قيم متطابقة متطابقة بناءً على تشابه الصوت. تتضمن نتائج هذه الدراسة ما يلي: (١) تطبيق مقدمة على أنواع إيقاعات القروعة باستخدام الطريقة Naïve Bayes Classifier الاستفادة من التطبيق Java NetBeans IDE. (٢) حصل الاختبار باستخدام قيمة دقة لنسبة بيانات التدريب بنسبة ٨٥,٧٪ وبيانات الاختبار باستخدام التحقق من مصفوفة الخلط على دقة ٥٧٪ ودقة ٥٧٪ واستدعاء ٧٢٪. (٣) بناءً على الأداء الكلي الذي تم الحصول عليه ، يمكن اختتامه بتطبيق الطريقة Naïve Bayes Classifier يمكن استخدامه لتصنيف إدخال أنواع إيقاعات القروعة البياتي ، والحجاز ، ونهاوان ، وروست ، وجيهاركا ، وشوبا ، وسيكة بناءً على صوت القرع.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Al-Quran merupakan kitab suci bagi umat Islam. Umat Islam mempercayai bahwa Al-Quran merupakan wahyu terakhir dari Allah SWT yang diturunkan bagi manusia. Percaya kepada Al-Quran termasuk bagian dari rukun iman yang telah disampaikan kepada nabi Muhammad SAW melalui perantara malaikat Jibril. Kewajiban bagi setiap umat muslim yaitu selalu membaca Al-Quran. Membaca Al-Quran pada dasarnya sangat penting bagi umat Islam. Oleh karena itu membaca Al-Quran hendaknya dilatih sejak usia dini. Pada zaman sekarang terbukti kalangan anak-anak usia dini sudah mulai banyak yang bisa membaca Al-Quran, bahkan tidak sedikit yang sudah menghafal Al-Quran.

Membaca Al-Quran pada dasarnya mempunyai tata aturan tertentu yang telah ditetapkan. Allah SWT telah mensyariatkan kepada orang yang membaca Al-Quran untuk mengetahui dan menetapkan tata cara membaca Al-Quran. Allah SWT memerintahkan nabi Muhammad SAW untuk membaca Al-Quran sebagaimana firman-Nya yang berbunyi,

... وَرَتِّلِ الْقُرْآنَ تَرْتِيلًا (٤)

Artinya: ...dan bacalah Al Qur'an itu perlahan-lahan" (Q.S Al-Muzzammil : 4).

Ayat di atas mengandung arti seseorang yang sedang membaca Al-Quran diharuskan dengan *khusu'* atau *tabaddur* (memperhatikan isinya) dan membacanya secara terus-menerus walaupun hanya satu ayat, sesuai dengan aturan tata cara membacanya yang sudah diajarkan Nabi Muhammad SAW.

Selama ini pencapaian seseorang dalam hal membaca Al-Quran khususnya dalam seni membaca Al-Quran atau yang disebut dengan *qiro'ah* masih minim. Hal ini disebabkan adanya masalah dalam hal pemahaman seseorang yang ditentukan oleh berbagai faktor yang mempengaruhinya. Salah satunya adalah faktor dalam diri seseorang yaitu belum bisa membedakan suatu jenis irama dalam seni membaca Al-Quran. Seni membaca Al-Quran merupakan bacaan yang harus didasarkan sesuai dengan tajwid yang sudah ditetapkan, dan tidak lepas dari irama dan lagu. Di dalam melagukan ayat Al-Quran akan terasa indah apabila dibacakan dengan irama. Menurut para ahli *Qurro* di Indonesia, mengiramakan bacaan Al-Quran membagi tujuh macam bagian, yaitu: *Bayati, Shoba, Hijaz, Nahawand, Rost, Jiharkah, dan Sikah*. *qiro'ah* pada saat ini sudah menjadi populer di Indonesia bahkan kerap kali diadakan perlombaan *qiro'ah* yang biasa disebut MTQ (*Musabaqoh Tilawatil Quran*).

Banyak teknik vokal yang khusus untuk mencapai hasil yang diinginkan sehingga diperlukan latihan khusus dan disiplin yang kuat. Di samping itu, bakat dan minat juga sangat berpengaruh terhadap kualitas keberhasilan pengolahan teknik vokal dalam seni membaca Al-Quran. *qiro'ah* memiliki tiga komponen dasar yaitu: suara, pernafasan, dan bacaan Al-Quran dengan baik dan lancar. Masalah pengenalan suara ini menarik dan penting karena permasalahan dalam identifikasi pembicara memiliki variasi pengucapan setiap suku kata berbeda. Karakteristik yang sangat spesifik suatu sinyal suara diakibatkan oleh adanya perbedaan struktur fisiologi dan aspek-aspek pembawaan dalam masing-masing individu (Safriadi & Risawandi, 2014). Sehingga, dalam mengiramakan ayat Al-Quran selain suara yang baik, nafas juga sangat dibutuhkan. Seorang *qori'* atau

qori'ah harus memiliki kemampuan pernafasan yang lebih dari pada orang biasa. Seorang *qori'* atau *qori'ah* yang memiliki nafas yang panjang akan terjaga kesempurnaan ketika membacanya dan terhindar dari *waqaf* (berhenti) yang bukan pada tempatnya atau terhindar dari akhiran bacaan yang kurang harmonis karena kehabisan nafas.

Berdasarkan ajaran agama Islam, bahwa membaca Al-Quran dengan seni baca keindahan suara adalah dapat dikategorikan sebagai ibadah atau dakwah sebab lagu yang indah sesuai dengan kaidah-kaidah seni bacaan Al-Quran dapat mengantarkan suatu bacaan yang lebih meresap ke dalam sanubari pembacanya maupun pendengarnya. Keindahan dalam membaca *qiro'ah* dibutuhkan pada saat acara seperti perlombaan, sambutan, pidato, pernikahan dan lain-lain. Namun, semua itu belum bisa dimaksimalkan ketika tampil di mimbar, mereka bisa mengarang lagu dan mengaransemen irama sesuai dengan kemampuan sendiri. Hal ini masih sering terjadi kesalahan *jail*, dan kurang sempurnanya dalam hal *fashohah adab* (Mausuli, 2010). Kebanyakan dari mereka yang ingin belajar irama *qiro'ah* hanya memahami sebagian teorinya saja, namun pada saat mempraktikkan mereka tidak dapat memahami jenis irama bacaan *qiro'ah* apa yang sedang mereka baca. Kurangnya pelatihan yang khusus untuk memahami hal tersebut disebabkan oleh kesulitan dalam mencari tenaga pengajar yang bisa mempraktikkan dan membedakan cara membaca Al-Quran dengan jenis-jenis irama *qiro'ah*.

Irama *qiro'ah* tergantung dari masing-masing yang membacanya dalam membawakan bacaan sesuai dengan metode *qiro'ah*. Seorang yang ingin mempelajari *qiro'ah* harus bisa menggunakan dan mengambil nafas selektif

mungkin, sehingga untuk mencapai pernapasan yang diinginkan membutuhkan aktivitas fisik dan latihan pernapasan yang teratur dan benar (Munir, 1995). Ketika seorang *qori'* atau *qori'ah* sedang membacakan ayat-ayat Al-Quran dengan begitu indah dan merdu, tentunya sebagai seorang *mustami'* (pendengar) akan terasa bergetar hatinya. Pada saat itulah seseorang mulai mengingatkan kebesaran Allah SWT, bahwa keindahan suatu anugrah yang diberikan oleh-Nya.

Dengan adanya suatu jenis irama dalam seni membaca Al-Quran, sangat diperlukan adanya suatu sistem untuk membantu para pecinta *qiro'ah* yang ingin belajar dengan menggunakan irama dalam berlatih seni membaca Al-Quran. Melalui perancangan dan pembangunan sebuah sistem yang dapat membantu para pecinta *qiro'ah* untuk mengetahui jenis irama *qiro'ah* apa yang sedang dibaca, dengan sebuah pendekatan terhadap suara manusia yang telah dianggap faham dengan jenis-jenis irama *qiro'ah*, sesuai dari tujuh jenis irama *qiro'ah*. Maka dari tujuh jenis tersebut diambil salah satu macam dari setiap jenis irama *qiro'ah*. Kemudian, sampel yang diambil menggunakan rekaman dari suara manusia. Kemudian, dari masalah ini diperlukan suatu algoritma pembelajaran mesin atau komputasi yang dapat digunakan untuk melakukan pengklasifikasian ciri suara dengan mempelajari ciri-ciri sebelumnya. Maka dari hal ini peneliti berkeinginan membuat sebuah aplikasi yang dapat membedakan tujuh jenis irama *qiro'ah* secara mandiri dan dapat membantu proses pembelajaran ilmu seni membaca Al-Quran sesuai jenis irama *qiro'ah* yang sudah ada dalam bentuk komputasi.

Berdasarkan latar belakang masalah yang sudah diuraikan, maka peneliti mencoba membuat sebuah sistem yang dapat membantu dalam hal proses pembelajaran seni membaca Al-Quran dengan irama *qiro'ah* dan juga yang baru

ingin mempelajari atau ingin mengetahui jenis irama *qiro'ah* apa yang sedang dibacakan. Melalui pendekatan suara manusia dengan judul penelitian: “Aplikasi Pengenalan Jenis-Jenis Irama *qiro'ah* Menggunakan Metode *Naive Bayes Classifier*”.

Sebelumnya, penelitian ini menggunakan sebuah pendekatan metode *Naive Bayes Classifier*. *Naive Bayes Classifier* (NBC) merupakan algoritma klasifikasi yang sangat efektif (mendapatkan hasil yang tepat) dan efisien (proses penalaran dilakukan dengan memanfaatkan input yang ada dengan cara yang relatif cepat) (Apriliana & Pulung, 2016). Metode ini dirasakan cukup untuk sebuah klasifikasi menentukan sebuah jenis irama *qiro'ah*. Karena, metode ini memiliki kelebihan yaitu, tingkat akurasi yang baik. Selain memiliki tingkat akurasi yang baik, metode ini juga mudah dipahami, sehingga dapat dijadikan sebuah pondasi untuk pengambilan keputusan. Kemudian, untuk membangun aplikasi ini, maka pada metode ini diperlukan sebuah *class* yang dapat menjadi acuan dari *data testing* yang akan diujicobakan dari pola model *data training* melalui penerapan klasifikasi probabilitas atau peluang.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka dapat dirumuskan suatu masalah yang relevan dengan judul penelitian ini yaitu:

1. Bagaimana menganalisis karakter suara sesuai jenis-jenis irama *qiro'ah* menggunakan metode *Naive Bayes Classifier*?
2. Seberapa tingkat akurasi yang dihasilkan dari penggunaan metode *Naive Bayes Classifier* dapat mengidentifikasi jenis-jenis irama *qiro'ah*?

1.3 Tujuan Penelitian

Tujuan dari penelitian yang dilakukan ini yaitu:

1. Mengimplementasikan metode *Naive Bayes Classifier* untuk pengenalan jenis-jenis irama *qiro'ah*.
2. Mengetahui tingkat akurasi yang dihasilkan dari penggunaan metode *Naive Bayes Classifier* dalam menentukan pengenalan jenis-jenis irama *qiro'ah*.

1.4 Batasan Penelitian

Adapun batasan penelitian yang terdapat dalam penelitian ini adalah:

1. Jenis-jenis irama *qiro'ah* yang akan diteliti adalah suara manusia yang mahir dalam membacakan tujuh jenis irama *qiro'ah* yaitu:
 - a. *Bayyati* dengan tingkatan suara *Jawab*
 - b. *Hijaz* dengan variasi *Kard*
 - c. *Shoba* dengan variasi Asli
 - d. *Rost* dengan tingkatan suara *Jawab*
 - e. *Jiharkah* dengan variasi Asli
 - f. *Sikah* dengan variasi Asli
 - g. *Nahawan* dengan variasi Asli
2. Data diambil dari perekaman suara manusia yang sudah terlatih.
3. Perekaman menggunakan *software Audacity* dengan kualitas rekaman, stereo 44100 Hz/32-bit float.
4. Durasi waktu perekaman suara antara 20 s/d 30 detik dan ayat yang akan dibacakan diambil di dalam Al-Quran yang akan ditentukan pada bab berikutnya.

5. Jarak perekaman antara mulut dan *microphone* pada sumber suara berjarak $\pm$ 10 cm.
6. Data suara rekaman disimpan dalam format .wav atau audio.
7. Jumlah data set yang diambil sebanyak 10 sampel suara, setiap suaranya diambil sebanyak 3 ayat dari setiap jenis iramanya. Jadi, total setiap orangnya memberikan 21 sampel rekaman suara.

1.5 Manfaat Penelitian

1. Manfaat teoretis adalah:
 - a. Sebagai referensi tambahan pada penelitian di bidang pengolahan suara terutama pada penelitian pengenalan irama suara.
 - b. Sebagai metode baru dalam pembelajaran seni *qiro'ah* Al-Quran secara otodidak.
2. Manfaat praktis adalah:
 - a. Tersedianya sebuah sistem yang mampu mengklasifikasikan irama *qiro'ah* sesuai dengan jenisnya.
 - b. Membantu calon *qori'* atau *qori'ah* dalam hal proses pembelajaran seni membaca Al-Quran dengan irama *qiro'ah*.
 - c. Memudahkan para pemula yang baru ingin mempelajari atau ingin mengetahui jenis irama *qiro'ah* apa yang sedang dibacakan.

1.6 Sistematika Penulisan

Susunan istematika penulisan pada laporan ini sebagai berikut:

1. BAB I Pendahuluan, pada bab ini peneliti menjabarkan tentang sub-bab yang terdiri dari, latar belakang, pernyataan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan. Latar Belakang masalah yang menjadi landasan dalam penelitian ini. Pernyataan masalah menjabarkan tentang pertanyaan-pertanyaan yang muncul berdasar pada latar belakang masalah. Tujuan Penelitian berisikan tujuan yang ingin dicapai peneliti dalam penelitian berdasar pernyataan masalah. Manfaat penelitian berisikan tentang manfaat yang didapat dalam penelitian. Batasan masalah yang berisikan tentang batasan yang terdapat dalam penelitian. Sistematika penulisan yang berisikan sub-bab yang terdapat dalam penelitian untuk mempermudah dalam menyelesaikan penelitian.
2. BAB II Studi Pustaka, dalam bab ini peneliti menjabarkan tentang berbagai pihak yang memiliki topik sama dengan penelitian yang sedang dilakukan. Landasan teori berisi pengertian dan definisi yang berdasar pada referensi kutipan buku tentunya berkaitan dengan penelitian.
3. BAB III Desain dan Implementasi, dalam bab ini peneliti menjabarkan tentang desain dari sistem hingga desain dari penelitiannya. Penjabaran tentang implementasi sistem hingga penjelasannya juga terdapat dalam bab ini.
4. BAB IV Uji Coba dan Pembahasan, dalam bab ini peneliti menjabarkan tentang langkah-langkah dan hasil uji coba. Pembahasan menjabarkan tentang analisis hasil uji coba yang sifatnya terpadu dan saling terkait, baik secara hasil yang diperoleh dan penelitian yang berbasis integrasi sains dan Islam.

5. BAB V Penutup, dalam bab ini peneliti menjabarkan tentang sub-bab yang terdiri dari kesimpulan dan saran. Kesimpulan berisi tentang pernyataan singkat dan tepat dari peneliti yang dijabarkan dari hasil penelitian pada pembahasan untuk menjawab pernyataan masalah secara singkat dan jelas, terlebih dalam menjawab dari tujuan penelitian. Saran berisi tentang pendapat peneliti berdasar pada pengalaman dan pertimbangan peneliti, yang ditujukan pada peneliti bidang sejenis dalam mengembangkan penelitian yang telah diselesaikan.
6. Daftar Pustaka, dalam bab ini berisi sumber-sumber atau buku-buku yang menjadi referensi dan acuan dalam menyelesaikan penelitian.

BAB II

KAJIAN PUSTAKA

2.1 Definisi-Definisi

Pada bagian ini, akan menjelaskan mengenai definisi dari seni membaca Al-Quran dengan *qiro'ah*, macam-macam jenis irama pada *qiro'ah*, dan lain sebagainya yang berkaitan erat terhadap penelitian ini.

2.1.1 Seni Membaca Al-Quran (*qiro'ah*)

Seni membaca Al-Quran dengan *qiro'ah* adalah membaca Al-Quran dengan menggunakan irama sebanyak tujuh macam irama dasar yaitu; *Bayati*, *Hijaz*, *Nahwand*, *Sikah*, *Rost*, *Shoba*, dan *Jiharkan*. Seni membaca Al-Quran dengan *qiro'ah* merupakan hal yang disunnahkan Rasulullah SAW dalam hadits yang artinya: *Hiasilah Al-Quran itu dengan suaramu, karena suara bagus itu akan menambahkan keindahan bagi Al-Quran*. (HR. Imam Hakim).

Dalam hadits di atas memberi pesan yaitu, tidak semua orang yang memiliki suara dengan indah. Namun, dengan suara yang indah menambahkan daya tarik tersendiri dalam hal seni membaca Al-Quran dengan *qiro'ah*. Karena, seni membaca Al-Quran sangat dibutuhkan oleh masyarakat umum seperti untuk acara pengajian, akad nikah, tasyakuran bahkan kerap dilombakan mulai tingkat desa, kecamatan, kabupaten, provinsi, sampai dengan nasional/internasional.

2.1.2 Persiapan Seorang Menjadi *Qori'* dan *Qori'ah*

Seorang yang akan mempelajari dan ingin menjadi *qori'* atau *qori'ah* wajib baginya untuk mempersiapkan hal-hal berikut:

1. Niat

Menjadi seorang *qori'* atau *qori'ah* hendaklah berniat belajar ikhlas hatinya semata-mata hanya karena Allah SWT, dengan meminta keridhoan dan mengharapkan pahalanya.

2. Mental

Faktor mental sangat diperlukan saat ingin menjadi seorang *qori'* atau *qori'ah*, sebab tanpa memiliki mental yang kuat pada saat belajar maka akan putus asa, tidak istiqomah, dan tidak sabar dalam proses belajar sehingga terganggu mentalnya.

3. Suara dan Vokal

Salah satu bagian terpenting menjadi seorang *qori'* atau *qori'ah* adalah suara atau vokal, dengan memiliki suara yang baik akan menambah semangat untuk belajar. Namun, jika memiliki suara yang kurang pas perlu pelatihan yang khusus mulai dari suara keras, sedang, dan merdu untuk memacu semangat untuk belajar kembali.

4. Pernafasan

Pernafasan bisa menjadi maksimal dengan baik, seorang *qori'* atau *qori'ah* diminta untuk berlatih dengan membaca ayat-ayat suci Al-Quran menggunakan pernapasan biasa atau panjang.

5. Membaca Al-Quran sesuai Tajwid

Terakhir, membaca Al-Quran sesuai tajwid dengan baik dan benar merupakan modal utama sebelum belajar seni membaca Al-Quran dengan *qiro'ah*. Karena, jika belum banyak mengetahui tentang tajwid akan mempengaruhi ketika sedang belajar *qiro'ah*.

Sedangkan, menurut Tarmuzi pada *ebook*-nya, *qiro'ah* memiliki tiga komponen dasar yaitu; suara, pernafasan dan bacaan Al-Quran dengan baik dan lancar (Syarif, 2015). Dalam hal mempelajari seni membaca Al-Quran dengan *qiro'ah* harus memerlukan persiapan yang khusus baik suara, nafas dan irama yang seimbang atau harmonis dengan bacaannya lebih hati-hati dengan penuh penjiwaan pada pembacanya disebut *qori'* atau *qori'ah*.

2.1.3 Jenis-Jenis Irama *qiro'ah*

Seperti pada penjelasan sebelumnya, bahwa ada tujuh jenis irama *qiro'ah* dalam seni membaca Al-Quran. Dari tujuh jenis irama tersebut terdapat beberapa macam tingkatan suara dan variasi yang dijelaskan sebagai berikut:

1. *Bayyati*

Irama *Bayyati* terdiri dari 12 bentuk (nada), dan tiga tingkatan suara, yaitu: *Qoror - Jawab - Jawabul Jawab*, dengan satu variasi yaitu *Syuri*.

2. *Hijaz*

Irama *Hijazi* atau *Hijaz* terdiri dari 7 bentuk dan 4 Variasi, yaitu: *Kard, Kard - Kurd, Naqrisy, dan Kurd*. Sedangkan, untuk tingkatan suara ada tiga: *Jawab - Jawabul Jawab - Qoror*.

3. *Shoba*

Irama *Shoba* terdiri dari 5 bentuk dengan 3 variasi, yaitu: *Ajami, Mahur (Muhur), dan Bastanjar*. Sedangkan, untuk tingkatan suaranya ada 2 yaitu: *Jawab dan Jawabul Jawab*.

4. *Rast*

Irama *Rost* terdiri dari 7 bentuk dan 3 Variasi, yaitu: *Usyaq*, *Zanjiron* (*Zinjiron*), dan *Syabir Alarros*. Sedangkan, tingkatan suaranya ada 2: *Jawab* dan *Jawabul Jawab*.

5. *Jiharkah*

Irama *Jiharkah* terdiri dari 4 bentuk dan 1 variasi, yaitu, *Kurdi*. Sedangkan, tingkatan suara ada 2, *Jawab* dan *Jawabul Jawab*.

6. *Sikah*

Irama *Sikah* terdiri dari 6 bentuk dan 4 variasi, yaitu: *Misri*, *Turki*, *Roml*, dan *Uroq*. Sedangkan, tingkatan suaranya ada 3 : *Qoror*, *Jawab*, dan *Jawabul Jawab*.

7. *Nahawan*

Irama *Nahawan* atau *iroqi* terdiri 5 bentuk dan 2 variasi, yaitu: *Nuqrosy* dan *Murokkab*. Ciri-ciri variasi *Nuqrosy* adalah bernada rendah (turun). Sedangkan, variasi *Murokkab* bernada tinggi (naik). Adapun tingkatan suaranya ada 2, yaitu: *Jawab* dan *Jawabul Jawab*.

2.1.4 Suara

Dalam istilah fisik murni, suara adalah gelombang longitudinal yang bergerak melalui udara (atau gelombang melintang di beberapa media lain) karena getaran molekul. Di udara, suara ditransmisikan sebagai variasi tekanan, antara tekanan tinggi dan rendah, dengan laju variasi tekanan dari rendah, ke tinggi, ke rendah lagi, menentukan frekuensi. Tingkat variasi tekanan (yaitu perbedaan antara tinggi dan rendah) menentukan amplitude (Mcloughlin, 2009). Dalam istilah lain suara adalah fenomenal fisik disebabkan oleh getaran suatu benda

berupa sinyal analog dan amplitudo yang bisa berubah secara kontinu terhadap waktu disebut frekuensi.

Pembentukan suara terbagi menjadi tiga proses, yaitu; artikulasi, radiasi, dan pembangkitan sumber. Organ tubuh yang berkaitan dalam proses produksi suara manusia meliputi paru-paru, tenggorokan (*trachea*), laring (*larinx*), faring (*pharynx*), rongga hidung (*nasal cavity*), dan rongga mulut (*oral cavity*). Sehingga, terdapat suatu lintasan *vocal tract* yang terdiri dari faring atau hubungan antara krongkongan dengan mulut (Rabiner & Juang, 1993). Apabila lintas vokal berubah itu disebabkan oleh adanya pergerakan rahang, bibir, lidah, dan organ internal lainnya.

Sistem pendengaran manusia terjadi pada telinga yang terbagi pada tiga bagian utama yaitu; telinga luar, telinga tengah, dan telinga dalam (Sanjaya, 2016). Prinsip kerja pada sistem pendengaran manusia yaitu:

1. Gelombang suara (bunyi) yang masuk ke dalam telinga luar menggetarkan bagian gendang telinga.
2. Getaran tersebut mengakibatkan cairan limfa yang bergerak pada saluran telinga tengah hingga menggetarkan cairan dalam saluran timpani.
3. Pada frekuensi tertentu akan menggetarkan selaput-selaput *basiler* yang berfungsi menggerakkan sel-sel rambut telinga ke atas dan ke bawah. Sehingga, getaran tersebut menghasilkan sebuah impuls yang akan dikirim ke pusat pendengar didalam otak melalui saraf pendengaran.

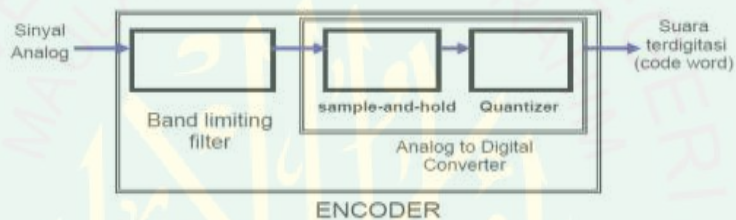
2.1.5 Representasi Suara

Gelombang bunyi atau suara analog tidak dapat langsung direpresentasikan ke dalam komputer. Komputer terlebih dahulu mengukur

amplitudo pada satuan waktu (t) tertentu, kemudian menghasilkan sebuah angka. Pada setiap satuan pengukuran ini dinamakan *sample*. Sehingga, hal tersebut harus dilakukan *conversion* data analog untuk dijadikan data digital.

1. ADC (*Analog to Digital Conversion*)

Analog to Digital Conversion merupakan sebuah proses mengubah amplitudo gelombang bunyi ke dalam waktu interval yang disebut *sampling*, sehingga menghasilkan representasi digital pada suara. Dalam proses ini dikenal dalam istilah *sampling rate* yaitu, proses beberapa gelombang yang diambil dalam waktu satu detik.



Gambar 2.1 Proses Sampling

Pada Gambar 2.1 merupakan proses *sampling* melibatkan pembuangan frekuensi yang tinggi dari *source* sinyal, sehingga dapat diambil sebuah sampel pada waktu interval tertentu (*sampling*). Sampel amplitudo akan menyimpannya dan mengubahnya ke dalam bentuk diskrit. Dari proses tersebut akan menghasilkan bentuk menjadi nilai biner 0 dan 1.

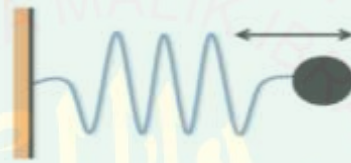
2. DAC (*Digital to Analog Conversion*)

Sebuah Proses mengubah digital audio menjadi sinyal analog. *Digital to Analog Converter* biasanya hanya dapat menerima sinyal digital (PCM) *Pulse Code Modulation*. PCM merupakan sebuah representasi digital dari sinyal analog, dimana gelombang yang terjadi secara beraturan berdasarkan waktu

interval tertentu. Kemudian, akan diubah menjadi nilai biner, proses pengubahan ke dalam biner disebut Quantisasi.

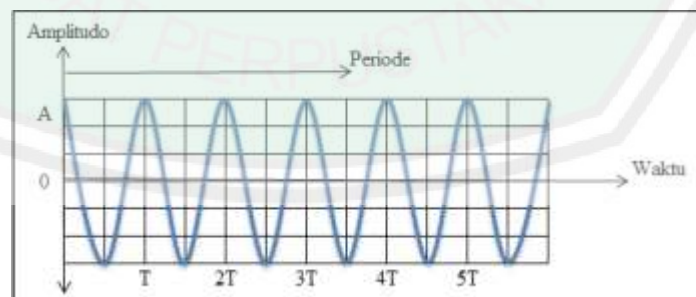
2.1.6 Amplitudo

Amplitudo merupakan simpangan dari suatu getaran yang menghasilkan suara tersebut (Susetyo & Altedzar, 2012). Amplitudo dalam sistem internasional disimbolkan dengan (A) dan satuan amplitudo tersebut adalah meter (m) yang bisa diartikan simpangan atau jarak terjauh dari titik kesetimbangan. Berikut gambar di bawah ini merupakan ilustrasi berkaitan dengan amplitudo:



Gambar 2.2 Visualisasi Gelombang Amplitudo

Pada Gambar 2.2 dapat diambil contoh, jika sebuah gelombang awal yang diam berada diposisi kesetimbangan, kemudian bergerak dari posisi awal ke bawah selanjutnya ke atas dan bergerak terus menurun secara teratur hingga gelombang kembali pada posisi awal kesetimbangan, maka simpangan atau jarak diukur yang terjauh dari titik kesetimbangan disebut amplitudo.



Gambar 2.3 Gelombang Amplitudo Berkaitan dengan Waktu

Pada Gambar 2.3 merupakan amplitudo dihasilkan berupa bunyi oleh suatu yang bergetar. Amplitudo dapat menentukan keras lemahnya suatu bunyi

atau suara, sehingga jika amplitudo puncak gelombang maksimum dan energi yang dibawa oleh gelombang suara akan semakin besar, maka intensitas bunyinya juga semakin tinggi, dan sebaliknya.

2.1.7 *Similarity*

Similarity merupakan metode untuk menghitung kesamaan antara dua buah objek yang dinyatakan dalam dua buah *vector* dengan menggunakan *keywords* (kata kunci) dari sebuah dokumen sebagai ukuran (Rizky Tri Wahyuni, Dkk, 2017). Namun, pada proses ini berfungsi untuk menambahkan proses klasifikasi agar nantinya pada proses tahap akhir menemukan hasil yang baik.

2.2 *Klasifikasi Naïve Bayes Classifier*

Klasifikasi adalah suatu pelatihan terhadap misalkan target A untuk pemetaan setiap fitur atau ciri X ke satu dari label Y yang telah tersedia. Algoritma klasifikasi ini memerlukan sebuah data *training* untuk membuat suatu model pelatihan atau pembelajaran. Dari model yang sudah ditetapkan tersebut, nantinya akan digunakan untuk memprediksi kemungkinan label kelas dari data baru, tetapi belum diketahui.

Ada beberapa algoritma atau metode untuk menyelesaikan klasifikasi antara lain seperti *Decision Tree*, Jaringan Saraf Tiruan (JST), *K-Nearest Neighbour* (KNN), *Naïve Bayes Classification* (NBC), dan lain sebagainya. Adapun prinsip kerja algoritma klasifikasi untuk melakukan suatu pelatihan, sehingga pada tahap akhir petihan model tersebut dapat memprediksi setiap *vector* masukan yang pada label kelas yang tepat untuk data baru tersebut.

2.2.1 Teorema Bayes

Algoritma *Naïve Bayes Classifier* merupakan suatu bentuk klasifikasi data dengan menggunakan metode statistik dan probabilitas. Metode ini pertama kali dikenalkan oleh ilmuwan Inggris Thomas Bayes, yaitu digunakan untuk memprediksi peluang yang terjadi berdasarkan pengalaman sebelumnya, maka dikenal sebagai teorema Bayes. Teorema Bayes merupakan satu metode yang digunakan untuk menghitung ketidakpastian data menjadi data yang pasti dengan membandingkan antara data ya dan tidak (Andriani & Dwi, 2016). Untuk mengukur suatu kejadian secara acak yang berhubungan dengan ketidakpastian dapat menggunakan prediksi yang didasarkan pada bentuk teorema Bayes.

Bentuk teorema Bayes untuk *evidence* tunggal E dan hipotesis tunggal H adalah:

$$p(H|E) = \frac{p(E|H) \times p(H)}{p(E)} \dots (1)$$

Dimana:

$p(H|E)$ = Probabilitas hipotesis H terjadi jika *evidence* E terjadi.

$p(E|H)$ = Probabilitas munculnya *evidence* E, jika hipotesis H terjadi.

$p(H)$ = Probabilitas hipotesis H tanpa memandang *evidence* apapun.

$p(E)$ = Probabilitas *evidence* E tanpa memandang apapun.

Teorima Bayes tersebut merupakan dasar dari hasil hipotesis atau terjadinya peristiwa (H) yang dapat diperkirakan berdasarkan pada beberapa bukti (E) yang diamati.

2.2.2 Penerapan *Naïve Bayes Classifier* untuk Klasifikasi

Klasifikasi suatu data dengan menggunakan *Naïve Bayes* merupakan suatu pengembangan dari teorema Bayes yang dilihat bahwa suatu peristiwa A terjadi

dengan syarat adanya bukti yaitu B, dan bukti tersebut terjadi dengan syarat peristiwa A telah terjadi. Sehingga teorema ini menjadi teorema yang disebut “*Naïve*” karena diterima sebagai dasar bahwa kondisi antar *atribut* saling memiliki kebebasan. Pada metode *Naïve Bayes Classifier* memiliki *class-class* yang menjadi target sebuah pemetaan dalam klasifikasi, kemudian pada tahap data uji akan ditentukan kelas mana yang lebih optimal. Sehingga, klasifikasi didapatkan berdasarkan bukti yang telah berubah menjadi fitur pada proses tahap ini.

Selama berada pada proses pelatihan harus dilakukan suatu pembelajaran probabilitas akhir $P(Y|X)$ pada model dari setiap kombinasi X dan Y berdasarkan informasi yang didapatkan dari data *training*. Dalam membangun model tersebut, suatu data *testing* X' dapat diklasifikasikan dengan mencari nilai dari Y' dengan memaksimalkan nilai dari $P(X'|Y')$ yang dihasilkan. Formula *Naïve Bayes* untuk klasifikasi adalah:

$$P(X|Y) = \frac{P(Y) \prod_{i=1}^q P(X_i|Y)}{P(X)} \dots (2)$$

Dimana:

$P(Y|X)$ merupakan probabilitas data dengan *vector* X pada kelas Y. $P(Y)$ probabilitas awal dari kelas Y.

$\prod_{i=1}^q P(X_i|Y)$ adalah probabilitas independen kelas Y dari semua fitur dalam *vector* X.

Nilai $P(X)$ tetap sehingga dalam perhitungan kemungkinan nantinya tinggal mengitung bagian $P(Y) \prod_{i=1}^q P(X_i|Y)$ dengan memilih nilai yang terbesar sebagai kelas yang dipilih sebagai hasil prediksi.

Secara umum, metode *Bayes* dihitung untuk fitur dengan tipe berkategori seperti pada kasus implementasi *Naïve Bayes Classifier* pada program bantu penentuan buku referensi matakuliah jurnal karya Atri bersama rekan-rekannya. Namun, pada fitur numerik terdapat perlakuan khusus sebelum dimasukan kedalam metode *Naïve Bayes Classifier* dengan cara, yaitu:

1. Dengan melakukan sebuah diskritisasi dari setiap fitur kontinu dan juga mengganti nilai kontinu tersebut kedalam nilai interval diskrit. Kemudian, pendekatan ini dilakukan dengan merubah fitur kontinu tersebut kedalam fitur ordinal.
2. Memperkirakan bentuk tertentu dari probabilitas distribusi untuk fitur kontinu dan mengasumsikan parameter distribusi dengan data *training*.

Sehingga, distribusi Gaussian lebih dipilih untuk mempresentasikan probabilitas bersyarat dari fitur kontinu pada sebuah kelas $P(X_i|Y)$. Sedangkan, distribusi Gaussian memiliki karakteristik dengan dua parameter yaitu, *mean* (μ) dan *varian* (σ^2) Untuk setiap kelas y_j dan probabilitas bersyarat kelas y_j untuk X_i adalah:

$$P(X_i = x_i | Y = y_j) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} \exp \frac{-(x_i - \mu_{ij})^2}{2\sigma_{ij}^2} \dots (3)$$

Parameter μ_{ij} didapatkan dari *mean* sampel $X_i(x')$ dari seluruh data *training* yang menjadi milik kelas y_j . Sedangkan, σ_{ij}^2 dapat diperkirakan dari *varian* sampel (σ^2) dari data *training*.

2.2.3 Kinerja Klasifikasi Metode *Naïve Bayes Classifier*

Salah satu cara untuk mengukur suatu kinerja klasifikasi suatu objek yaitu, dengan menghitung tingkat akurasi sebuah metode yang digunakan terhadap

aplikasi yang ingin dibangun. Selanjutnya, dari pengukuran tersebut diharapkan aplikasi yang telah dibangun dapat melakukan sebuah proses klasifikasi terhadap seluruh set data yang telah diuji cobakan. Meskipun tidak dipungkiri bahwa dalam membangun sebuah aplikasi, tingkat akurasi suatu aplikasi tidak harus 100% akurat atau mendekati 100% akurat, tetapi bagaimana proses dalam membangun sebuah aplikasi dan menemukan hasil akurasi terbaik yang diperoleh.

2.3 *NetBeans IDE*

NetBeans IDE merupakan pengembangan perangkat lunak yang ditulis dalam bahasa pemrograman *Java*. Pengembangan suatu aplikasi dapat dilakukan dimulai dari setelan perangkat lunak modular bernama modules. Semula, aplikasi *NetBeans IDE* ini diperuntukkan bagi pengembangan dalam *Java*. Perangkat lunak ini biasa digunakan untuk teknik komputasi numerik, yang didalamnya dapat menyelesaikan terkait masalah operasi elemen matematika, optimasi, visualisasi, grafis, simulasi, statistika, analisis data, dan lain sebagainya. Contoh program dengan memanfaatkan *software NetBeans IDE* antara lain seperti:

1. Kalkulator menggunakan *GUI* pada *NetBeans*.
2. Aplikasi Penjualan Barang
3. Analisis data untuk penentuan beasiswa.

Hampir sama dengan bahasa pemrograman lainnya, pemrograman *java* juga meliputi perulangan, *array*, grafik, dan lain sebagainya. Ketika membuat program pada *NetBeans*, didalam *folder* terdapat *class - class* yang didalamnya berisi *code-code* program. Dalam penelitian ini, penulis memanfaatkan *software NetBeans*

sebagai *editor* dalam membangun aplikasi pengenalan jenis irama *qiro'ah* memanfaatkan *GUI-NetBeans IDE 8.1*.

2.4 Penelitian Terkait

Berdasarkan latar belakang yang sudah diuraikan pada bab sebelumnya, maka pada tahap selanjutnya adalah penelitian terkait sebagai pendukung penelitian ini.

Penelitian terkait pertama bidang pengolahan suara tertuju pada Susetyo Bagas Bhaskoro dan Altedzar Riedho W. D tentang aplikasi pengenalan gender menggunakan suara menggunakan *Linear Predictive Coefficiency* (LPC) yang merupakan salah satu metode pemodelan suara dan *Learning Vector Quantization* (LVQ) sebagai proses pengklasifikasian suatu objek terhadap fitur yang sudah tersimpan. Pada penelitian ini data fitur yang digunakan untuk menentukan jenis kelamin berdasarkan suara adalah pitch dan formant. Sehingga, secara keseluruhan didapatkan bahwa penelitian menggunakan pemodelan suara *Linear Predictive Coefficiency* (LPC) dan *Learning Vector Quantization* (LVQ) dapat dikembangkan ke tahap lebih lanjut dengan data *training* yang lebih banyak lagi, tempat yang lebih sunyi, dan parameter lainnya agar mendapatkan hasil yang lebih dari yang diharapkan. Pada penelitian ini hasil yang didapatkan berdasarkan jumlah data *training* 5 data (5 laki-laki dan 5 perempuan), 10 data (10 laki-laki dan 10 perempuan) dan 15 data (15 laki-laki dan 15 perempuan) dengan uji coba ke-33 pembicara yang berbeda maka akan mendapatkan hasil sesuai urutan adalah 53,94%, 56,36% dan 75,76% (Susetyo & Altedzar, 2012).

Kemudian, penelitian terkait selanjutnya yaitu dengan judul “Klasifikasi Genre Pada Lagu Dengan Metode *Naïve Bayes*” makalah ini ditulis oleh Ardi Priagung bersama rekan-rekannya menjelaskan bahwa metode *Naïve Bayes* merupakan klasifikasi yang memang benar-benar dapat diterapkan secara global, sehingga klasifikasi tersebut dapat spesifik untuk golongan manusia tertentu dan musik-musik tertentu. Dimana setiap rentang frekuensi dari setiap fitur dibagi menjadi 3 kategori yaitu, rendah, sedang, dan tinggi. Kategori dari setiap fitur inilah yang nantinya dilanjutkan kedalam proses klasifikasi menggunakan metode *Naive Bayes*. Dalam jurnal ini menjelaskan bahwa frekuensi merupakan fitur yang tidak selamanya benar dalam memprediksi atau menentukan sebuah genre sebuah lagu, masih banyak fitur lain lagi yang dapat dipakai. Sehingga, dari hasil pengujian yang telah dilakukan maka dapat ditarik beberapa kesimpulan bahwa metode yang digunakan pengklasifikasian genre pada lagu dengan fitur frekuensi rendah, sedang, dan tinggi yang mempunyai akurasi sebesar 83,3% untuk data latih dan 60% untuk data test (Priagung, dkk. 2014).

Penelitian terkait selanjutnya yaitu berjudul “Pengujian Metode *Naïve Bayes Classifier* untuk Klasifikasi Genre Musik Berdasarkan Frekuensi” yang ditulis oleh Evitasari Nur Apriliana dan Pulung Nurtantio Andono menjelaskan bahwa metode *naïve bayes classifier* merupakan salah satu metode yang digunakan dalam data mining untuk mengklasifikasikan data berdasarkan kelas yang telah ditentukan sebelumnya, metode ini sangat efektif dan efisien dalam pengklasifikasian data. Hampir sama dengan penelitian sebelumnya tentang klasifikasi genre pada lagu, penggunaan metode *Naive Bayes* yang berguna untuk memberikan kemudahan kepada para *user* untuk menentukan *genre music* yang

akan didengarkan. Dalam jurnal ini, untuk mengklasifikasikan genre musik tersebut, ekstraksi fitur pada setiap genre musik yang akan diolah dilakukan secara bertahap setelah *pre-processing*. Dari semua genre musik baik genre jazz, blues, klasik, pop, qasidah, dan rock akan memiliki nilai frekuensi yang berbeda. Nilai dari setiap fitur inilah yang nantinya akan diproses menggunakan metode *Naïve Bayes*. Sehingga, dari hasil pengujian yang telah dilakukan maka dapat diambil beberapa kesimpulan bahwa metode *Naïve Bayes Classifier* dapat digunakan dalam pengklasifikasian genre musik dengan fitur atribut frekuensi minimal, frekuensi rata-rata, frekuensi maksimal, amplitudo minimal, maksimal dan rata-rata yang mempunyai akurasi sebesar 60 % (Apriliana & Pulung, 2016).

Penelitian terkait yang terakhir yaitu Ramadhan Rosihadi Perdana bersama rekan-rekannya membahas tentang implementasi ekstraksi fitur untuk pengelompokan berkas musik berdasarkan kemiripan karakteristik suara. Ekstraksi fitur yang digunakan adalah untuk mencari *sample data* dan *sample rate*. Pada penelitian ini uji coba dilakukan dengan membandingkan nilai akurasi hasil ekstraksi fitur yang dilakukan dengan beberapa metode klasifikasi, uji coba tersebut adalah untuk membandingkan metode *naïve bayes*, *random forest*, *multi layer*, *perceptron*, *support vector machine*, dan J48. Uji coba skenario 3 ini menggunakan sebuah kombinasi dari 1000 buah berkas musik dengan 7 *ganre music classical*, *country*, *hiphop*, *jazz*, *metal*, *pop*, *rock*. Dari hasil uji coba yang dilakukan ditarik kesimpulan, fitur yang dihasilkan saat proses ekstraksi fitur musik ini mampu mempresentasikan musik berdasarkan genre dengan baik. Terbukti dengan nilai akurasi persentase 80,4762% ketika diklasifikasikan menggunakan metode *naïve bayes* (Ramadhan, dkk. 2017).

Pada penelitian ini penulis mencoba membuat aplikasi pengenalan jenis-jenis irama *qiro'ah* menggunakan metode *Naïve Bayes Classifier* untuk mengklasifikasikan suara. Penelitian ini berhubungan dengan penelitian terkait yang berjudul 'Klasifikasi Genre Pada Lagu Dengan Metode *Naïve Bayes*' (Priagung, dkk. 2014) dan berjudul 'Pengujian Metode *Naïve Bayes Classifier* untuk Klasifikasi Genre Musik Berdasarkan Frekuensi' (Apriliana & Pulung, 2016). Penelitian ini saling berhubungan pada proses tahapan yang dilakukan untuk menguji hasil ekstraksi fitur dalam merepresentasikan suara berdasarkan objek yang dimiliki. Data nilai atribut yang dihasilkan pada tahapan ekstraksi fitur akan digunakan sebagai data *training* dan data *testing* untuk dilakukan klasifikasi dengan menggunakan metode klasifikasi *naïve bayes classifier*. Keterbaruan penelitian ini adalah penulis menggunakan objek dengan jenis irama *qiro'ah*. Parameter yang akan digunakan dengan mencari nilai minimal, maksimal amplitudo dan menambahkan parameter sudut turun, dan sudut naik amplitudo. Sehingga, diharapkan sebagai referensi tambahan pada penelitian di bidang pengolah suara terutama pada penelitian pengenalan jenis irama *qiro'ah*.

BAB III

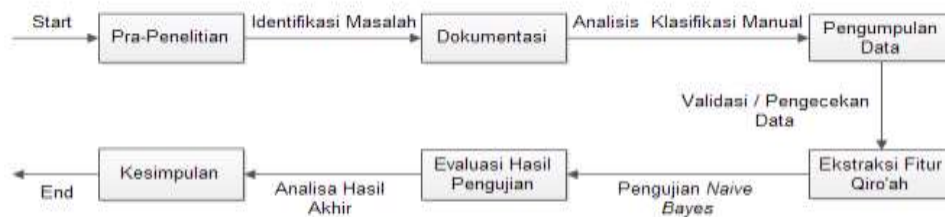
METODE PENELITIAN

Pada bab ini akan dijabarkan mengenai metode penelitian yang mulai dari prosedur penelitian, pengumpulan data sampel rekaman *qiro'ah*, desain proses, desain *interface*, serta implementasi metode yang akan digunakan. Penelitian ini merupakan sebuah sistem aplikasi pengenalan jenis irama *qiro'ah* yang mengolah nilai minimal, maksimal, sudut turun, dan sudut naik amplitudo untuk diproses ketahap berikutnya menggunakan metode *Naïve Bayes Classifier*. Selanjutnya, akan dilakukan proses perhitungan dari nilai-nilai tersebut untuk dijadikan sebuah parameter atau nilai *input* pada perlakuan *Naïve Bayes Classifier*.

Sistem aplikasi ini dibangun untuk pengenalan jenis irama *qiro'ah* yang sedang dalam tahap proses belajar dan mempraktikan langsung melalui rekaman jenis irama *qiro'ah* sebagai data *testing*. Kemudian, sampel rekaman *qiro'ah* yang sudah mahir dalam hal tersebut digunakan sebagai data *training*.

3.1 Prosedur Penelitian

Prosedur penelitian ini menjelaskan langkah-langkah dari setiap alat untuk pengumpulan data dan menjawab pertanyaan-pertanyaan dalam penelitian. Adapun prosedur atau cara kerja mengenai sejumlah kegiatan yang akan dilakukan dalam penelitian ini akan dipaparkan ke dalam bentuk *blok diagram* seperti yang terlihat pada Gambar 3.1 berikut:



Gambar 3.1 Prosedur Penelitian

Berdasarkan pada Gambar 3.1 prosedur penelitian di atas, dapat dilihat bahwa ada beberapa tahap yang harus dilakukan untuk menyelesaikan penelitian ini. Berikut di bawah ini adalah penjelasan mengenai setiap proses dari prosedur penelitian:

1. Pra-Penelitian

Bagian Pra-Penelitian ini, penulis secara tidak sengaja menemukan adanya permasalahan yang terjadi di sekitar kita dalam hal pembelajaran dan mempraktikan seni membaca Al-Quran menggunakan *qiro'ah*. Hal tersebut memicu penulis untuk memecahkan sebuah masalah yang ada sebagai topik penelitian.

2. Identifikasi Masalah

Agar dapat menentukan sebuah solusi yang terbaik untuk memecahkan masalah tersebut, maka, perlu adanya suatu identifikasi masalah sebagai berikut:

- a. Belum adanya suatu sistem untuk pembelajaran dan mempraktikan seni membaca Al-Quran menggunakan *qiro'ah*.
- b. Para pecinta *qiro'ah* yang ingin mempelajari lebih memperdalam terkait seni membaca Al-Quran secara mandiri untuk mengoreksi jenis irama apa yang sedang dibaca.

3. Dokumentasi

Setelah identifikasi masalah diketahui, selanjutnya dilakukan proses dokumentasi penelitian. Baik dokumentasi berupa *tekstual*, data *audio.wav* maupun data *visual*.

4. Analisis Klasifikasi Manual

Tahap selanjutnya, mengklasifikasi data rekaman *qiro'ah* yang akan dicari. Klasifikasi ini nantinya akan berpengaruh terhadap metode *Naïve Bayes Classifier* yang akan digunakan. Klasifikasi tersebut mengacu kepada *standar* tujuh jenis irama *qori'ah*.

5. Pengumpulan Data

Pada bagian ini, penulis mengumpulkan data rekaman *qiro'ah* melalui perekaman menggunakan laptop. Proses perekaman ini dilakukan selama $\pm$ 20-30 menit setiap sampel rekaman. Rekaman akan dilakukan didalam ruangan atau studio karena, sebuah rekaman yang baik harus dilakukan secara strategis diruangan yang tenang dan jauh dari kebisingan. Jarak mikrofon atau media perekam konstan pada $\pm$ jarak 10 cm.

Proses perekaman *qiro'ah* untuk satu data dilakukan sebanyak 2-3 kali perekaman yang bertujuan agar data yang diperoleh nantinya akan divalidasi data *qiro'ah* mana yang terbaik untuk diteliti. Untuk data *training* yang di ambil sebanyak 10 sampel suara, masing-masing sample suara akan membacakan 3 ayat Al-Quran yang sudah ditetapkan dan setiap ayat akan memberikan sample 7 jenis irama Qori'ah. Adapun Ayat yang akan yang dibacakan pada saat pengambilan sample suara rekaman sebagai berikut:

❖ - وَأَقِيمُوا الصَّلَاةَ وَآتُوا الزَّكَاةَ وَارْكَعُوا مَعَ الرَّاكِعِينَ

Artinya: “Dan dirikanlah shalat, tunaikanlah zakat dan ruku'lah beserta orang-orang yang ruku'” (Q.S Al-Baqarah: 43).

❖ وَإِذْ آتَيْنَا مُوسَى الْكِتَابَ وَالْفُرْقَانَ لَعَلَّكُمْ تَهْتَدُونَ

Artinya: “Dan (ingatlah), ketika Kami berikan kepada Musa Al Kitab (Taurat) dan keterangan yang membedakan antara yang benar dan yang salah, agar kamu mendapat petunjuk” (Q.S Al-Baqarah: 53).

❖ وَالَّذِينَ آمَنُوا وَعَمِلُوا الصَّالِحَاتِ أُولَٰئِكَ أَصْحَابُ الْجَنَّةِ ۖ هُمْ فِيهَا خَالِدُونَ

Artinya: “Dan orang-orang yang beriman serta beramal saleh, mereka itu penghuni surga; mereka kekal di dalamnya” (Q.S Al-Baqarah: 82).

6. Validasi / Pengecekan Data

Selanjutnya, tahap validasi atau pengecekan data. Pengecekan data dilakukan untuk menyeleksi data mana saja yang dapat digunakan untuk penelitian dan data mana saja yang tidak dapat digunakan, data yang tidak dapat digunakan akan diamankan atau dipisahkan. Apabila data rekaman *qiro'ah* sudah diperoleh, maka dilakukan proses *editing*. Proses *editing* ini dilakukan untuk memotong data suara yang tidak memiliki gelombang suara yang dapat diteliti. Karena, suara satu dengan suara yang lainnya bisa muncul pada selang waktu yang berbeda. Sehingga, dapat mengakibatkan fase gelombang tidak sama.

Tahap untuk melakukan proses *editing* data rekaman *qiro'ah* penulis menggunakan aplikasi “Audacity”. Kemudian, dilanjutkan dengan proses pengolahan data rekaman *qiro'ah* berikutnya.

7. Ekstraksi Fitur Jenis *qiro'ah*

Pada tahap ini, setelah data yang diperoleh tervalidasi akan diekstraksi jenisnya agar dapat dikenali dan sebagai data pembanding terhadap data *testing*. Hasil dari ekstraksi ini akan disimpan pada *database*.

8. Pengujian *Naïve Bayes Classifier*

Pengujian *Naïve Bayes Classifier* dilakukan dengan cara memasukkan data *testing* pada aplikasi pengenalan jenis irama *qiro'ah* dan akan diambil jenisnya atau fitur dari data *testing*. Selanjutnya, hasil dari jenis data *testing* diperbandingkan dengan seluruh data yang ada pada *database*. Kemudian, hasil yang sudah diperoleh akan dijumlah dengan hasil dari *similarity*.

9. Evaluasi Hasil Pengujian

Proses evaluasi hasil pengujian dilakukan untuk melihat bagaimana sistem pengenalan jenis irama *qiro'ah* digunakan dengan baik. Baik dari segi penggunaan maupun dari segi keakuratan komputasinya.

10. Analisis Hasil Akhir

Tahap selanjutnya, adalah analisa hasil akhir penelitian keseluruhan terhadap hasil pengujian. Analisa hasil akhir ini bertujuan mengetahui tingkat akurasi metode *Naïve Bayes Classifier* yang digunakan terhadap 63 sampel rekaman yang diujicobakan. Sehingga, dapat menghasilkan kesimpulan terhadap apa yang telah dikerjakan.

11. Kesimpulan

Hasil keseluruhan baik terhadap bentuk kinerja sistem dan hasil komputasinya selanjutnya, akan direkapitulasi dan akan menjadi hasil akhir tentang

bagaimana metode *Naïve Bayes Classifier* dapat diterapkan pada penelitian ini.

Kesimpulan ini akan dicantumkan pada bab terakhir penulisan penelitian atau karya tulis ilmiah.

3.2 Deskripsi Data Objek

Data yang akan digunakan dalam penelitian ini merupakan diambil dari data suara manusia yang sudah mahir dalam bidangnya yaitu, mempraktikan dengan membacakan sebuah ayat Al-Quran menggunakan irama ke tujuh jenis irama *qiro'ah*. Data suara rekaman *qiro'ah* yang sudah mahir tersebut ialah; *Bayyati* dengan tingkatan suara *Jawab*, *Hijaz* dengan variasi *Kard*, *Shoba* dengan variasi *Asli*, *Rost* dengan tingkatan suara *Jawab*, *Jiharkah* dengan variasi *Asli*, *Sikah* dengan variasi *Asli*, *Nahawan* dengan variasi *Asli*.

Adapun pemilihan data objek untuk data *training* dan data *testing* sebagai berikut:

1. Pemilihan sampel suara rekaman setiap jenis irama *qiro'ah* yang terbaik sesuai kriteria akan dimasukkan ke dalam data *training*, di antaranya sebanyak:
 - a. 30 data rekaman jenis irama *qiro'ah* (*Bayyati* dengan tingkatan suara *Jawab*).
 - b. 30 data rekaman jenis irama *qiro'ah* (*Hijaz* dengan variasi *Kard*).
 - c. 30 data rekaman jenis irama *qiro'ah* (*Shoba* dengan variasi *Asli*).
 - d. 30 data rekaman jenis irama *qiro'ah* (*Rost* dengan tingkatan suara *Jawab*).
 - e. 30 data rekaman jenis irama *qiro'ah* (*Jiharkah* dengan variasi *Asli*).
 - f. 30 data rekaman jenis irama *qiro'ah* (*Sikah* dengan variasi *Asli*).

g. 30 data rekaman jenis irama *qiro'ah* (Nahawan dengan variasi Asli).

Kemudian, dari kriteria rekaman di atas akan dikumpulkan sesuai ayat yang telah dibaca, untuk diproses ketahap ekstraksi ciri fitur.

2. Pemilihan data *testing* yang terdiri dari 21 data sampel rekaman. Dari data sampel rekaman tersebut terdiri 3 sampel setiap jenis iramanya dengan membacakan ayat Al - Qur'an yang sama pada saat mencari sampel untuk data *training*, baik jenis irama Bayyati dengan tingkatan suara Jawab, Hijaz dengan variasi Kard, Shoba dengan variasi Asli, Rost dengan tingkatan suara Jawab, Jiharkah dengan variasi Asli, Sikah dengan variasi Asli, maupun Nahawan dengan variasi Asli. Nantinya, akan dijadikan sebagai data *testing* untuk melakukan uji coba sistem aplikasi yang akan dikerjakan.

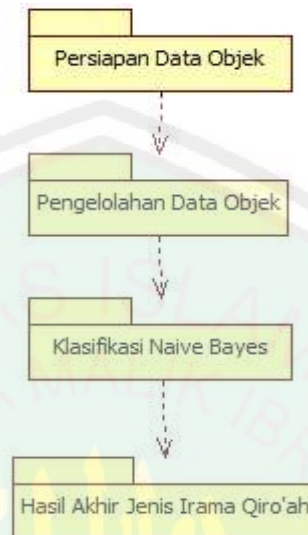
3.3 Desain Penelitian

3.3.1 Desain Proses

Pada umumnya, desain proses dasar pendukung perancangan untuk membangun sebuah aplikasi menggunakan UML (*Unified Modelling Language*). *Unified Modelling Language* merupakan sebuah “bahasa” yang telah menjadi standart dalam industri untuk visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak (Dharwiyanti & Satria, 2003). UML menawarkan banyak model diagram, pada penelitian ini penulis menggunakan *software* “*StarUML*” untuk membuat model perancangan. Adapun model untuk membuat diagram menggunakan *Use Case Diagram*.

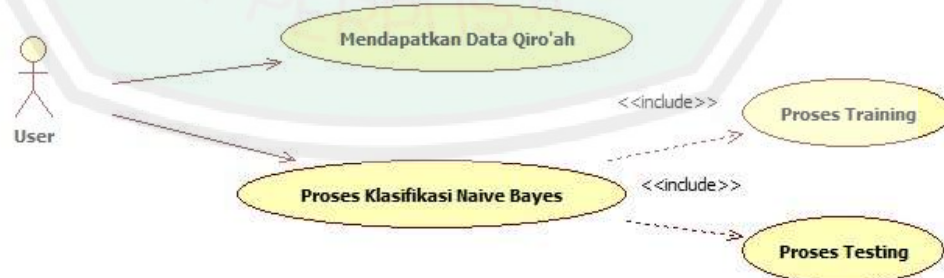
Use Case Diagram merupakan gambaran fungsionalitas yang diharapkan dalam sebuah sistem yang akan dibangun. Menggunakan *Use Case Diagram*

disini lebih menekankan gambaran “Apa?” yang ingin dilakukan sebuah sistem, bukan “Bagaimana?”. Dengan kata lain, sesuatu pekerjaan tertentu yang mempresentasikan sebuah interaksi antara pengguna / aktor dengan sistem.



Gambar 3.2 Desain Proses

Gambar 3.2 merupakan gambaran yang akan dibangun secara keseluruhan sistem yang terbagi menjadi empat proses, yaitu; proses persiapan data objek, pengolahan data objek, klasifikasi *Naïve Bayes*, dan menampilkan hasil akhir jenis irama *qiro'ah*. Untuk lebih detail mengenai fitur aplikasi yang akan dibangun yang dijelaskan menggunakan *Use Case Diagram* sebagai berikut:



Gambar 3.3 Uce Case Diagram Fitur Aplikasi

Pada Gambar 3.3 menjelaskan detail fitur aplikasi yang nantinya berfungsi untuk memudahkan *user* dalam menjalankan aplikasi. Adapun penjelasannya sebagai berikut:

1. Mendapatkan data Qiro'ah, pertama *user* dapat mengambil *file* rekaman yang telah disimpan didalam direktori, nantinya akan tampil berdasarkan nama *file* yang telah di input pada aplikasi.
2. Proses Klasifikasi *Naïve Bayes*, pada tahap ini *user* menerima sekumpulan nilai yang didapat dari proses perhitungan klasifikasi *naïve bayes* menggunakan data *training* yang nantinya terdapat nilai probabilitas dari hasil klasifikasi untuk data *testing*. Sehingga, akan muncul hasil *output* jenis irama.

3.3.2 Desain Sistem

Untuk menyusun aplikasi pengenalan jenis irama *qiro'ah*, maka diperlukan desain sistem untuk membuat penerapan aplikasi secara rinci dan teratur sesuai yang diinginkan. Berikut Gambar 3.4 merupakan *flowchart* sistem yang diterapkan pada penulisan ini.



Gambar 3.4 *Flowchart* Sistem

Sistem aplikasi penentuan jenis irama *qiro'ah* ini digunakan satu kali proses, sehingga proses klasifikasi hanya bisa dimasukan satu rekaman dengan format **.wav** untuk data *testing*. Jika ingin melakukan pengujian kembali, maka harus memasukan data rekaman baru.

3.4 Proses Klasifikasi

3.4.1 Proses Pengambilan Data *Training*

Berikut adalah alur proses pengambilan data *training* pengenalan jenis-jenis irama *qiro'ah*:



Gambar 3.5 Alur Proses Pengambilan Data Training

1. Baca *Audio* Rekaman*.wav

Pada proses ini rekaman akan digunakan sebagai data sinyal input untuk diolah dengan menggunakan bantuan program *NetBeansIDE*. Pada

NetBeansIDE data yang dapat dikenali sebagai sinyal suara adalah dengan format **wav*.

2. *Pre-Processing*

Pada tahap ini bertujuan untuk memilih irama yang akan diuji benar-benar siap untuk dilatih (*valid*). Seperti, pengecekan jumlah irama setiap jenisnya sudah sama satu dengan yang lainnya.

3. Ekstraksi Ciri

Pada tahap ini akan dilakukan secara prosedural dengan membuat *folder* dari data *training*.

4. Sederetan Nilai

Setelah Proses ekstraksi ciri, nantinya akan muncul sederetan angka atau nilai dari rekaman-rekaman yang tersimpan dalam *folder* dari data *training*. Nilai tersebut didapat dari ekstraksi ciri yang telah dilakukan.

5. Data Training

Selanjutnya, *output* dari proses awal hingga akhir akan menghasilkan sebuah sekumpulan data-data yang nantinya akan diolah menggunakan metode *Naïve Bayes Classifier*, dan untuk sebagai pembandingan pada saat melakukan uji coba / *testing*.

3.4.2 Data Rekaman *qiro'ah*

Data suara rekaman ini merupakan sekumpulan data set yang akan diolah. Nantinya, dari setiap ke tujuh jenis irama *qiro'ah* terdapat 30 data sebagai data *training*.

Tabel 3. 1 Jumlah Keseluruhan Data Training Rekaman

Jumlah Data Training Rekaman	
IRAMA	Jumlah
Bayyati	30
Hijaz	30
Shoba	30
Jiharka	30
Sikah	30
Rost	30
Nahawan	30
Total	210

Untuk data *testing*, hampir sama dengan proses data *training*. Tetapi, yang bedakan adalah 3 data sample suara yang mahir membawakan ke tujuh jenis irama *qiro'ah*. Sehingga, menghasilkan 21 data dari setiap jenis irama *qiro'ah* sebagai data *testing*.

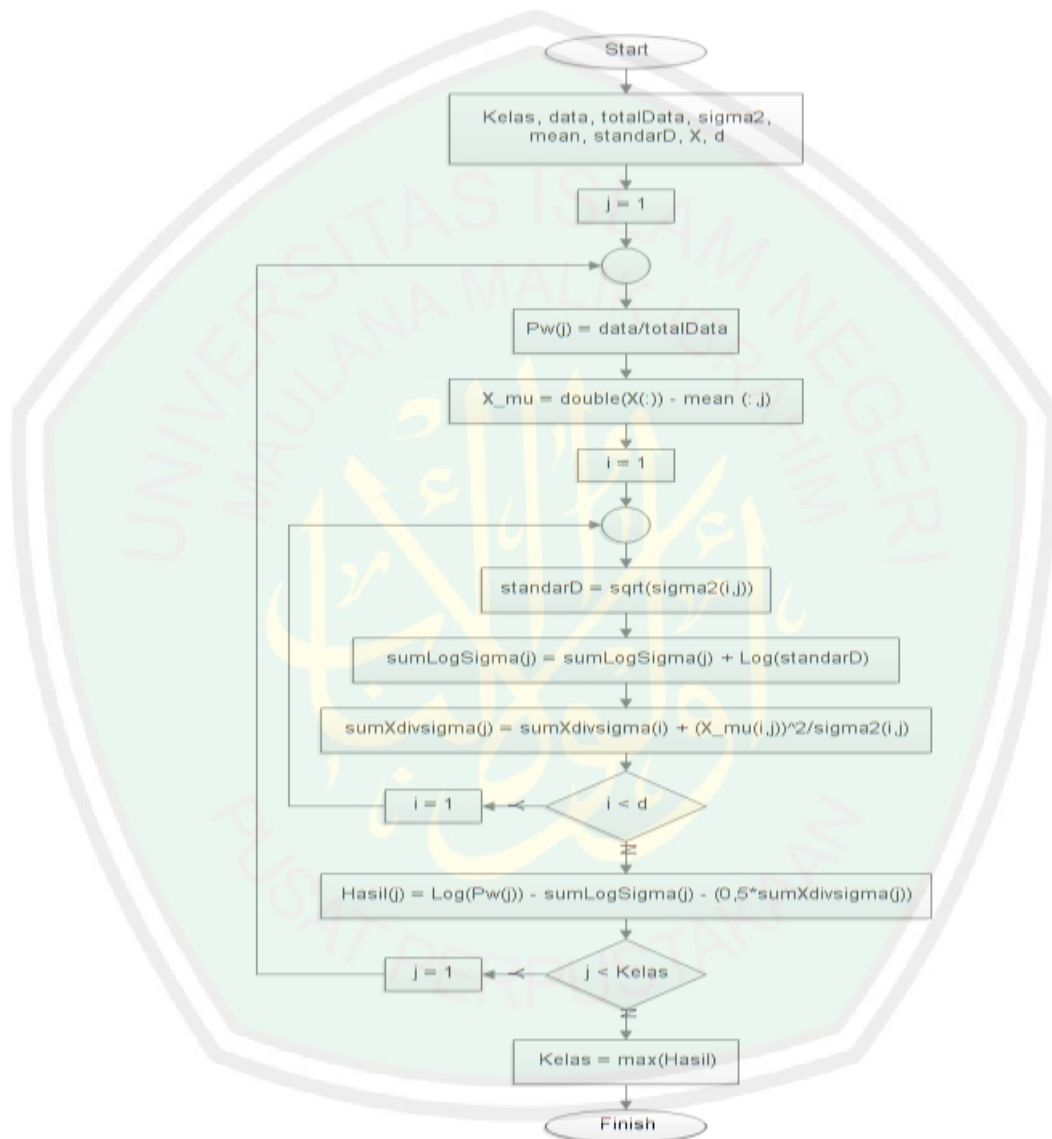
Tabel 3.2 Jumlah Keseluruhan Data Testing Rekaman

Jumlah Data Testing Rekaman	
IRAMA	Jumlah
Bayyati	9
Hijaz	9
Shoba	9
Jiharka	9
Sikah	9
Rost	9
Nahawan	9
Total	63

3.4.3 Flowchart Algoritma Naïve Bayes Classifier

Cara kerja metode *naïve bayes classifier* yaitu: menghitung probabilitas setiap kelas untuk dibandingkan, melakukan pengklasifikasian, menghitung peluang terbesar dengan menghitung peluang fitur perkelas, serta menghitung peluang terbesar dengan inputan *mean* dan *varian* data *training*. Fitur yang akan diuji adalah jumlah data setiap kelas dan jumlah data keseluruhan. Kelebihan metode *naïve bayes classifier* adalah sederhana, namun memiliki akurasi yang

baik hal tersebut terbukti dari beberapa penelitian terkait yang membahas mengenai pembuatan sistem dengan metode *naïve bayes classifier*. Adapun algoritma *Naïve Bayes Classifier* dalam bentuk *flowchart* ditunjukkan pada Gambar 3.6 sebagai berikut:



Gambar 3.6 Flowchart Algoritma Naïve Bayes Classifier

Adapun klasifikasi menggunakan metode *naive bayes classifier* yang harus dilakukan pertama adalah menginputkan fitur yang akan di uji coba, nilai *mean*, *varian*, serta data perkelas dan total data. Selanjutnya, menghitung peluang kelas

$\log(p(W_j))$ dengan cara menghitung data perkelas dibagi dengan total data kemudian hasil bagi di log-kan. Setelah itu peluang kelas selanjutnya menjumlahkan *varian* yang sudah diakar $\sum_{i=1}^d \log(\sigma_{ij})$ lalu di-log-kan.

Selanjutnya, hitung dengan rumus $\frac{1}{2} \sum_{j=1}^n \frac{(x_i - \mu_{ic})^2}{\sigma_{ic}}$ yaitu data fitur dikurangi *mean* kemudian dikuadratkan lalu dibagi dengan *varian* dan terakhir dikali dengan $\frac{1}{2}$.

Selanjutnya, $\log(p(W_j)) - \sum_{i=1}^d \log(\sigma_{ij}) - \frac{1}{2} \sum_{j=1}^n \frac{(x_i - \mu_{ic})^2}{\sigma_{ic}}$ yaitu hasil dari pehitungan peluang kelas dikurangi dengan hasil penjumlahan akar *varian* kemudian dikurangi lagi dengan hasil data fitur yang sudah dikurangi *mean* dan dibagi dengan *varian* dan dikali $\frac{1}{2}$. Kemudian setelah ditemukan hasilnya maka dicari nilai terbesar dari setiap kelas dengan menggunakan formula 4.

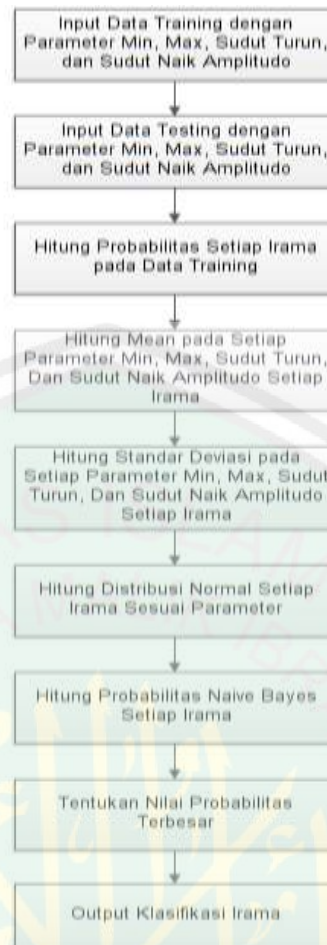
$$g_j X = \log(p(W_j)) - \sum_{i=1}^d \log(\sigma_{ij}) - \frac{1}{2} \sum_{j=1}^n \frac{(x_i - \mu_{ic})^2}{\sigma_{ic}} \dots (4)$$

Dimana W_j diketahui sebagai vektor fitur pada kelas j dengan parameter *mean* vektor σ_{ij} dan *covariance* Σ yang merupakan hasil estimasi berdasarkan data *training*. Selanjutnya, untuk melakukan proses klasifikasi terhadap pola data *testing* digunakan aturan pada formula 6 (Kusrini & E. T. Luthfi, 2009).

$$\hat{c} = \operatorname{argmax}_c (g_j(x)), j = 1, \dots, C \dots (5)$$

3.4.4 Perhitungan Naïve Bayes Classifier

Pada tahap ini peneliti mencoba menghitung secara manual metode *naïve bayes classifier*, dengan mempersiapkan data rekaman berupa nilai parameter yang digunakan sebagai bahan untuk melakukan proses klasifikasi manual. Adapun bentuk diagram blok perhitungan *Naïve Bayes Classifier* ditunjukkan pada Gambar 3.7 sebagai berikut:



Gambar 3.7 Diagram Blok Perhitungan Naïve Bayes Classifier

Data *training* rekaman jenis irama *qiro'ah* beserta informasi irama, nilai amplitudo minimal, maksimal, sudut turun, dan sudut naik amplitudo. Terlihat pada Tabel 3.3 dengan menggunakan 70 data rekaman untuk melakukan proses perhitungan *Naïve Bayes Classifier* sebagai berikut:

Tabel 3.3 Data Training Rekaman dan Parameter Perhitungan Naïve Bayes

DATA TRAINING NAÏVE BAYES					
No	IRAMA	MIN AMPLITUDO	MAX AMPLITUDO	SUDUT TURUN	SUDUT NAIK
1	Bayyati	73	1049265	3091	3105
2	Bayyati	24	850624	3801	3567
3	Bayyati	92	1024536	4010	3993
4	Bayyati	2	891360	3147	3342
5	Bayyati	32	928940	3631	3726
6	Bayyati	41	922432	3594	3711
...		..			
70	Sikah	53	871807	3752	4027

Tabel 3.4 Jumlah 70 Data Training

Irama	Jumlah Training
Bayyati	10
Hijaz	10
Shoba	10
Jiharka	10
Sikah	10
Rost	10
Nahawan	10
Total	70

Setelah mendapatkan objek yang diharapkan pada penelitian ini, maka proses pertama yang dilakukan; yaitu memasukan data rekaman untuk data *training* jenis irama *qiro'ah* maka dilakukan proses ekstraksi fitur. Pada tahap ekstraksi fitur jenis-jenis irama *qiro'ah* untuk mencari nilai minimal, maksimal, sudut turun, dan sudut naik amplitudo.

Tabel 3.5 Contoh Pengambilan Data Testing untuk Perhitungan Naïve Bayes

DATA TESTING	
MIN AMPLITUDO	30
MAX AMPLITUDO	922001
SUDUT TURUN	3631
SUDUT NAIK	3726

Kemudian, jika data sudah didapatkan atau yang sudah terekstraksi fiturnya, maka di lakukan proses perhitungan untuk mencari nilai rata-rata dari setiap parameter dan menggunakan formula 6.

$$\mu = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n} \dots (6)$$

Dimana:

μ = Mean / Rata-rata

$x_1, x_2, \dots, x_n$ = Nilai sampel ke-i

n = Jumlah sampel

Tabel 3.6 Perhitungan Rata-Rata dari Setiap Atribut Jenis Irama

Hitung Mean dari Setiap Atribut Jenis Irama				
Irama	Mean Min Amp	Mean Max Amp	Mean Sudut Turun	Mean Sudut Naik
Bayyati	40,90	925758,10	3568,60	3616,90
Hijaz	55,00	884971,00	3969,30	4045,10
Shoba	45,90	869145,40	3557,70	3658,90
Jiharka	45,90	869145,40	3557,70	3658,90
Sikah	47,40	865921,00	3427,60	3541,80
Rost	56,30	863161,60	4100,20	4186,40
Nahawan	52,70	845579,60	3388,20	3515,80

Kemudian, dilanjutkan kembali pada proses penilaian *standart deviasi*.

Untuk mengukur besarnya perbedaan dari setiap nilai sampel terhadap rata-rata dengan formula 7.

$$\sigma = \sqrt{\frac{\sum x_i^2 (x_i - \mu)^2}{n-1}} \dots (7)$$

Dimana:

σ = Standar Deviasi

x_i = Nilai sampel ke- i

μ = Mean / Rata-rata

n = Jumlah sampel keseluruhan

Tabel 3.7 Perhitungan Standart Deviasi Setiap Atribut Jenis Irama

Hitung Standart Deviasi dari Setiap Atribut Jenis Irama				
Irama	Mean Min Amp	Mean Max Amp	Mean Sudut Turun	Mean Sudut Naik
Bayyati	25,91	63009,34	405,92	413,09
Hijaz	28,27	48493,04	506,32	462,97
Shoba	26,72	63211,03	434,24	519,45
Jiharka	23,20	56313,38	564,45	592,20
Sikah	16,06	35123,93	522,09	609,58
Rost	24,21	85839,65	551,35	521,89
Nahawan	36,96	47392,90	413,38	500,98

Jika, hasil perhitungan standar deviasi sudah didapatkan. Selanjutnya, hitung probabilitas atau kemungkinan setiap jenis irama *qiro'ah* (Bayyati dengan tingkatan suara *Jawab*, *Hijaz* dengan variasi *Kard*, *Shoba* dengan variasi *Asli*, *Rost* dengan tingkatan suara *Jawab*, *Jiharkah* dengan variasi *Asli*, *Sikah* dengan variasi *Asli*, *Nahawan* dengan variasi *Asli*) dengan rumus:

$$\text{Probabilitas} = \frac{\text{Total data dari setiap jenis irama Qiro'ah}}{\text{Total data jenis irama Qiro'ah keseluruhan}}$$

Tabel 3.8 Probabilitas Naïve Bayes Setiap Jenis Irama

PROBABILITAS NAÏVE BAYES	
Irama	Probabilitas
Bayyati	0,0016572272
Hijaz	0,0009834997
Shoba	0,0156821052
Jiharka	0,0154980133
Sikah	0,0060420489
Rost	0,0006058364
Nahawan	0,0177449557

Tahap selanjutnya, yaitu melakukan proses *densitas gasuss* atau yang disebut juga dengan distribusi normal adalah untuk mengetahui probabilitas yang paling banyak digunakan dalam penentuan sampel. Formula 8 adalah distribusi normal sebagai berikut:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp \frac{-(x-\mu)^2}{2\sigma^2} \dots (8)$$

Dimana:

x = Nilai *variable input*

μ = *Mean / Rata-rata*

π = 3,14 atau $\frac{22}{7}$

σ = Standar Deviasi / *varian*

$\exp$ = 2,71828183

Tabel 3.9 Perhitungan Distribusi Normal dari Setiap Atribut Jenis Irama

Hitung Distribusi Normal dari Setiap Atribut Jenis Irama				
Irama	Mean Min Amp	Mean Max Amp	Mean Sudut Turun	Mean Sudut Naik
Bayyati	0,34	0,48	0,56	0,60
Hijaz	0,19	0,78	0,25	0,25
Shoba	0,28	0,80	0,57	0,55
Jiharka	0,25	0,83	0,55	0,55
Sikah	0,14	0,94	0,65	0,62
Rost	0,14	0,75	0,20	0,19
Nahawan	0,27	0,95	0,72	0,66

Setelah memperoleh nilai distribusi normal menggunakan formula 8 di atas, maka proses selanjutnya yaitu mengolah kembali nilai-nilai tersebut untuk dihitung menggunakan persamaan *Likelihood* agar dapat mengukur seberapa sering parameter muncul.

$$P(E | X_i) = (X_j * X_i) * Probabilitas \dots (9)$$

Dimana :

$$P(E | X_i) = \text{Likelihood}$$

$$X_j = \text{Nilai sampel ke- } j$$

$$X_i = \text{Nilai sampel ke- } i$$

$$prob = \text{Nilai probabilitas}$$

Hasil akhir klasifikasi jenis irama dari perhitungan diatas adalah Nahawan: 0,01774495572238320000 sebagai nilai terbesar dari jenis irama yang lain.

3.4.5 Proses Pengecekan *Similarity*

Kemudian, setelah semua perhitungan metode *Naïve Bayes Classifier* dilakukan, selanjutnya menambahkan tahap pengecekan *similarity* dengan menggunakan formula 10:

$$\text{Similarity} = \sum \text{Amplitudo } a / \sum \text{Amplitudo } i \dots (10)$$

Dimana:

$\sum Amplitudo a$ = Rata-rata setiap parameter pada data *testing*

$\sum Amplitudo i$ = Rata-rata keseluruhan data *training*

Setelah mendapatkan hasil dari *similarity*, selanjutnya tahap terakhir menjumlahkan hasil dari *Naïve Bayes* dengan hasil *similarity* untuk mendapatkan nilai akhir total probabiliti.

$$Total Probability = Hasil Naïve Bayes + Hasil Similarity$$

3.5 Perhitungan Akurasi

Confusion matrix adalah suatu metode yang digunakan untuk melakukan perhitungan akurasi pada konsep data mining. Evaluasi dengan *confusion matrix* menghasilkan nilai akurasi, presisi dan *recall*. Akurasi dalam klasifikasi adalah persentase ketepatan record data yang diklasifikasikan secara benar setelah dilakukan pengujian pada hasil klasifikasi (Han & Kamber, 2006). Presisi atau *confidence* adalah proporsi kasus yang diprediksi positif yang juga positif benar pada data yang sebenarnya. *Recall* atau *sensitivity* adalah proporsi kasus positif yang sebenarnya yang diprediksi positif secara benar (Powers, 2011).

Tabel 3.10 Model Confusion Matrix

<i>Confusion Matrix</i>		Nilai Sebenarnya	
		<i>True</i>	<i>False</i>
Nilai	<i>True</i>	TP (<i>True Positive</i>)	FP (<i>False Positive</i>)
Prediksi	<i>False</i>	FN (<i>False Negative</i>)	TN (<i>True Negative</i>)

Rumus untuk menghitung Akurasi, presisi dan *recall* pada *confusion matrix* adalah sebagai berikut (Gorunescu, 2011):

$$Prescision = \frac{TP}{FP+TP} \times 100\%$$

Presisi adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$$

Akurasi didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual.

$$Recall = \frac{TP}{FP+TP} \times 100\%$$

Recall adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi.

3.6 Desain Interface

Sebelum membangun aplikasi pengenalan jenis irama *qiro'ah* yang baik, bagus sertanya dapat memberikan kenyamanan baik secara interaksi aplikasi terhadap *user*, maka perlu dibuat desain *interface* terlebih dahulu. Berikut pada Gambar 3.7 adalah desain *interface* aplikasi pengenalan jenis irama *qiro'ah* yang akan dibangun dengan menggunakan *software* pendukung yaitu, *NetBeansIDE*.

Aplikasi Pengenalan Jenis-Jenis Irama Qiro'ah
Menggunakan Metode Naive Bayes Classifier

Keterangan *wav

Frekuensi
Maksimum
Minimum

Amplitudo
Maksimum
Minimum

Input Rekaman *wav

Hasil Klasifikasi Naive Bayes Classifier

Frekuensi	Amplitudo
Bayati	
Hijaz	
Shoba	
Rost	
Nahawan	
Jiharkah	
Sikah	

Hasil Akhir ==>

Gambar 3.8 Desain Interface Aplikasi Pengenalan Jenis Irama qiro'ah

BAB IV

UJI COBA DAN PEMBAHASAN

Pada bab ini akan dibahas tentang bagaimana ujicoba metode yang digunakan dan analisa dalam membangun aplikasi yang sudah dibuat serta penerapan atau implementasi metode *Naive Bayes Classifier*. Adapaun tujuan yang dilakukan pengujian ini adalah untuk mengetahui apakah metode yang dipakai cocok dan dapat digunakan dalam aplikasi penentuan jenis-jenis irama *qiro'ah*. Sehingga, dengan dilakukannya penelitian menggunakan metode *Naive Bayes Classifier* dapat diketahui kekurangannya dan juga agar dapat dikembangkan kembali dengan metode lainnya yang sesuai dengan kelebihan masing-masing. Supaya aplikasi yang dirasa kurang akan menjadi lebih inovatif dan revolusioner.

4.1 Peralatan yang digunakan

Sebelum membahas implementasi program, terlebih dahulu dijelaskan spesifikasi *hardware* dan *software* yang telah digunakan adalah:

4.1.1 *Hardware*

Adapun perangkat keras (*hardware*) yang digunakan untuk pengerjaan aplikasi dan pengujian metode adalah sebagai berikut:

1. *Processor* : *Processor Intel Inside*
2. *Memory* : *RAM 4 GB-System*
3. *System Type* : *32bit Operation System*
4. *Alat Perekam* : *Microfon BM800 + Stand Arm Condenser*

4.1.2 Software

Sedangkan, perangkat lunak (*software*) yang digunakan untuk pengerjaan aplikasi dan pengujian metode adalah:

1. *Operating System* : *Windows 10*
2. *Penulisan* : *Microsoft Word 2010*
3. *Pengembangan Aplikasi* : *NetBeans IDE 8.1*

4.2 User Interface

Tampilan (*interface*) aplikasi pengenalan jenis-jenis irama *qiro'ah* ini terbilang cukup sederhana, pada Gambar 4.1 terdapat 4 tombol *button* yaitu: Pilih Audio, Putar Audio, Proses Klasifikasi, dan Reset. Berikut pada Gambar 4.1 adalah tampilan (*Interface*) dari aplikasi penentuan jenis-jenis irama *qiro'ah*.



Gambar 4.1 Running Aplikasi dan Tampilan Utama

Aplikasi ini terdapat dua fungsi utama, yaitu fungsi tombol pilih *audio* (*getdata*) dan fungsi tombol proses (*data analysis*). Fungsi pilih *audio* tersebut mempunyai 2 fungsi sekaligus, pertama berfungsi untuk data *testing* yaitu rekaman *audio* yang berekstensi **wav* yang terdapat pada Gambar 4.2.

```

// Wavfile baru dan inisialisasi
    WavFile wavFile = new WavFile();
    wavFile.file = file;
    wavFile.numChannels = numChannels;
    wavFile.numFrames = numFrames;
    wavFile.sampleRate = sampleRate;
    wavFile.bytesPerSample = (validBits + 7) / 8;
    wavFile.blockAlign = wavFile.bytesPerSample * numChannels;
    wavFile.validBits = validBits;

// Sanity check arguments
    if (numChannels < 1 || numChannels > 65535) throw new
WavFileException("Illegal number of channels, valid range 1 to 65536");
    if (numFrames < 0) throw new WavFileException("Number of
frames must be positive");
    if (validBits < 2 || validBits > 65535) throw new
WavFileException("Illegal number of valid bits, valid range 2 to 65536");
    if (sampleRate < 0) throw new WavFileException("Sample rate
must be positive");

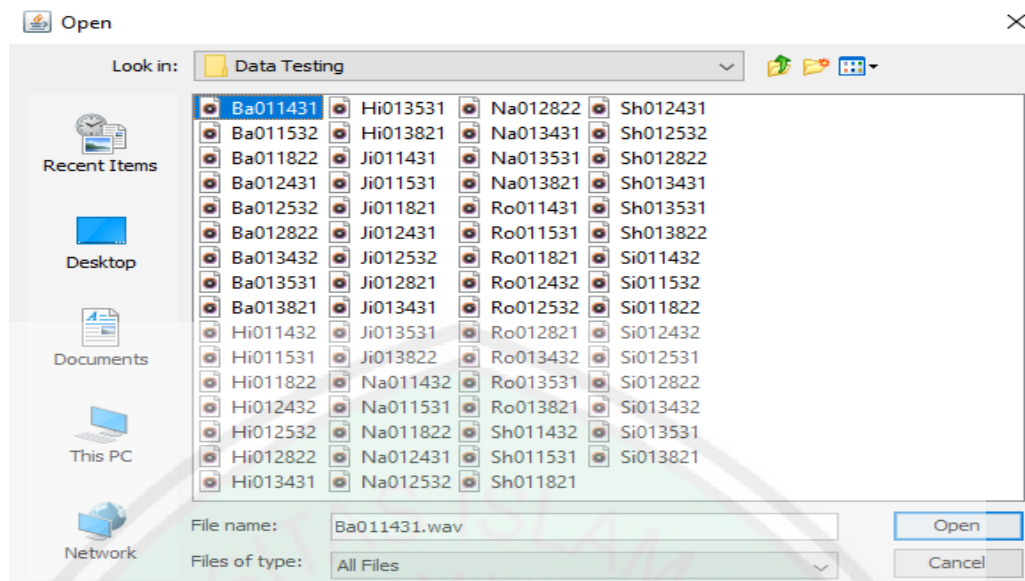
// Create output stream for writing data
    wavFile.oStream = new FileOutputStream(file);

File filenya;
JFileChooser chooser = new JFileChooser(".");
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    int buka_dialog = chooser.showOpenDialog(mainUtama.this);
    if (buka_dialog == JFileChooser.APPROVE_OPTION) {

// ketika tombol audio ditekan, maka user memilih audio, sistem akan cek
apakah file nya benar2 wav atau tidak.
        filenya = chooser.getSelectedFile();
        String a =
filenya.getName().substring(filenya.getName().length() - 3,
filenya.getName().length());
        try{
            if (a.equals("wav")) {
                url.setText("Audio File : " + filenya.getName());
            } else {
                JOptionPane.showMessageDialog(null, "Bukan WAV File");
            }
        } catch (Exception e){
            System.out.println(e);
        }
    }
}

//path lokasi direktori image bakal ditampilkan di textfield
DecimalFormat numberFormat = new DecimalFormat("#.###");

```

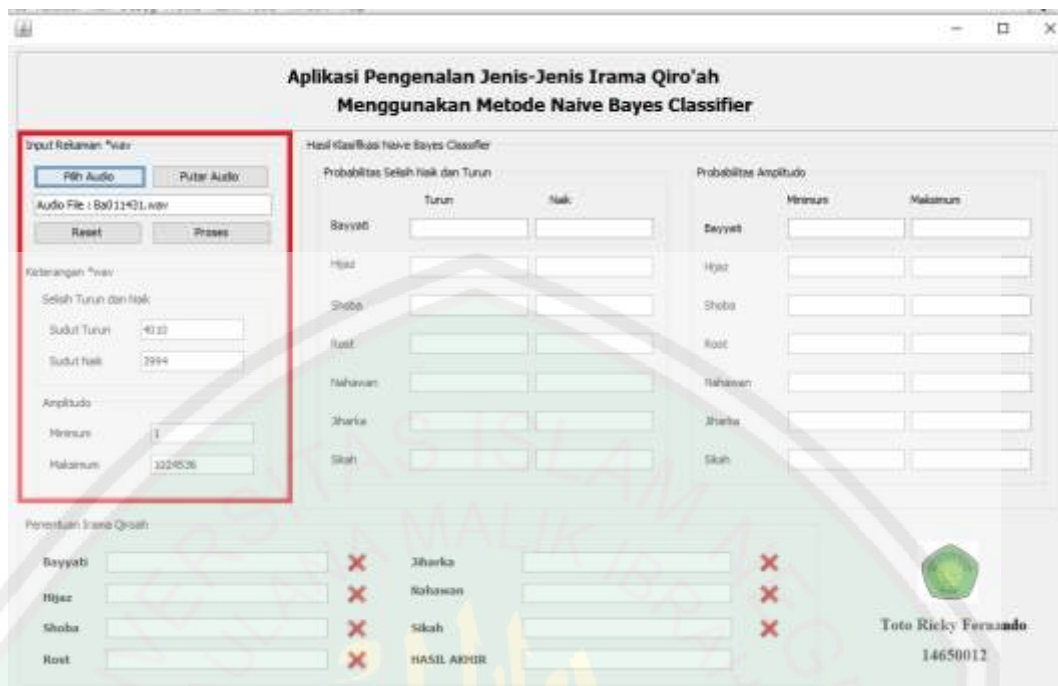


Gambar 4.2 Tampilan untuk Mengambil Audio

Selanjutnya, fungsi kedua yaitu data *testing* diambil namanya akan tertulis otomatis di *textfield* sebagai nama *file* dan parameter yang digunakan juga otomatis akan muncul nilai-nilainya pada *textfield* di class ***getAmplitudo.java*** dan ***getNaikTurun.java***. Kemudian, terdapat fitur tombol putar *audio* berfungsi untuk memutar *audio.wav* yang sudah dipilih dan fitur tombol reset berfungsi untuk mereset atau menghilangkan tampilan sebelumnya menjadi seperti semula. Tampilan lebih detail terdapat pada Gambar 4.3.

```
amplitudeAvg = (int) (jumlahTot / wav.getFramesCount());
System.out.println("count : " + wav.getFramesCount());
System.out.println("amplitude max : " + amplitudeMax);
System.out.println("amplitude min : " + amplitudeMin);
```

```
sudutNaik = Math.abs(jumlahArrNaik/panjangArrNaik);
int jumlahArrTurun = 0;
int panjangArrTurun = 0;
for(int a =0; a<arr_turun.length;a++){
    if(arr_turun[a]!=0){
        jumlahArrTurun = jumlahArrTurun + arr_turun[a];
        panjangArrTurun++; }
}
sudutTurun = Math.abs(jumlahArrTurun/panjangArrTurun);
System.out.println("panjang " + panjang);
System.out.println("panjang naik : " + panjangArrNaik);
System.out.println("panjang turun : " + panjangArrTurun);
System.out.println("rata2 naik : " + sudutTurun);
System.out.println("rata2 turun : "+ sudutNaik);
Arrays.sort(arr);
```



Gambar 4.3 Tampilan Aplikasi setelah Memilih Audio.wav

Selanjutnya, pada fungsi proses atau fungsi tombol proses, didalamnya terdapat fungsi set probabilitas jenis-jenis irama *qiro'ah* Bayyati, Hijaz, Shoba, Rost, Jiharka, Nahawan, dan Sikah pada bagian *textfield* di class *NaiveBayes.java* terdapat pada Gambar 4.4.

```
System.out.println("===== PROBABILITAS KELAS =====");

int jumlahBayyati = hitungJumlah(Irama, B);
int jumlahHijaz = hitungJumlah(Irama, H);
int jumlahRost = hitungJumlah(Irama, R);
int jumlahJiharka = hitungJumlah(Irama, J);
int jumlahNahawan = hitungJumlah(Irama, N);
int jumlahShoba = hitungJumlah(Irama, Sh);
int jumlahSikah = hitungJumlah(Irama, Si);

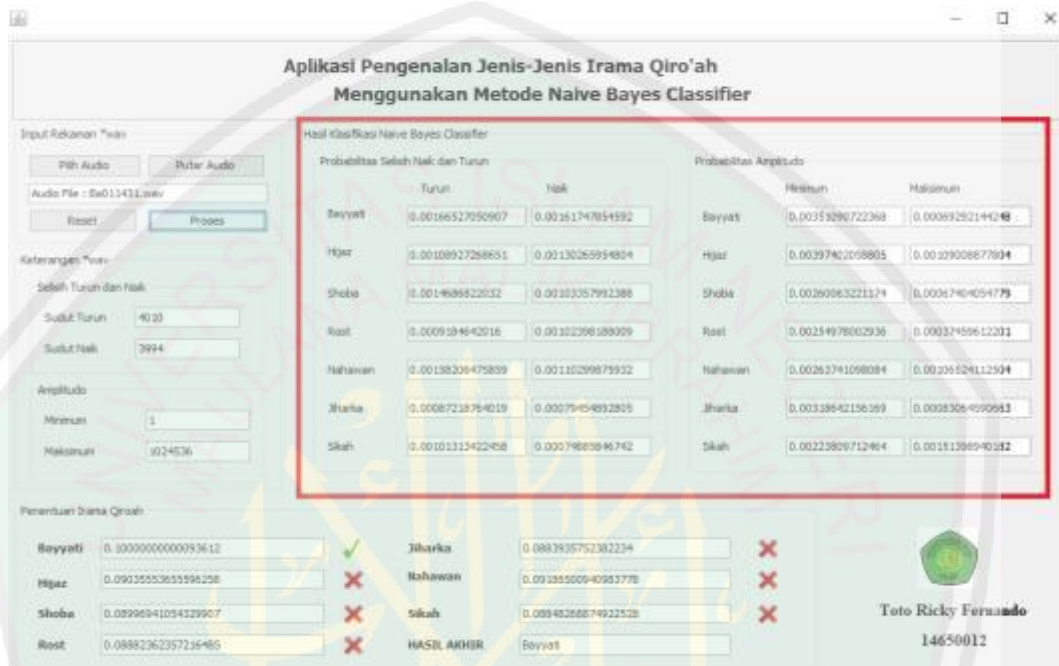
System.out.println("Jumlah Irama Bayyati : " + jumlahBayyati);
System.out.println("Jumlah Irama Hijaz : " + jumlahHijaz);
System.out.println("Jumlah Irama Rost : " + jumlahRost);
System.out.println("Jumlah Irama Jiharka : " + jumlahJiharka);
System.out.println("Jumlah Irama Nahawan : " + jumlahNahawan);
System.out.println("Jumlah Irama Shoba : " + jumlahShoba);
System.out.println("Jumlah Irama Sikah : " + jumlahSikah);

P_Bayyati = (P_irama(jumlahBayyati, jumlahHijaz, jumlahRost,
jumlahJiharka, jumlahNahawan, jumlahShoba, jumlahSikah));
P_Hijaz = P_irama(jumlahHijaz, jumlahBayyati, jumlahRost,
jumlahJiharka, jumlahNahawan, jumlahShoba, jumlahSikah);
P_Rost = P_irama(jumlahRost, jumlahHijaz, jumlahBayyati,
```

```

jumlahJiharka, jumlahNahawan, jumlahShoba, jumlahSikah);
    P_Jiharka = P_irama(jumlahJiharka, jumlahHijaz, jumlahRost,
jumlahBayyati, jumlahNahawan, jumlahShoba, jumlahSikah);
    P_Nahawan = P_irama(jumlahNahawan, jumlahHijaz, jumlahRost,
jumlahJiharka, jumlahBayyati, jumlahShoba, jumlahSikah);
    P_Shoba = P_irama(jumlahShoba, jumlahHijaz, jumlahRost,
jumlahJiharka, jumlahNahawan, jumlahBayyati, jumlahSikah);
    P_Sikah = P_irama(jumlahSikah, jumlahHijaz, jumlahRost,
jumlahJiharka, jumlahNahawan, jumlahShoba, jumlahBayyati);

```



Gambar 4.4 Tampilan Probabilitas Data Testing

Kemudian, terdapat proses hasil analisa yang di mana nantinya irama yang menjadi data *testing* termasuk irama, *qiro'ah* Bayyati, Hijaz, Shoba, Rost, Jiharka, Nahawan, atau Sikah dengan cara perbandingan hasil proses probabilitas yang sudah dihitung terdapat pada Gambar 4.5.



Gambar 4.5 Tampilan Hasil Akhir Penentuan Jenis-Jenis Irama qiro'ah

Serta terdapat fungsi perhitungan dengan algoritma metode *Naïve Bayes Classifier* tentu yang nantinya akan dibahas pada sub bab pembahasan.

4.3 Pembahasan

Pada sub bab ini akan menjelaskan beberapa tahapan - tahapan prosedur dari algoritma *Naïve Bayes Classifier* sebagai berikut: tahap pertama yaitu, menghitung nilai masing-masing parameter maksimum dan minimum gelombang amplitudo dan menghitung nilai selisih sudut naik dan sudut turun pada gelombang amplitudo untuk seluruh data *training* yang totalnya 210 sampel data.

```
// Array untuk menampung data training
static String[] Irama = new String[210];
static double[] AmplitudoMin = new double[210];
static double[] AmplitudoMax = new double[210];
static double[] NaikAvg = new double[210];
static double[] TurunAvg = new double[210];
```

Nantinya, data yang sudah dihasilkan akan dimasukan kedalam *variable array* maksimum minimum gelombang amplitudo dan selisih sudut naik turun gelombang amplitudo data rekaman *qiro'ah*.

Bayyati	30.0	1049.265	309.1	310.5
Bayyati	94.0	850.624	380.1	356.7
Bayyati	92.0	1024.536	401.0	399.3
Bayyati	81.0	891.36	314.7	334.2
Bayyati	56.0	928.94	363.1	372.6

Sikah	100.0	1304.786	319.9	320.4
Sikah	95.0	1219.868	332.9	347.3
PRINT MAP				
Am Min : 210				

Gambar 4.6 Tampilan Nilai-Nilai pada Data Training

Dari nilai-nilai data *training* pada Gambar 4.6 nantinya akan disatukan data rekaman *qiro'ah* sesuai dengan jenis iramanya. Ambil data pada *excel* yang bernama “*data.xls*”.

Tabel 4.1 Pendataan Data Training Rekaman Qiro'ah Berjumlah 210 Sampel

No.	Jenis Irama	Nama File. *wav	Amplitudo Min	Amplitudo Max	Sudut Turun / 10	Sudut Naik / 10	Sudut Turun	Sudut Naik	Amplitudo Max * 1000
1	Bayyati	Ba001433.wav	78	1049,265	309,1	310,5	3091	3105	1049265
2	Bayyati	Ba002432.wav	83	850,624	380,1	356,7	3801	3567	850624
3	Bayyati	Ba003431.wav	4	1024,536	401	399,3	4010	3993	1024536
4	Bayyati	Ba004432.wav	15	891,36	314,7	334,2	3147	3342	891360
5	Bayyati	Ba005432.wav	99	928,94	363,1	372,6	3631	3726	928940

No.	Jenis Irama	Nama File. *wav	Amplitudo Min	Amplitudo Max	Sudut Turun / 10	Sudut Naik / 10	Sudut Turun	Sudut Naik	Amplitudo Max * 1000
6	Bayyati	Ba006431.wav	100	922,432	359,4	371,1	3594	3711	922432
7	Bayyati	Ba007432.wav	87	858,949	306,7	317,7	3067	3177	858949
...									
209	Sikah	Si009822.wav	71	1304,786	319,9	320,4	3199	3204	1304786
210	Sikah	Si010822.wav	2	1219,868	332,9	347,3	3329	3473	1219868

Pengambilan data tersebut nantinya akan dibagi dan dimasukan *variable* jenis irama *qiro'ah*.

```
readFromExcel("D://data.xls");
System.out.println("PRINT MAP");
System.out.println("Am Min : " + Amplitudo_Min.size());
konvert2arrString(Irama_Irama, Irama);
konvert2arrDouble(Amplitudo_Min, AmplitudoMin);
konvert2arrDouble(Amplitudo_Max, AmplitudoMax);
konvert2arrDouble(Turun_Avg, TurunAvg);
konvert2arrDouble(Naik_Avg, NaikAvg);
```

Data sampel yang ada pada *file* NetBeans masih dalam kondisi tercampur antara jenis irama Bayyati, Hijaz, Shoba, Nahawan, Rost, Jiharkah, dan Sikah, maka selajutnya dilakukan penyaringan nilai AmplitudoMin, AmplitudoMax, SudutTurun, dan SudutNaik sesuai dengan jenis irama *qiro'ah*.

```
static double hasilBayyati = 0;
static double hasilHijaz = 0;
static double hasilShoba = 0;
static double hasilJiharka = 0;
static double hasilNahawan = 0;
static double hasilRost = 0;
static double hasilSikah = 0;

static String hasilAkhir = " ";

static Map<Integer, String> Irama_Irama = new HashMap<>();
static Map<Integer, Double> Amplitudo_Min = new HashMap<>();
static Map<Integer, Double> Amplitudo_Max = new HashMap<>();

static Map<Integer, Double> Turun_Avg = new HashMap<>();
static Map<Integer, Double> Naik_Avg = new HashMap<>();

static double maxValue = 0;
static String B = "Bayyati";
static String H = "Hijaz";
static String R = "Rost";
static String J = "Jiharka";
static String N = "Nahawan";
static String Sh = "Shoba";
static String Si = "Sikah";
```

Hasil *source code* di atas disimpan dalam *variable*:

- Amp_MinBayyati, Amp_MaxBayyati, Turun_AvgBayyati, Naik_AvgBayyati
- Amp_MinHijaz, Amp_MaxHijaz, Turun_AvgHijaz, Naik_AvgHijaz
- Amp_MinJiharkah, Amp_MaxJiharkah, Turun_AvgJiharkah, Naik_AvgJiharkah
- Amp_MinNahawan, Amp_MaxNahawan, Turun_AvgNahawan, Naik_AvgNahawan
- Amp_MinRost, Amp_MaxRost, Turun_AvgRost, Naik_AvgRost
- Amp_MinShoba, Amp_MaxShoba, Turun_AvgShoba, Naik_AvgShoba
- Amp_MinSikah, Amp_MaxSikah, Turun_AvgSikah, Naik_AvgSikah

===== DATA TRAINING =====				
12.0	927.119	336.7	342.2	Bayyati
18.0	935.822	357.5	352.7	Bayyati
72.0	876.828	373.1	381.6	Hijaz
4.0	888.642	467.6	436.8	Hijaz
66.0	925.725	446.6	443.3	Hijaz
37.0	869.812	326.5	348.2	Hijaz

18.0	1246.044	319.1	324.1	Shoba
31.0	1215.202	268.0	294.7	Shoba
16.0	1179.975	274.4	271.8	Sikah

Gambar 4.7 Running Nilai AmplitudoMin, AmplitudoMax, TurunAvg, dan NaikAvg Berdasarkan Jenis Irama qiro'ah

Selanjutnya, sesuai dengan prosedur metode *Naïve Bayes Classifier* dilakukan perhitungan rata-rata dari setiap jenis irama *qiro'ah* yang digunakan sebagai data sampel.

```
public static double hitungMean(double[] atribut) {
    double avg = 0;
    double jumlah = 0;
    for (int a = 0; a < atribut.length; a++) {
        jumlah = jumlah + atribut[a];
    }
    avg = jumlah / atribut.length;
    return avg;
}
```

Setelah di *run* pada aplikasi *NetBeans* menemui hasil Gambar 4.8 adalah sebagai berikut:

```

run:
count: 935822
amplitude max: 935822
amplitude min: 1
rata-rata: 780
Panjang: 935822
panjang naik: 469540
panjang turun: 463206
rata2 naik: 3575
rata2 turun: 3527

```

Gambar 4.8 Output Perhitungan Rata-Rata dari Setiap Jenis Irama *qiro'ah*

Selanjutnya, menghitung nilai Standar Deviasi dengan formula 11 sebagai berikut:

$$S = \sqrt{\frac{\sum y^2 - \frac{(\sum y)^2}{n}}{n-1}} \dots (11)$$

Perhitungan tersebut dapat dijabarkan menjadi beberapa tahapan, yaitu:

1. Kurangi nilai maksimal amplitudo jenis irama *qiro'ah* tiap data dengan nilai mean dari maksimal amplitudo jenis irama *qiro'ah*.
2. Dari hasil tersebut, lakukan perkalian pangkat 2.
3. Jumlahkan ke semua nilai (210) yang sudah di hitung dan dibagi dengan nilai panjang data dikurangi 1.
4. Lakukan fungsi akar kuadrat terhadap nilai tersebut, hasil tersebut dapat dinamakan dengan nilai standart deviasi.

Lakukan juga terhadap parameter jenis irama *qiro'ah* nilai amplitudo minimal Bayyati, nilai amplitudo maksimal Bayyati, nilai sudut turun Bayyati, dan nilai sudut naik Bayyati dan jenis irama *qiro'ah* lainnya.

```

public static double hitungSDeviasi(double[] atribut, double mean) {
    double avg = mean;
    double sDeviasi = 0;
    double jumlah = 0;
    for (int i = 0; i < atribut.length; i++) {
        atribut[i] = Math.pow((atribut[i] - avg), 2);
    }
}

```

```

        jumlah = jumlah + atribut[i];
    }
    sDeviasi = jumlah / atribut.length;
    return sDeviasi;
}

//sDeviasi AmplitudoMin

double sDeviasi_BayyatiAmplitudoMin =
hitungProbabilitasAtribut_sDeviasi(AmplitudoMin, Irama, B,
Mean_BayyatiAmplitudoMin);
    double sDeviasi_HijazAmplitudoMin =
hitungProbabilitasAtribut_sDeviasi(AmplitudoMin, Irama, H,
Mean_HijazAmplitudoMin);
    double sDeviasi_ShobaAmplitudoMin =
hitungProbabilitasAtribut_sDeviasi(AmplitudoMin, Irama, Sh,
Mean_ShobaAmplitudoMin);
    double sDeviasi_SikahAmplitudoMin =
hitungProbabilitasAtribut_sDeviasi(AmplitudoMin, Irama, Si,
Mean_SikahAmplitudoMin);
    double sDeviasi_JiharkaAmplitudoMin =
hitungProbabilitasAtribut_sDeviasi(AmplitudoMin, Irama, J,
Mean_JikarkaAmplitudoMin);
    double sDeviasi_RostAmplitudoMin =
hitungProbabilitasAtribut_sDeviasi(AmplitudoMin, Irama, R,
Mean_RostAmplitudoMin);
    double sDeviasi_NahawanAmplitudoMin =
hitungProbabilitasAtribut_sDeviasi(AmplitudoMin, Irama, N,
Mean_NahawanAmplitudoMin);

//sDeviasi AmplitudoMax

double sDeviasi_BayyatiAmplitudoMax =
hitungProbabilitasAtribut_sDeviasi(AmplitudoMax, Irama, B,
Mean_BayyatiAmplitudoMax);
    double sDeviasi_HijazAmplitudoMax =
hitungProbabilitasAtribut_sDeviasi(AmplitudoMax, Irama, H,
Mean_HijazAmplitudoMax);
    double sDeviasi_ShobaAmplitudoMax =
hitungProbabilitasAtribut_sDeviasi(AmplitudoMax, Irama, Sh,
Mean_ShobaAmplitudoMax);
    double sDeviasi_SikahAmplitudoMax =
hitungProbabilitasAtribut_sDeviasi(AmplitudoMax, Irama, Si,
Mean_SikahAmplitudoMax);
    double sDeviasi_JiharkaAmplitudoMax =
hitungProbabilitasAtribut_sDeviasi(AmplitudoMax, Irama, J,
Mean_JikarkaAmplitudoMax);
    double sDeviasi_RostAmplitudoMax =
hitungProbabilitasAtribut_sDeviasi(AmplitudoMax, Irama, R,
Mean_RostAmplitudoMax);
    double sDeviasi_NahawanAmplitudoMax =
hitungProbabilitasAtribut_sDeviasi(AmplitudoMax, Irama, N,
Mean_NahawanAmplitudoMax);

//sDeviasi Turun

double sDeviasi_Bayyati_T = hitungProbabilitasAtribut_sDeviasi(TurunAvg,
Irama, B, Mean_Bayyati_T);
    double sDeviasi_Hijaz_T =
hitungProbabilitasAtribut_sDeviasi(TurunAvg, Irama, H, Mean_Hijaz_T);
    double sDeviasi_Shoba_T =
hitungProbabilitasAtribut_sDeviasi(TurunAvg, Irama, Sh, Mean_Shoba_T);
    double sDeviasi_Sikah_T =
hitungProbabilitasAtribut_sDeviasi(TurunAvg, Irama, Si, Mean_Sikah_T);
    double sDeviasi_Jiharka_T =

```

```

hitungProbabilitasAtribut_sDeviasi(TurunAvg, Irama, J, Mean_Jikarka_T);
double sDeviasi_Rost_T =
hitungProbabilitasAtribut_sDeviasi(TurunAvg, Irama, R, Mean_Rost_T);
double sDeviasi_Nahawan_T =
hitungProbabilitasAtribut_sDeviasi(TurunAvg, Irama, N, Mean_Nahawan_T);

//sDeviasi Naik

double sDeviasi_Bayyati_N =
hitungProbabilitasAtribut_sDeviasi(NaikAvg, Irama, B, Mean_Bayyati_N);
double sDeviasi_Hijaz_N =
hitungProbabilitasAtribut_sDeviasi(NaikAvg, Irama, H, Mean_Hijaz_N);
double sDeviasi_Shoba_N =
hitungProbabilitasAtribut_sDeviasi(NaikAvg, Irama, Sh, Mean_Shoba_N);
double sDeviasi_Sikah_N =
hitungProbabilitasAtribut_sDeviasi(NaikAvg, Irama, Si, Mean_Sikah_N);
double sDeviasi_Jiharka_N =
hitungProbabilitasAtribut_sDeviasi(NaikAvg, Irama, J, Mean_Jikarka_N);
double sDeviasi_Rost_N =
hitungProbabilitasAtribut_sDeviasi(NaikAvg, Irama, R, Mean_Rost_N);
double sDeviasi_Nahawan_N =
hitungProbabilitasAtribut_sDeviasi(NaikAvg, Irama, N, Mean_Nahawan_N);

```

Selanjutnya, langsung menghitung nilai probabilitas tiap jenis irama *qiro'ah*.

```

public static double hitungProbabilitasAtribut_Mean(double[] atribut,
String[] Irama, String irama) {
double hitungMean = 0;
int i = 0;
double jumlah = 0;
for (int a = 0; a < Irama.length; a++) {
if (Irama[a].equals(irma)) {
jumlah = jumlah + atribut[a];
i++;
}
}
hitungMean = jumlah / i;
return hitungMean;
}

public static double hitungProbabilitasAtribut_sDeviasi(double[]
atribut, String[] Irama, String irama, double mean)
double avg = mean;
double sDeviasi = 0;
double jumlah = 0;
int i = 0;
for (int a = 0; a < Irama.length; a++) {
if (Irama[a].equals(irma)) {
atribut[a] = Math.pow((atribut[a] - avg), 2);
jumlah = jumlah + atribut[a];
i++;
}
}
sDeviasi = jumlah / atribut.length;
return sDeviasi;
}

P_Bayyati = (P_irama(jumlahBayyati, jumlahHijaz, jumlahRost,
jumlahJiharka, jumlahNahawan, jumlahShoba, jumlahSikah));
P_Hijaz = P_irama(jumlahHijaz, jumlahBayyati, jumlahRost,
jumlahJiharka, jumlahNahawan, jumlahShoba, jumlahSikah);
P_Rost = P_irama(jumlahRost, jumlahHijaz, jumlahBayyati,
jumlahJiharka, jumlahNahawan, jumlahShoba, jumlahSikah);
P_Jiharka = P_irama(jumlahJiharka, jumlahHijaz, jumlahRost,
jumlahBayyati, jumlahNahawan, jumlahShoba, jumlahSikah);
P_Nahawan = P_irama(jumlahNahawan, jumlahHijaz, jumlahRost,

```

```

jumlahJiharka, jumlahBayyati, jumlahShoba, jumlahSikah);
    P_Shoba = P_irama(jumlahShoba, jumlahHijaz, jumlahRost,
jumlahJiharka, jumlahNahawan, jumlahBayyati, jumlahSikah);
    P_Sikah = P_irama(jumlahSikah, jumlahHijaz, jumlahRost,
jumlahJiharka, jumlahNahawan, jumlahShoba, jumlahBayyati);

    System.out.println("Probabilitas Irama Bayyati: " + P_Bayyati);
    System.out.println("Probabilitas Irama Hijaz: " + P_Hijaz);
    System.out.println("Probabilitas Irama Rost: " + P_Rost);
    System.out.println("Probabilitas Irama Jiharka: " + P_Jiharka);
    System.out.println("Probabilitas Irama Nahawan: " + P_Nahawan);
    System.out.println("Probabilitas Irama Shoba: " + P_Shoba);
    System.out.println("Probabilitas Irama Sikah: " + P_Sikah);

    double Mean_BayyatiAmplitudoMin =
hitungProbabilitasAtribut_Mean(AmplitudoMin, Irama, B);
    double Mean_HijazAmplitudoMin =
hitungProbabilitasAtribut_Mean(AmplitudoMin, Irama, H);
    double Mean_ShobaAmplitudoMin =
hitungProbabilitasAtribut_Mean(AmplitudoMin, Irama, Sh);
    double Mean_SikahAmplitudoMin =
hitungProbabilitasAtribut_Mean(AmplitudoMin, Irama, Si);
    double Mean_JikarkaAmplitudoMin =
hitungProbabilitasAtribut_Mean(AmplitudoMin, Irama, J);
    double Mean_RostAmplitudoMin =
hitungProbabilitasAtribut_Mean(AmplitudoMin, Irama, R);
    double Mean_NahawanAmplitudoMin =
hitungProbabilitasAtribut_Mean(AmplitudoMin, Irama, N);

```

Dari hasil *source code* di atas terhadap perhitungan peluang *class-class* keseluruhan, maka dihasilkan kesamaan nilai sebagai berikut:

Probabilitas Irama Bayyati : 0.14285714285714285
 Probabilitas Irama Hijaz : 0.14285714285714285
 Probabilitas Irama Rost : 0.14285714285714285
 Probabilitas Irama Jiharka : 0.14285714285714285
 Probabilitas Irama Nahawan : 0.14285714285714285
 Probabilitas Irama Shoba : 0.14285714285714285
 Probabilitas Irama Sikah : 0.14285714285714285

Hasil peluang atau probabilitas jenis irama di atas sama dikarenakan jumlah data sampel pembelajaran sama yaitu 30 data sampel yang dibandingkan dengan data sampel keseluruhan yang berjumlah 210 data sampel. Langkah selanjutnya, yaitu pengambilan nilai amplitudo minimal, amplitudo maksimal,

sudut turun dan sudut naik dari data ujicoba atau data *testing* untuk dihitung dengan fungsi distribusi normal.

```
public static double hitungDistribusiNormal(double x, double mean,
double sDeviasi) {
    double hasil = 5;
    double phi = 3.14159;
    double e = 2.718;
    hasil = (1 / (sDeviasi * Math.sqrt(2 * phi))) * Math.pow(e, -
(Math.pow(x - mean, 2)) / (2 * Math.pow(sDeviasi, 2)));
    return hasil;
}
```

Perhitungan fungsi distribusi normal dilakukan pada formula 12 sebagai berikut:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma_j^i} \exp \frac{-(x-\mu_{ij})^2}{2\sigma_{ij}^2} \dots (12)$$

```
// DISTRIBUSI NORMAL
    BayyatiDisNormal_AmpMin = hitungDistribusiNormal(AmpMin,
Mean_BayyatiAmplitudoMin, sDeviasi_BayyatiAmplitudoMin);
    HijazDisNormal_AmpMin = hitungDistribusiNormal(AmpMin,
Mean_HijazAmplitudoMin, sDeviasi_HijazAmplitudoMin);
    ShobaDisNormal_AmpMin = hitungDistribusiNormal(AmpMin,
Mean_ShobaAmplitudoMin, sDeviasi_ShobaAmplitudoMin);
    SikahDisNormal_AmpMin = hitungDistribusiNormal(AmpMin,
Mean_SikahAmplitudoMin, sDeviasi_SikahAmplitudoMin);
    JiharkaDisNormal_AmpMin = hitungDistribusiNormal(AmpMin,
Mean_JikarkaAmplitudoMin, sDeviasi_JiharkaAmplitudoMin);
    RostDisNormal_AmpMin = hitungDistribusiNormal(AmpMin,
Mean_RostAmplitudoMin, sDeviasi_RostAmplitudoMin);
    NahawanDisNormal_AmpMin = hitungDistribusiNormal(AmpMin,
Mean_NahawanAmplitudoMin, sDeviasi_NahawanAmplitudoMin);

System.out.println("=====");

    System.out.println("Bayyati Dis Normal Amp Min: " +
BayyatiDisNormal_AmpMin);
    System.out.println("Hijaz Dis Normal Amp Min: " +
HijazDisNormal_AmpMin);
    System.out.println("Shoba Dis Normal Amp Min: " +
ShobaDisNormal_AmpMin);
    System.out.println("Sikah Dis Normal Amp Min: " +
SikahDisNormal_AmpMin);
    System.out.println("Jiharka Dis Normal Amp Min: " +
JiharkaDisNormal_AmpMin);
    System.out.println("Rost Dis Normal Amp Min: " +
RostDisNormal_AmpMin);
    System.out.println("Nahawan Dis Normal Amp Min: " +
NahawanDisNormal_AmpMin);

    BayyatiDisNormal_AmpMax = hitungDistribusiNormal(AmpMax,
Mean_BayyatiAmplitudoMax, sDeviasi_BayyatiAmplitudoMax);
    HijazDisNormal_AmpMax = hitungDistribusiNormal(AmpMax,
Mean_HijazAmplitudoMax, sDeviasi_HijazAmplitudoMax);
    ShobaDisNormal_AmpMax = hitungDistribusiNormal(AmpMax,
Mean_ShobaAmplitudoMax, sDeviasi_ShobaAmplitudoMax);
    SikahDisNormal_AmpMax = hitungDistribusiNormal(AmpMax,
```

```

Mean_SikahAmplitudoMax, sDeviasi_SikahAmplitudoMax);
    JiharkaDisNormal_AmpMax = hitungDistribusiNormal(AmpMax,
Mean_JikarkaAmplitudoMax, sDeviasi_JiharkaAmplitudoMax);
    RostDisNormal_AmpMax = hitungDistribusiNormal(AmpMax,
Mean_RostAmplitudoMax, sDeviasi_RostAmplitudoMax);
    NahawanDisNormal_AmpMax = hitungDistribusiNormal(AmpMax,
Mean_NahawanAmplitudoMax, sDeviasi_NahawanAmplitudoMax);

System.out.println("=====");

    System.out.println("Bayyati Dis Normal Amp Max: " +
BayyatiDisNormal_AmpMax);
    System.out.println("Hijaz Dis Normal Amp Max: " +
HijazDisNormal_AmpMax);
    System.out.println("Shoba Dis Normal Amp Max: " +
ShobaDisNormal_AmpMax);
    System.out.println("Sikah Dis Normal Amp Max: " +
SikahDisNormal_AmpMax);
    System.out.println("Jiharka Dis Normal Amp Max: " +
JiharkaDisNormal_AmpMax);
    System.out.println("Rost Dis Normal Amp Max: " +
RostDisNormal_AmpMax);
    System.out.println("Nahawan Dis Normal Amp Max: " +
NahawanDisNormal_AmpMax);

    BayyatiDisNormal_T = hitungDistribusiNormal(SudutTurun,
Mean_Bayyati_T, sDeviasi_Bayyati_T);
    HijazDisNormal_T = hitungDistribusiNormal(SudutTurun,
Mean_Hijaz_T, sDeviasi_Hijaz_T);
    ShobaDisNormal_T = hitungDistribusiNormal(SudutTurun,
Mean_Shoba_T, sDeviasi_Shoba_T);
    SikahDisNormal_T = hitungDistribusiNormal(SudutTurun,
Mean_Sikah_T, sDeviasi_Sikah_T);
    JiharkaDisNormal_T = hitungDistribusiNormal(SudutTurun,
Mean_Jikarka_T, sDeviasi_Jiharka_T);
    RostDisNormal_T = hitungDistribusiNormal(SudutTurun, Mean_Rost_T,
sDeviasi_Rost_T);
    NahawanDisNormal_T = hitungDistribusiNormal(SudutTurun,
Mean_Nahawan_T, sDeviasi_Nahawan_T);

System.out.println("=====");

    System.out.println("Bayyati Dis Normal Fre Min: " + BayyatiDisNormal_T);
    System.out.println("Bayyati Dis Normal Turun: " + BayyatiDisNormal_T);
    System.out.println("Hijaz Dis Normal Turun: " + HijazDisNormal_T);
    System.out.println("Shoba Dis Normal Turun: " + ShobaDisNormal_T);
    System.out.println("Sikah Dis Normal Turun: " + SikahDisNormal_T);
    System.out.println("Jiharka Dis Turun: " + JiharkaDisNormal_T);
    System.out.println("Rost Dis Normal Turun: " + RostDisNormal_T);
    System.out.println("Nahawan Dis Normal Turun: " + NahawanDisNormal_T);

    BayyatiDisNormal_N = hitungDistribusiNormal(SudutNaik,
Mean_Bayyati_N, sDeviasi_Bayyati_N);
    HijazDisNormal_N = hitungDistribusiNormal(SudutNaik,
Mean_Hijaz_N, sDeviasi_Hijaz_N);
    ShobaDisNormal_N = hitungDistribusiNormal(SudutNaik,
Mean_Shoba_N, sDeviasi_Shoba_N);
    SikahDisNormal_N = hitungDistribusiNormal(SudutNaik,
Mean_Sikah_N, sDeviasi_Sikah_N);
    JiharkaDisNormal_N = hitungDistribusiNormal(SudutNaik,
Mean_Jikarka_N, sDeviasi_Jiharka_N);
    RostDisNormal_N = hitungDistribusiNormal(SudutNaik, Mean_Rost_N,
sDeviasi_Rost_N);
    NahawanDisNormal_N = hitungDistribusiNormal(SudutNaik,
Mean_Nahawan_N, sDeviasi_Nahawan_N);

```

```

System.out.println("=====");

System.out.println("Bayyati Dis Normal Naik: " + BayyatiDisNormal_N);
System.out.println("Hijaz Dis Normal Naik: " + HijazDisNormal_N);
System.out.println("Shoba Dis Normal Naik: " + ShobaDisNormal_N);
System.out.println("Sikah Dis Normal Naik: " + SikahDisNormal_N);
System.out.println("Jiharka Dis Normal Naik: " + JiharkaDisNormal_N);
System.out.println("Rost Dis Normal Naik: " + RostDisNormal_N);
System.out.println("Nahawan Dis Normal Naik: " + NahawanDisNormal_N);

System.out.println("=====");

    hasilBayyati = P_Bayyati * BayyatiDisNormal_AmpMin *
BayyatiDisNormal_AmpMax * BayyatiDisNormal_N * BayyatiDisNormal_T;
    hasilHijaz = P_Hijaz * HijazDisNormal_AmpMin *
HijazDisNormal_AmpMax * HijazDisNormal_N * HijazDisNormal_T;
    hasilShoba = P_Shoba * ShobaDisNormal_AmpMin *
ShobaDisNormal_AmpMax * ShobaDisNormal_N * ShobaDisNormal_T;
    hasilSikah = P_Sikah * SikahDisNormal_AmpMin *
SikahDisNormal_AmpMax * SikahDisNormal_N * SikahDisNormal_T;
    hasilJiharka = P_Jiharka * JiharkaDisNormal_AmpMin *
JiharkaDisNormal_AmpMax * JiharkaDisNormal_N * JiharkaDisNormal_T;
    hasilRost = P_Rost * RostDisNormal_AmpMin * RostDisNormal_AmpMax
* RostDisNormal_N * RostDisNormal_T;
    hasilNahawan = P_Nahawan * NahawanDisNormal_AmpMin *
NahawanDisNormal_AmpMax * NahawanDisNormal N * NahawanDisNormal T;

```

Berikut adalah Gambar 4.9 hasil dari *running source code* di atas adalah:

Bayyati Dis Normal Amp Min	:	0.0029119034424750024
Hijaz Dis Normal Amp Min	:	0.003733831228619394
Shoba Dis Normal Amp Min	:	0.002308546081556545
Sikah Dis Normal Amp Min	:	0.003240205668268785
Jiharka Dis Normal Amp Min	:	0.003417794014315711
Rost Dis Normal Amp Min	:	0.003443249065131005
Nahawan Dis Normal Amp Min	:	0.002920184736473388
=====		
Bayyati Dis Normal Amp Max	:	9.756576420254856E-5
Hijaz Dis Normal Amp Max	:	8.987886276988416E-5
Shoba Dis Normal Amp Max	:	9.537468636364813E-5
Sikah Dis Normal Amp Max	:	8.879132555817315E-5
Jiharka Dis Normal Amp Max	:	9.731920790589895E-5
Rost Dis Normal Amp Max	:	7.782053960892893E-5
Nahawan Dis Normal Amp Max	:	7.423056613190145E-5
=====		
Bayyati Dis Normal Turun	:	0.0011781598273400032
Hijaz Dis Normal Turun	:	7.207622953100951E-4
Shoba Dis Normal Turun	:	0.0010239408439808152
Sikah Dis Normal Turun	:	0.0012501070727998832
Jiharka Dis Normal Turun	:	0.0010193980065815443
Rost Dis Normal Turun	:	5.231206132360681E-4
Nahawan Dis Normal Turun	:	0.0016885366527376181
=====		
Bayyati Dis Normal Naik	:	0.0010624058763669636
Hijaz Dis Normal Naik	:	7.61567719321965E-4
Shoba Dis Normal Naik	:	8.602121676741807E-4
Sikah Dis Normal Naik	:	9.605084607459681E-4
Jiharka Dis Normal Naik	:	8.994173352138229E-4
Rost Dis Normal Naik	:	5.03205393641857E-4
Nahawan Dis Normal Naik	:	0.0012515293155637615

Gambar 4.9 Output Perhitungan Distribusi Normal Tiap Jenis Irama

Pada hasil dari perhitungan *Naïve Bayes Classifier*, selanjutnya dilakukan proses pengecekan *similarity*. Dimana pada proses ini hanya dilakukan untuk mempertajam akurasi klasifikasi pada perhitungan *Naïve Bayes Classifier*.

```
// cek similarity
CekSimilarity CekSim = new CekSimilarity();
CekSim.getSim(path);

System.out.println("Sebelum Similarity");
System.out.println("bayyati " + hasilBayyati);
System.out.println("hijaz " + hasilHijaz);
System.out.println("shoba " + hasilShoba);
System.out.println("sukah " + hasilSukah);
System.out.println("jiharka " + hasilJiharka);
System.out.println("rost " + hasilRost);
System.out.println("nahawan " + hasilNahawan);

//tambahkan hasil naive bayes dengan hasil similarity
hasilBayyati = hasilBayyati + CekSim.finalSimBa;
hasilHijaz = hasilHijaz + CekSim.finalSimHi;
hasilShoba = hasilShoba + CekSim.finalSimSh;
hasilSukah = hasilSukah + CekSim.finalSimSi;
hasilJiharka = hasilJiharka + CekSim.finalSimJi;
hasilNahawan = hasilNahawan + CekSim.finalSimNa;
hasilRost = hasilRost + CekSim.finalSimRo;

double hasilKum[] = {hasilBayyati, hasilHijaz, hasilJiharka,
hasilNahawan, hasilRost, hasilShoba, hasilSukah};

System.out.println("Sesudah Similarity");
System.out.println("bayyati " + hasilBayyati);
System.out.println("hijaz " + hasilHijaz);
System.out.println("shoba " + hasilShoba);
System.out.println("sukah " + hasilSukah);
System.out.println("jiharka " + hasilJiharka);
System.out.println("rost " + hasilRost);
System.out.println("nahawan " + hasilNahawan);

maxValue = hasilKum[0];
int indeks = 0;
for (int i = 0; i < hasilKum.length; i++) {
    if (hasilKum[i] > maxValue) {
        maxValue = hasilKum[i];
        indeks = i;
    }
}
}
```

Sehingga hasil *running* dari proses sebelum dan sesudah pengecekan *similarity* seperti pada Gambar 4.10 adalah sebagai berikut:

Proses Similarity	
Hasil Sim Ba	1.0
Hasil Sim Hi	0.9130411552624323
Hasil Sim Sh	0.9685527803364316
Hasil Sim Si	0.9315948973202168
Hasil Sim Ji	0.9337844822597093
Hasil Sim Na	0.8823515583091657
Hasil Sim Ro	0.8928551324605987
Hasil Sim Ba	0.1
Hasil Sim Hi	0.09130411552624323
Hasil Sim Sh	0.09685527803364316
Hasil Sim Si	0.09315948973202168
Hasil Sim Ji	0.09337844822597094
Hasil Sim Na	0.08823515583091657
Hasil Sim Ro	0.08928551324605986
Sebelum Similarity	
bayyati	4.9414685473467616E-14
hijaz	2.2385441715348952E-14
shoba	3.0135582222882013E-14
sikah	4.9922656915878673E-14
jiharka	5.2990406798152626E-14
rost	1.0292962960652414E-14
nahawan	7.057536956975173E-14
Sesudah Similarity	
bayyati	0.100000000000004942
hijaz	0.09130411552626562
shoba	0.09685527803367329
sikah	0.0931594897320716
jiharka	0.09337844822602392
rost	0.08928551324607016
nahawan	0.08823515583098714

Gambar 4.10 Output Proses Sebelum dan Sesudah *Similarity*

4.4 Hasil dan Pengujian

Sebelum melakukan pengujian kinerja *system* menggunakan data *testing*, sebaiknya peneliti terlebih dahulu mencoba melakukan pengujian *system* dengan menggunakan 42 data rekaman diambil secara *random* dari 210 rekaman data *training* yang tersedia. Kemudian, untuk pengujian *system* pada data *training* yaitu dilakukan sebagai berikut:

Tabel 4.2 Hasil Pengujian Kinerja System Data Training

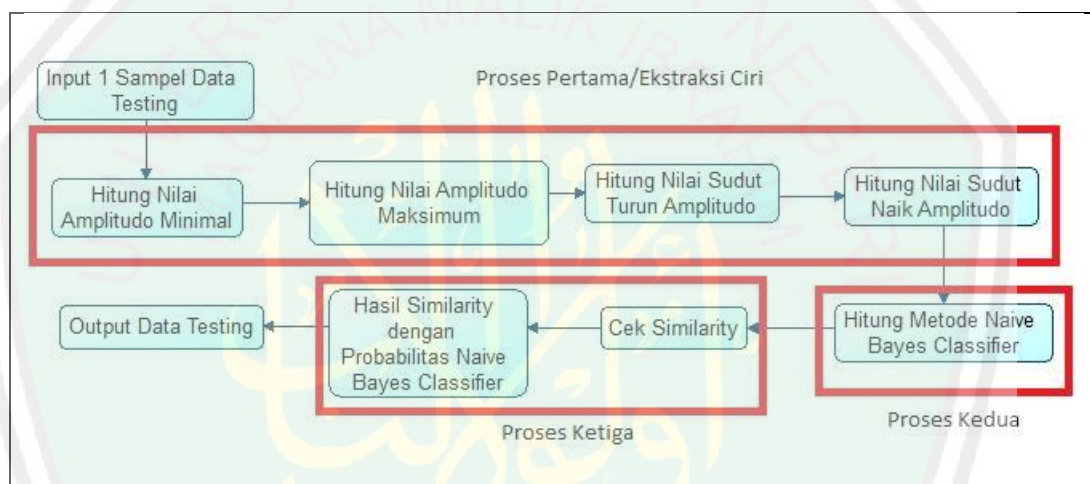
Data Ke-n	Hasil Pengujian			Keterangan
	Jenis Irama	Nama File.wav	Hasil System	
5	Bayyati	Ba005432.wav	Bayyati	Sesuai
10	Bayyati	Ba010431.wav	Bayyati	Sesuai
11	Hijaz	Hi001433.wav	Hijaz	Sesuai
18	Hijaz	Hi008432.wav	Jiharkah	Tidak Sesuai
23	Jiharka	Ji003431.wav	Jiharka	Sesuai
29	Jiharka	Ji009431.wav	Jiharka	Sesuai

Data Ke-n	Hasil Pengujian			Keterangan
	Jenis Irama	Nama File.wav	Hasil System	
32	Nahawan	Na002431.wav	Nahawan	Sesuai
35	Nahawan	Na005431.wav	Nahawan	Sesuai
44	Rost	Ro004432.wav	Rost	Sesuai
50	Rost	Ro010432.wav	Rost	Sesuai
51	Shoba	Sh001433.wav	Shoba	Sesuai
56	Shoba	Sh006431.wav	Shoba	Sesuai
63	Sikah	Si003432.wav	Sikah	Sesuai
69	Sikah	Si009431.wav	Jiharka	Tidak Sesuai
74	Bayyati	Ba004532.wav	Bayyati	Sesuai
78	Bayyati	Ba008532.wav	Bayyati	Sesuai
81	Hijaz	Hi001531.wav	Hijaz	Sesuai
86	Hijaz	Hi006532.wav	Bayyati	Tidak Sesuai
97	Jiharka	Ji007531.wav	Jiharka	Sesuai
99	Jiharka	Ji009531.wav	Jiharka	Sesuai
103	Nahawan	Na003531.wav	Nahawan	Sesuai
108	Nahawan	Na008531.wav	Nahawan	Sesuai
115	Rost	Ro005531.wav	Rost	Sesuai
118	Rost	Ro008531.wav	Rost	Sesuai
121	Shoba	Sh001531.wav	Shoba	Sesuai
125	Shoba	Sh005531.wav	Shoba	Sesuai
132	Sikah	Si002531.wav	Sikah	Sesuai
134	Sikah	Si004531.wav	Jiharka	Tidak Sesuai
148	Bayyati	Ba008821.wav	Bayyati	Sesuai
150	Bayyati	Ba010822.wav	Bayyati	Sesuai
153	Hijaz	Hi003822.wav	Hijaz	Sesuai
158	Hijaz	Hi008822.wav	Hijaz	Sesuai
161	Jiharka	Ji001821.wav	Jiharka	Sesuai
163	Jiharka	Ji003821.wav	Jiharka	Sesuai
172	Nahawan	Na002821.wav	Nahawan	Sesuai
180	Nahawan	Na010821.wav	Nahawan	Sesuai
184	Rost	Ro004821.wav	Rost	Sesuai
189	Rost	Ro009821.wav	Jiharkah	Tidak Sesuai
194	Shoba	Sh004821.wav	Shoba	Sesuai
196	Shoba	Sh006822.wav	Jiharkah	Tidak Sesuai
207	Sikah	Si007821.wav	Sikah	Sesuai
210	Sikah	Si010822.wav	Sikah	Sesuai

Berdasarkan Tabel 4.2 maka, hasil pengujian terhadap 42 data rekaman diambil secara *random* dari 210 rekaman data *training*, diperoleh hasil bahwa terdapat 36 data rekaman diklasifikasikan dengan benar dan terdapat 6 data rekaman misklasifikasi, hal ini disebabkan dikarenakan antara satu rekaman jenis

irama *qiro'ah* dengan rekaman jenis irama *qiro'ah* lainnya mempunyai amplitudo yang tidak menentu. $Akurasi(\%) = \frac{36}{42} \times 100 = 85,7\%$ maka, persentase pengujian data *training* ini diperoleh hasil sebesar akurasi 85,7%

Selanjutnya, untuk menilai kinerja *system* pengenalan jenis-jenis irama *qiro'ah* dilakukan pengujian, terdapat 63 data *testing* yang sudah disediakan. Pengujian ini dilakukan dengan menggunakan parameter-parameter sesuai dengan prosedur penelitian yang telah ditentukan. Di bawah ini merupakan Gambar 4.11 diagram blok pengujian terhadap data *testing*.



Gambar 4.11 Diagram Blok Pengujian Data Testing

Keterangan:

1. Masukkan sampel data *testing*, pada tahap ini merupakan langkah awal dalam proses pengujian menggunakan data atau *file* suara rekaman *qiro'ah* dengan ke-tujuh jenis iramanya dari 63 data yang telah tersedia diantaranya:
 - 9 data jenis irama Bayyati
 - 9 data jenis irama Hijaz
 - 9 data jenis irama Nahawan

- 9 data jenis irama Rost
 - 9 data jenis irama Jiharkah
 - 9 data jenis irama Shoba
 - 9 data jenis irama Sikah
2. Proses pertama ekstraksi ciri, yaitu dibutuhkan untuk mengetahui karakteristik berupa pengambilan nilai minimal amplitudo, maksimal amplitudo, sudut turun amplitudo, dan sudut naik amplitudo suara jenis irama *qiro'ah*.
 3. Proses kedua menghitung metode *Naïve Bayes Classifier*, pada tahap ini didapatkan dengan memanfaatkan nilai rata-rata minimal, maksimal, sudut turun, dan sudut naik amplitudo. Selanjutnya, standar deviasi pada data *training* yang telah diperoleh dan disimpan sebelumnya sesuai nilai-nilai ke-tujuh *class* jenis irama *qiro'ah*. Kemudian, langkah selanjutnya diperlukan untuk mengukur seberapa sering parameter minimal, maksimal, sudut turun, dan sudut naik amplitudo data *testing* muncul dengan mengalikan nilai maksimal, minimal, sudut turun, dan sudut naik amplitudo dengan nilai probabilitas pada data *training*. Pada akhir tahap ini, logika perulangan dilakukan untuk membandingkan nilai dan peluang data *testing* berdasarkan nilai probabilitas peluang data *training*.
 4. Proses ketiga cek *similarity*, pada tahap ini nilai-nilai pada yang sudah terhitung dengan menggunakan metode *Naïve Bayes Classifier* selanjutnya akan dikombinasikan supaya pada saat melakukan ujicoba data *testing* akurasi menjadi bertambah.

Berdasarkan pada Gambar 4.11 blok diagram pengujian data *testing* di atas, berikut Gambar 4.12 adalah tampilan dari hasil pengujian data *testing*.



Gambar 4.12 Hasil Pengujian Data Testing contoh File Ba011431.wav

Pada Tabel 4.3 di bawah ini menunjukkan hasil pengujian kinerja *system* menggunakan 63 suara rekaman data *testing*.

Tabel 4.3 Hasil Pengujian Kinerja System Data Testing

Data Ke-n	Hasil Pengujian			Keterangan
	Jenis Irama	Nama File.wav	Hasil System	
1	Bayyati	Ba011431.wav	Bayyati	BENAR
2	Bayyati	Ba011532.wav	Jiharkah	SALAH
3	Bayyati	Ba011822.wav	Hijaz	SALAH
4	Bayyati	Ba012431.wav	Jiharkah	SALAH
5	Bayyati	Ba012532.wav	Bayyati	BENAR
6	Bayyati	Ba012822.wav	Bayyati	BENAR
7	Bayyati	Ba013432.wav	Bayyati	BENAR
8	Bayyati	Ba013531.wav	Bayyati	BENAR
9	Bayyati	Ba013821.wav	Bayyati	BENAR
10	Hijaz	Hi011432.wav	Hijaz	BENAR
11	Hijaz	Hi011531.wav	Jiharkah	SALAH
12	Hijaz	Hi011822.wav	Hijaz	BENAR
13	Hijaz	Hi012432.wav	Jiharkah	SALAH
14	Hijaz	Hi012532.wav	Jiharkah	SALAH
15	Hijaz	Hi012822.wav	Hijaz	BENAR

Data Ke-n	Hasil Pengujian			Keterangan
	Jenis Irama	Nama File.wav	Hasil System	
16	Hijaz	Hi013431.wav	Jiharkah	SALAH
17	Hijaz	Hi013531.wav	Jiharkah	SALAH
18	Hijaz	Hi013821.wav	Bayyati	SALAH
19	Jiharkah	Ji011431.wav	Jiharkah	BENAR
20	Jiharkah	Ji011531.wav	Jihakah	BENAR
21	Jiharkah	Ji011821.wav	Hijaz	SALAH
22	Jiharkah	Ji012431.wav	Jiharkah	BENAR
23	Jiharkah	Ji012532.wav	Jiharkah	BENAR
24	Jiharkah	Ji012821.wav	Shoba	SALAH
25	Jiharkah	Ji013431.wav	Jiharkah	BENAR
26	Jiharkah	Ji013531.wav	Jiharkah	BENAR
27	Jiharkah	Ji013822.wav	Bayyati	SALAH
28	Nahawan	Na011432.wav	Nahawan	BENAR
29	Nahawan	Na011531.wav	Nahawan	BENAR
30	Nahawan	Na011822.wav	Bayyati	SALAH
31	Nahawan	Na012431.wav	Nahawan	BENAR
32	Nahawan	Na012532.wav	Jiharkah	SALAH
33	Nahawan	Na012822.wav	Nahawan	BENAR
34	Nahawan	Na013431.wav	Nahawan	BENAR
35	Nahawan	Na013531.wav	Nahawan	BENAR
36	Nahawan	Na013821.wav	Nahawan	BENAR
37	Rost	Ro011431.wav	Rost	BENAR
38	Rost	Ro011531.wav	Jiharkah	SALAH
39	Rost	Ro011821.wav	Hijaz	SALAH
40	Rost	Ro012432.wav	Jiharkah	SALAH
41	Rost	Ro012532.wav	Rost	BENAR
42	Rost	Ro012821.wav	Rost	BENAR
43	Rost	Ro013432.wav	Rost	BENAR
44	Rost	Ro013531.wav	Jiharkah	SALAH
45	Rost	Ro013821.wav	Rost	BENAR
46	Shoba	Sh011432.wav	Shoba	BENAR
47	Shoba	Sh011531.wav	Jiharkah	SALAH
48	Shoba	Sh011821.wav	Hijaz	SALAH
49	Shoba	Sh012431.wav	Shoba	BENAR
50	Shoba	Sh012532.wav	Shoba	BENAR
51	Shoba	Sh012822.wav	Shoba	BENAR
52	Shoba	Sh013431.wav	Shoba	BENAR
53	Shoba	Sh013531.wav	Jiharkah	SALAH
54	Shoba	Sh013822.wav	Bayyati	SALAH
55	Sikah	Si011432.wav	Sikah	BENAR
56	Sikah	Si011532.wav	Jiharkah	SALAH
57	Sikah	Si011822.wav	Hijaz	SALAH
58	Sikah	Si012432.wav	Sikah	BENAR
59	Sikah	Si012531.wav	Jiharkah	SALAH

Data Ke-n	Hasil Pengujian			Keterangan
	Jenis Irama	Nama File.wav	Hasil System	
60	Sikah	Si012822.wav	Sikah	BENAR
61	Sikah	Si013432.wav	Jiharkah	SALAH
62	Sikah	Si013531.wav	Jiharkah	SALAH
63	Sikah	Si013821.wav	Sikah	BENAR

Berdasarkan Tabel 4.3 hasil pengujian terhadap semua data, diperoleh hasil bahwa terdapat 36 hasil yang sesuai dengan hasil yang diberikan dan 27 hasil yang tidak sesuai dengan hasil yang diberikan, hal ini disebabkan dikarenakan antara satu rekaman jenis irama *qiro'ah* dengan rekaman jenis irama *qiro'ah* lainnya mempunyai amplitudo yang tidak menentu.

Setelah melakukan pengujian kinerja *system*, maka selanjutnya dilakukan proses perhitungan akurasi. Pada proses perhitungan akurasi, penulis menggunakan metode *confusion matrix*. Metode ini terdapat *True-Positive* (TP) adalah jumlah rekaman *positive* yang diklasifikasikan dengan benar, *False-Positive* (FP) adalah jumlah rekaman *negative* yang diklasifikasikan sebagai *positive*, kemudian *False-Negative* (FN) adalah jumlah rekaman *positive* yang diklasifikasikan sebagai *negative*, dan *True-Negative* adalah jumlah rekaman data benar dan tidak diklasifikasikan. Setelah data *testing* diklasifikasikan, selanjutnya menghitung jumlah akurasi, presisi dan *recall*. Akurasi pada perhitungan ini menggambarkan seberapa akurat *system* dapat mengklasifikasikan data secara benar. Presisi merupakan tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh *system*. *Recall* adalah tingkat keberhasilan *system* dalam menemukan kembali sebuah informasi.

Tabel 4.4 Confusion Matrix

Confusion Matrix		Kebenaran	
		+	-
Prediksi	+	<i>True-Positive (TP)</i>	<i>False-Positive (FP)</i>
	-	<i>False-Negative (FN)</i>	<i>True-Negative (TN)</i>

Berdasarkan 63 rekaman data *testing* dengan menggunakan perhitungan akurasi *confusion matrix*. Akurasi adalah persentase dari total data yang diidentifikasi dan dinilai benar, berikut Tabel 4.5 adalah hasil klasifikasi akurasi metode *confusion matrix*.

Tabel 4.5 Hasil Klasifikasi Akurasi Perhitungan Metode Confusion Matrix

Hasil Prediksi	Kebenaran						
	Bayati	Hijaz	Jiharkah	Nahawan	Rost	Shoba	Sikah
Bayati	6	1	1	1	0	1	0
Hijaz	1	3	1	0	1	1	1
Jiharkah	2	5	6	1	3	2	4
Nahawan	0	0	0	7	0	0	0
Rost	0	0	0	0	5	0	0
Shoba	0	0	1	0	0	5	0
Sikah	0	0	0	0	0	0	4

Pada Tabel 4.4 diatas terdapat total 63 rekaman sebagai data *testing*, sehingga terdapat total 36 data rekaman diklasifikasikan dengan benar dan 27 hasil yang tidak sesuai atau misklasifikasi. Hal ini disebabkan dikarenakan antara satu rekaman jenis irama *qiro'ah* dengan rekaman jenis irama *qiro'ah* lainnya mempunyai amplitudo yang tidak menentu. Jika dirata-ratakan nilai fitur dari amplitudonya, ada yang memiliki nilai hampir sama meskipun memiliki jenis irama *qiro'ah* yang berbeda, dengan hasil yang diberikan diantaranya sebagai berikut:

- 9 jenis irama Bayyati, dari pengujian *system* terdapat 6 data rekaman diklasifikasikan dengan benar dan terdapat 3 data rekaman misklasifikasi yaitu: 1 data rekaman diklasifikasikan kedalam jenis

irama Hijaz dan 2 data rekaman diklasifikasikan kedalam jenis irama Jiharkah.

- 9 jenis irama Hijaz, dari pengujian *system* terdapat 3 data rekaman diklasifikasikan dengan benar dan terdapat 6 data rekaman misklasifikasi yaitu: 1 data rekaman diklasifikasikan kedalam jenis irama Bayyati dan 5 data rekaman diklasifikasikan kedalam jenis irama Jiharkah.
- 9 jenis irama Jiharkah, dari pengujian *system* terdapat 6 data rekaman diklasifikasikan dengan benar dan terdapat 3 data rekaman misklasifikasi yaitu: 1 data rekaman diklasifikasikan kedalam jenis irama Bayyati, 1 data rekaman diklasifikasikan kedalam jenis irama Hijaz, dan 1 data rekaman diklasifikasikan kedalam jenis irama Shoba.
- 9 jenis irama Nahawan, dari pengujian *system* terdapat 7 data rekaman diklasifikasikan dengan benar dan terdapat 2 data rekaman misklasifikasi yaitu: 1 data rekaman diklasifikasikan kedalam jenis irama Bayyati dan 1 data rekaman diklasifikasikan kedalam jenis irama Jiharkah.
- 9 jenis irama Rost, dari pengujian *system* terdapat 5 data rekaman diklasifikasikan dengan benar dan terdapat 4 data rekaman misklasifikasi yaitu: 1 data rekaman diklasifikasikan kedalam jenis irama Hijaz dan 3 data rekaman diklasifikasikan kedalam jenis irama Jiharkah.
- 9 jenis irama Shoba, dari pengujian *system* terdapat 5 data rekaman diklasifikasikan dengan benar dan terdapat 4 data rekaman

misklasifikasi yaitu: 1 data rekaman diklasifikasikan kedalam jenis irama Bayyati, 1 data rekaman diklasifikasikan kedalam jenis irama Hijaz, dan 2 data rekaman diklasifikasikan kedalam jenis irama Jiharkah.

- 9 jenis irama Sikah, dari pengujian *system* terdapat 4 data rekaman diklasifikasikan dengan benar dan terdapat 5 data rekaman misklasifikasi yaitu: 1 data rekaman diklasifikasikan kedalam jenis irama Hijaz dan 4 data rekaman diklasifikasikan kedalam jenis irama Jiharkah.

Selanjutnya, untuk menghitung akurasi, presisi, dan *recall* pada proses perhitungan metode *confusion matrix* adalah sebagai berikut:

$$- \text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$- \text{Presisi} = \frac{TP}{FP + TP}$$

$$- \text{Recall} = \frac{TP}{FP + TP}$$

Berdasarkan dari perhitungan dengan menggunakan *confusion matrix* dapat dihasilkan perhitungan akurasi, presisi, dan *recall* seperti Tabel 4.6 dibawah ini:

Tabel 4.6 Hasil Nilai Akurasi, Presisi, dan Recall Setiap Jenis Irama qiro'ah

No.	Jenis Irama qiro'ah	Akurasi	Presisi	Recall
1	Bayati	0,66	0,66	0,6
2	Hijaz	0,33	0,33	0,35
3	Jiharkah	0,66	0,66	0,26
4	Nahawan	0,77	0,77	1
5	Rost	0,56	0,56	1
6	Shoba	0,56	0,56	0,83
7	Sikah	0,44	0,44	1

Berdasarkan Tabel 4.6 hasil nilai akurasi di atas menunjukkan setiap jenis irama *qiro'ah* memiliki akurasi yang berbeda-beda. Selanjutnya, nilai rata-rata dari semua jenis irama *qiro'ah* ditampilkan pada Tabel 4.7.

Tabel 4.7 Nilai Akhir Akurasi, Presisi, dan Recall

No.	Variabel	Nilai Rata-Rata	Nilai Persentase
1	Akurasi	0,57	57 %
2	Presisi	0,57	57 %
3	Recall	0,72	72 %

Dari Tabel 4.7 di atas diambil nilai total kinerja *system* untuk semua jenis irama *qiro'ah*, dengan perhitungan metode akurasi *confusion matrix* akan diperoleh nilai total kinerja sebesar akurasi 57%, presisi 57%, dan *recall* 72%. Hasil tersebut menunjukkan bahwa nilai cukup sehingga dapat disimpulkan bahwa model *system* telah memiliki kinerja yang cukup.

4.5 Integrasi Penelitian terhadap Al-Quran dan Hadits

Seorang muslim dan muslimah yang taat dan baik adalah mereka yang mampu menuntut ilmu dengan baik dan mempraktekannya dalam kehidupan sehari-hari itu jauh lebih baik. Selain itu, dapat melihat keadaan sekitar dan memecahkan segala problematika di dalamnya berdasarkan teori-teori keilmuan yang dimilikinya sangat baik agar apa yang selama ini telah dipelajari dapat diterapkan dengan baik di dalam kehidupan. Sama halnya dengan penelitian pengenalan jenis-jenis irama *qiro'ah* ini. Latar belakang dalam penelitian ini penulis mengambil dari problematika nyata yang terjadi di sekitar lingkungan Universitas Islam Negeri Maulana Malik Ibrahim Malang. Sebuah hadits Rasulullah yang berbunyi, “Ya Allah, aku meminta kepada-Mu ilmu yang bermanfaat dan aku berlindung kepada-Mu dari ilmu yang tidak bermanfaat”.

(Muhayan, M., & Barito, S. R, 2007). Hadits tersebut mewajibkan seseorang untuk selalu meminta ilmu yang bermanfaat dari Allah, karena seluruh ilmu yang ada di dunia ini baik yang tidak dan atau bermanfaat, yang diketahui atau pun tidak harus selalu kembali dan bersanding diri kepada Allah sebagaimana Anas bin Malik berkata bahwa Rasulullah bersabda, “*Ya Allah, sesungguhnya aku berlindung kepada-Mu dari ilmu yang tidak bermanfaat, dari amal yang tidak diangkat, dari hati yang tidak khusyu’ dan dari perkataan yang tidak didengarkan*” (Muhayan, M., & Barito, S. R, 2007). Dalam Surat Shad ayat 29, Allah SWT berfirman:

كِتَابٌ أَنْزَلْنَاهُ إِلَيْكَ مُبَارَكٌ لِيَدَّبَّرُوا آيَاتِهِ وَلِيَتَذَكَّرَ أُولُو الْأَلْبَابِ

Artinya: Ini adalah sebuah kitab yang Kami turunkan kepadamu penuh dengan berkah supaya mereka memperhatikan ayat-ayatnya dan supaya mendapat pelajaran orang-orang yang mempunyai fikiran.

Kemudian, di dalam Al-Quran menjelaskan bahwa Dialah Allah yang menurunkan Al-Quran untuk hambanya agar memahami makna dari Kitab dan mengikuti Sunnahnya, dan hanya orang yang diberikan anugerah dialah yang dapat mengambil pelajaran dari firman-Nya. Kemudian, dilanjutkan Kembali pada surah Al-Baqarah ayat 269.

يُؤْتِي الْحِكْمَةَ مَنْ يَشَاءُ ۚ وَمَنْ يُؤْتَ الْحِكْمَةَ فَقَدْ أُوتِيَ خَيْرًا كَثِيرًا ۚ وَمَا يَذَّكَّرُ إِلَّا أُولُو الْأَلْبَابِ -

البقرة: ٢٦٩

Artinya: Allah menganugerahkan al hikmah (kefahaman yang dalam tentang Al Quran dan As Sunnah) kepada siapa yang dikehendaki-Nya. Dan barangsiapa yang dianugerahi hikmah, ia benar-benar telah dianugerahi karunia yang

banyak. Dan hanya orang-orang yang berakallah yang dapat mengambil pelajaran (dari firman Allah).

Dari kedua ayat di atas dapat diambil pelajarannya, bahwa memahami bacaan Al-Quran sangatlah berpengaruh bagi kehidupan lingkungan sekitar. Manfaat dalam hal ini bukan hanya membaca saja, namun hikmah apa yang kamu dapatkan setelah membacanya atau ilmu yang dapat dikembangkan dari keduanya. Seperti pada teori ilmu pengetahuan alam akan bermanfaat bagi para pembelajarnya, sebagai kendaraan akan membantu pekerjaan manusia. Maka, manfaat diberikan anugrah atau hikmah dari Allah sangat berarti bagi manusia.

Pada penelitian ini, penulis mengambil manfaat dari aplikasi penentuan jenis-jenis irama *qiro'ah* agar dapat digunakan sebagai bahan pembelajaran untuk mengetahui jenis kategori irama apa yang sedang dibacanya oleh pengguna. Dalam penelitian ini, rekan-rekan yang bersedia dalam proses prekaman jenis-jenis irama *qiro'ah* dilakukan secara formal dan tanpa ada paksaan sedikitpun kepada rekan-rekan. Tujuan dari aplikasi ini adalah untuk pembelajaran dan mengetahui seberapa benar ayat atau surah yang di baca melalui aplikasi penentuan jenis-jenis irama *qiro'ah* menggunakan metode *Naïve Bayes Classifier*. Penulis sangat berharap, semoga proses yang telah dilakukan ini dapat dilanjutkan dan dikembangkan kembali, karena dalam pembahasan ini sangatlah menarik. Agar penelitian-penelitian tentang penentuan jenis-jenis irama *qiro'ah* semakin bermunculan dengan berbagai macam metode yang diterapkan.

Sedikit kata mutiara menerangkan bahwa “Jika kamu merasakan lelahnya mencari ilmu, maka bersiaplah juga bahwa akan kau rasakan perihnya kebodohan”. Sehingga, dari hasil ikhtiyar penelitian ini telah didapatkan bahwa

untuk memperoleh hasil yang lain sebagai bahan perbandingan, masih ada lagi teori keilmuan yang dapat diterapkan dan tidak berhenti pada kesimpulan yang sudah dimiliki oleh penelitian penentuan jenis-jenis irama *qiro'ah* ini.



BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dari penelitian yang telah dilakukan, maka peneliti dapat menarik kesimpulan bahwa metode *Naive Bayes Classifier* dengan tambahan pengecekan menggunakan *similarity* dalam sistem penentuan klasifikasi yang diimplemetasikan untuk penentuan jenis-jenis irama *qiro'ah* dengan parameter nilai minimal amplitudo, maksimal amplitudo, sudut turun amplitudo, dan sudut naik amplitudo memperoleh hasil sebagai berikut:

1. Data yang digunakan dalam penelitian ini adalah data rekaman yang mahir membawakan ke-tujuh jenis-jenis irama *qiro'ah* dengan format *audio*.wav*. Data tersebut dikelompokkan kembali menjadi data *training* dan data *testing*. Untuk 210 rekaman sebagai data *training* dan 63 rekaman sebagai data *testing*. Data tersebut digunakan untuk proses klasifikasi 9 data *testing* terhadap salah satu *class* terhadap 7 (tujuh) *class* yang di ambil dari 63 rekaman keseluruhan data *testing*.
2. Proses klasifikasi yang dilakukan peneliti, yaitu menggunakan *algoritma Naive Bayes Classifier* yang dapat digunakan untuk mencocokkan nilai probabilitas data *testing* terhadap nilai probabilitas data *training* dan ditambah dengan pengecekan *similarity* yang telah dihitung dan disimpan sebelumnya. Sehingga, proses penentuan jenis-jenis irama *qiro'ah* diambil berdasarkan nilai probabilitas tertinggi terdekat dari 7 (tujuh) *class* yang ada.

3. Berdasarkan eksperimen hasil penelitian yang telah dilakukan menggunakan algoritma *Naïve Bayes Classifier*, dengan nilai akurasi persentase data *training* 85,7%. Untuk data *testing* telah diperoleh total kinerja *system* sebesar akurasi 57%, presisi 57%, dan *recall* 72%.
4. Berdasarkan total kinerja dan laju *error* yang telah diketahui, maka dapat disimpulkan bahwa algoritma *Naïve Bayes Classifier* dapat digunakan untuk mengklasifikasikan jenis-jenis irama *qiro'ah*. Namun, belum secara maksimal untuk diterapkan dalam aplikasi ini, dikarenakan tingkat akurasi yang diperoleh belum mencapai 80%.
5. Kurang maksimal hal ini disebabkan dikarenakan antara satu rekaman jenis irama *qiro'ah* dengan rekaman jenis irama *qiro'ah* lainnya mempunyai amplitudo yang tidak menentu. Jika dirata-ratakan nilai fitur dari amplitudonya, ada yang memiliki nilai hampir sama meskipun memiliki jenis irama *qiro'ah* yang berbeda.

5.2 Saran

1. Dengan hasil tersebut, peneliti menyarankan bahwa sebelum proses klasifikasi menggunakan metode *Naive Bayes Classifier* perlu dilakukan terlebih dahulu proses ekstraksi fitur yang lebih kompleks seperti menggunakan parameter yang baru, untuk memberikan tingkat akurasi yang lebih tinggi dan lebih baik.
2. Parameter yang digunakan untuk mengukur berdasarkan kriteria jenis-jenis irama *qiro'ah* yaitu: banyaknya kemiripan amplitudo dari seluruh data *training*, menyebabkan proses perhitungan menjadi sedikit rumit.

3. Pengembangan aplikasi yang lebih inovatif dan penerapan sistem aplikasi berbasis *web* atau *android*.



DAFTAR PUSTAKA

Al-Quran. (n.d.). An-Nahl. Kudus.

Amplitudo. (2016, Oktober 4). Retrieved November 22, 2016, from Wikipedia:

<https://id.wikipedia.org/wiki/Amplitudo>

Andi, I. D., & Andi, N. H. (2015, Juni-Desember). Analisis Music Mining Information Retrieval Untuk Klasifikasi Jenis Music Bergener Menggunakan Algoritma Naive Bayes. *I*.

Apriliana, E. N., & Pulung, A. N. (2016). Pengujian Metode Naive Bayes Classifier Untuk Klasifikasi Genre Musik Berdasarkan Frekuensi .

Bunyi. (2016, Maret 8). Retrieved 11 22, 2016, from Wikipedia:

<https://id.wikipedia.org/wiki/Bunyi>

Burung Kontes. (2016, Maret 16). Retrieved from Wikipedia:

https://id.wikipedia.org/wiki/Burung_kontes

Dejonckere, P. H., Bradley, P., Clemente, P., Cornut, G., Buchman, L. C., Friendrich, G., . . . Woisard, V. (2015, July 27). A Basic Protocol for Functional Assessment of Voice Pathology, Especially for Investigating the Efficacy of (Phonosurgical) Treatments and Evaluating New Assessment Techniques. *Archiv fur Klinische und Experimentelle Ohren-Nasen- und Kehlkopfheilkunde*, 78.

Ganteng, N. (2011, 6 16). *Kaskus*. Retrieved from

<http://archive.kaskus.co.id/thread/9137892/0/arti-quotbenderaquot-pada-kontes-burung-berkicau>

- Gorunescu, F. (2011). Data Mining: Concepts and Techniques. *Verlag berlin Heidelberg: Springer*.
- Hamzah, A. (2012). *KLASIFIKASI TEKS DENGAN NAÏVE BAYES CLASSIFIER (NBC)*.
- Han & Kamber. (2006). Data Mining : Concepts and Techniques 2nd. *San Fransisco: Elsevie*.
- Irwandi, Marwan, Mahmud, A. H., & Abdullah. (2005). Upaya Pemanfaatan Rekaman Suara Burung dan Analisis Spektogram untuk Menyusun Metode Klasifikasi Berdasarkan Suara.
- Iskandar, J., & Iskandar, B. S. (2015). Pemanfaatan aneka ragam burung dalam kontes burung kicau dan dampaknya terhadap konservasi burung di alam : Studi kasus di Kota Bandung, Jawa Barat.
- Kusrini & E. T. Luthfi. (2009). *Algoritma Data Mining*. Yogyakarta: ANDI.
- Muhayan, M., & Barito, S. R. (2007). *Shahih Ibnu Hibban-Jilid I*. Jakarta: Pustaka Azzam-Anggota IKAPI.
- Murdiyanto, J. (2013, Agustus). *Sistem Penjurian Burung*. Retrieved from <http://jokomurdiyanto.blogspot.co.id/2013>
- Muslim, D. A. (2013, Januari 3). Retrieved from <https://omkicau.com/2013/01/03/enam-alasan-orang-memelihara-burung/>
- Powers, D. M. (2011). Evaluation: From Precision, Recall, and F-Measure to. *Journal of Machine Learning Technologies*, 37-63.
- Rizky Tri Wahyuni, Dkk. (2017). Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi. *Jurnal Teknik Elektro Vol. 9 No. 1, 1*.

Rusfidra, A. (2006, Agustus 1). SEMINAR NASIONAL MIPA 2006 de. *Pengembangan Riset Bioakustik di Indonesia: Studi pada Ayam Kokok Balenggek, Ayam Pelung dan Ayam Bekisar.*

Suara. (2013). Retrieved November 22, 2016, from Multimedia:
<http://gugum08.blogspot.co.id>

Wikipedia. (2015, Oktober 27). *Burung Kenari*. Retrieved September 29, 2016,
 from Wikipedia: https://id.wikipedia.org/wiki/Burung_kenari

Wikipedia. (2016, Oktober 10). *Burung*. Retrieved November 11, 2016, from
 Wikipedia: <https://id.wikipedia.org/wiki/Burung>

Zosterops. (2012, September 21). Retrieved from Zosterops Pleci Mania:
<https://web.facebook.com/notes/burung-pleci-kacamata-mata-puteh-zosterops/kriteria-dasar-penilaian-lomba-burung-berkicau/387049721365839>

LAMPIRAN-LAMPIRAN

A. Data Ahli Membacakan 7 Jenis Irama Qiro'ah

No.	Nama	Jenis Kelamin	Universitas / Kampus
1	Ami	Perempuan	UIN Malang
2	Maria	Perempuan	UIN Malang
3	Ardi	Laki-Laki	UIN Malang
4	Yusuf	Laki-Laki	UIN Malang
5	Fauzan	Laki-Laki	UM
6	Firdaus	Laki-Laki	UNISMA
7	Fuqon	Laki-Laki	UNISMA
8	Fira	Perempuan	UM
9	Agus Salim	Laki-Laki	UM
10	Iliya	Perempuan	UIN Malang

B. Validasi Data *Training* untuk Kesesuaian Data Sampel

Data Ke	Jenis Irama	Nama File.*wav	Hasil Jenis Irama
1	Bayyati	Ba001433.wav	Sesuai
2	Bayyati	Ba002432.wav	Sesuai
3	Bayyati	Ba003431.wav	Sesuai
4	Bayyati	Ba004432.wav	Sesuai
5	Bayyati	Ba005432.wav	Sesuai
6	Bayyati	Ba006431.wav	Sesuai
7	Bayyati	Ba007432.wav	Sesuai
8	Bayyati	Ba008432.wav	Sesuai
9	Bayyati	Ba009431.wav	Sesuai
10	Bayyati	Ba010431.wav	Sesuai
11	Hijaz	Hi001433.wav	Sesuai
12	Hijaz	Hi002431.wav	Sesuai
13	Hijaz	Hi003432.wav	Sesuai
14	Hijaz	Hi004431.wav	Sesuai
15	Hijaz	Hi005431.wav	Sesuai
16	Hijaz	Hi006432.wav	Sesuai
17	Hijaz	Hi007431.wav	Sesuai
18	Hijaz	Hi008432.wav	Sesuai
19	Hijaz	Hi009431.wav	Sesuai
20	Hijaz	Hi010431.wav	Sesuai
21	Jiharka	Ji001432.wav	Sesuai
22	Jiharka	Ji002431.wav	Sesuai
23	Jiharka	Ji003431.wav	Sesuai
24	Jiharka	Ji004431.wav	Sesuai
25	Jiharka	Ji005432.wav	Sesuai

Data Ke	Jenis Irama	Nama File.*wav	Hasil Jenis Irama
26	Jiharka	Ji006431.wav	Sesuai
27	Jiharka	Ji007431.wav	Sesuai
28	Jiharka	Ji008431.wav	Sesuai
29	Jiharka	Ji009431.wav	Sesuai
30	Jiharka	Ji010431.wav	Sesuai
31	Nahawan	Na001431.wav	Sesuai
32	Nahawan	Na002431.wav	Sesuai
33	Nahawan	Na003432.wav	Sesuai
34	Nahawan	Na004432.wav	Sesuai
35	Nahawan	Na005431.wav	Sesuai
36	Nahawan	Na006431.wav	Sesuai
37	Nahawan	Na007431.wav	Sesuai
38	Nahawan	Na008431.wav	Sesuai
39	Nahawan	Na009431.wav	Sesuai
40	Nahawan	Na010431.wav	Sesuai
41	Rost	Ro001431.wav	Sesuai
42	Rost	Ro002432.wav	Sesuai
43	Rost	Ro003431.wav	Sesuai
44	Rost	Ro004432.wav	Sesuai
45	Rost	Ro005431.wav	Sesuai
46	Rost	Ro006432.wav	Sesuai
47	Rost	Ro007432.wav	Sesuai
48	Rost	Ro008431.wav	Sesuai
49	Rost	Ro009431.wav	Sesuai
50	Rost	Ro010432.wav	Sesuai
51	Shoba	Sh001433.wav	Sesuai
52	Shoba	Sh002431.wav	Sesuai
53	Shoba	Sh003432.wav	Sesuai
54	Shoba	Sh004431.wav	Sesuai
55	Shoba	Sh005432.wav	Sesuai
56	Shoba	Sh006431.wav	Sesuai
57	Shoba	Sh007432.wav	Sesuai
58	Shoba	Sh008432.wav	Sesuai
59	Shoba	Sh009432.wav	Sesuai
60	Shoba	Sh010432.wav	Sesuai
61	Sikah	Si001431.wav	Sesuai
62	Sikah	Si002432.wav	Sesuai
63	Sikah	Si003432.wav	Sesuai
64	Sikah	Si004432.wav	Sesuai
65	Sikah	Si005431.wav	Sesuai
66	Sikah	Si006432.wav	Sesuai
67	Sikah	Si007432.wav	Sesuai
68	Sikah	Si008432.wav	Sesuai
69	Sikah	Si009431.wav	Sesuai
70	Sikah	Si010432.wav	Sesuai
71	Bayyati	Ba001531.wav	Sesuai

Data Ke	Jenis Irama	Nama File.*wav	Hasil Jenis Irama
72	Bayyati	Ba002531.wav	Sesuai
73	Bayyati	Ba003532.wav	Sesuai
74	Bayyati	Ba004532.wav	Sesuai
75	Bayyati	Ba005531.wav	Sesuai
76	Bayyati	Ba006532.wav	Sesuai
77	Bayyati	Ba007532.wav	Sesuai
78	Bayyati	Ba008532.wav	Sesuai
79	Bayyati	Ba009532.wav	Sesuai
80	Bayyati	Ba010532.wav	Sesuai
81	Hijaz	Hi001531.wav	Sesuai
82	Hijaz	Hi002531.wav	Sesuai
83	Hijaz	Hi003531.wav	Sesuai
84	Hijaz	Hi004532.wav	Sesuai
85	Hijaz	Hi005531.wav	Sesuai
86	Hijaz	Hi006532.wav	Sesuai
87	Hijaz	Hi007531.wav	Sesuai
88	Hijaz	Hi008532.wav	Sesuai
89	Hijaz	Hi009531.wav	Sesuai
90	Hijaz	Hi010532.wav	Sesuai
91	Jiharka	Ji001532.wav	Sesuai
92	Jiharka	Ji002531.wav	Sesuai
93	Jiharka	Ji003531.wav	Sesuai
94	Jiharka	Ji004531.wav	Sesuai
95	Jiharka	Ji005531.wav	Sesuai
96	Jiharka	Ji006532.wav	Sesuai
97	Jiharka	Ji007531.wav	Sesuai
98	Jiharka	Ji008532.wav	Sesuai
99	Jiharka	Ji009531.wav	Sesuai
100	Jiharka	Ji010531.wav	Sesuai
101	Nahawan	Na001531.wav	Sesuai
102	Nahawan	Na002531.wav	Sesuai
103	Nahawan	Na003531.wav	Sesuai
104	Nahawan	Na004531.wav	Sesuai
105	Nahawan	Na005531.wav	Sesuai
106	Nahawan	Na006532.wav	Sesuai
107	Nahawan	Na007531.wav	Sesuai
108	Nahawan	Na008531.wav	Sesuai
109	Nahawan	Na009532.wav	Sesuai
110	Nahawan	Na010532.wav	Sesuai
111	Rost	Ro001532.wav	Sesuai
112	Rost	Ro002531.wav	Sesuai
113	Rost	Ro003531.wav	Sesuai
114	Rost	Ro004531.wav	Sesuai
115	Rost	Ro005531.wav	Sesuai
116	Rost	Ro006532.wav	Sesuai
117	Rost	Ro007532.wav	Sesuai

Data Ke	Jenis Irama	Nama File.*wav	Hasil Jenis Irama
118	Rost	Ro008531.wav	Sesuai
119	Rost	Ro009531.wav	Sesuai
120	Rost	Ro010531.wav	Sesuai
121	Shoba	Sh001531.wav	Sesuai
122	Shoba	Sh002531.wav	Sesuai
123	Shoba	Sh003531.wav	Sesuai
124	Shoba	Sh004532.wav	Sesuai
125	Shoba	Sh005531.wav	Sesuai
126	Shoba	Sh006532.wav	Sesuai
127	Shoba	Sh007532.wav	Sesuai
128	Shoba	Sh008532.wav	Sesuai
129	Shoba	Sh009532.wav	Sesuai
130	Shoba	Sh010532.wav	Sesuai
131	Sikah	Si001532.wav	Sesuai
132	Sikah	Si002531.wav	Sesuai
133	Sikah	Si003532.wav	Sesuai
134	Sikah	Si004531.wav	Sesuai
135	Sikah	Si005531.wav	Sesuai
136	Sikah	Si006531.wav	Sesuai
137	Sikah	Si007531.wav	Sesuai
138	Sikah	Si008531.wav	Sesuai
139	Sikah	Si009531.wav	Sesuai
140	Sikah	Si010532.wav	Sesuai
141	Bayyati	Ba001821.wav	Sesuai
142	Bayyati	Ba002821.wav	Sesuai
143	Bayyati	Ba003822.wav	Sesuai
144	Bayyati	Ba004821.wav	Sesuai
145	Bayyati	Ba005821.wav	Sesuai
146	Bayyati	Ba006822.wav	Sesuai
147	Bayyati	Ba007821.wav	Sesuai
148	Bayyati	Ba008821.wav	Sesuai
149	Bayyati	Ba009822.wav	Sesuai
150	Bayyati	Ba010822.wav	Sesuai
151	Hijaz	Hi001821.wav	Sesuai
152	Hijaz	Hi002821.wav	Sesuai
153	Hijaz	Hi003822.wav	Sesuai
154	Hijaz	Hi004821.wav	Sesuai
155	Hijaz	Hi005822.wav	Sesuai
156	Hijaz	Hi006822.wav	Sesuai
157	Hijaz	Hi007821.wav	Sesuai
158	Hijaz	Hi008822.wav	Sesuai
159	Hijaz	Hi009821.wav	Sesuai
160	Hijaz	Hi010822.wav	Sesuai
161	Jiharka	Ji001821.wav	Sesuai
162	Jiharka	Ji002822.wav	Sesuai
163	Jiharka	Ji003821.wav	Sesuai

Data Ke	Jenis Irama	Nama File.*wav	Hasil Jenis Irama
164	Jiharka	Ji004821.wav	Sesuai
165	Jiharka	Ji005821.wav	Sesuai
166	Jiharka	Ji006821.wav	Sesuai
167	Jiharka	Ji007822.wav	Sesuai
168	Jiharka	Ji008821.wav	Sesuai
169	Jiharka	Ji009822.wav	Sesuai
170	Jiharka	Ji010822.wav	Sesuai
171	Nahawan	Na001821.wav	Sesuai
172	Nahawan	Na002821.wav	Sesuai
173	Nahawan	Na003822.wav	Sesuai
174	Nahawan	Na004821.wav	Sesuai
175	Nahawan	Na005822.wav	Sesuai
176	Nahawan	Na006822.wav	Sesuai
177	Nahawan	Na007821.wav	Sesuai
178	Nahawan	Na008821.wav	Sesuai
179	Nahawan	Na009822.wav	Sesuai
180	Nahawan	Na010821.wav	Sesuai
181	Rost	Ro001822.wav	Sesuai
182	Rost	Ro002821.wav	Sesuai
183	Rost	Ro003821.wav	Sesuai
184	Rost	Ro004821.wav	Sesuai
185	Rost	Ro005821.wav	Sesuai
186	Rost	Ro006821.wav	Sesuai
187	Rost	Ro007821.wav	Sesuai
188	Rost	Ro008821.wav	Sesuai
189	Rost	Ro009821.wav	Sesuai
190	Rost	Ro010822.wav	Sesuai
191	Shoba	Sh001822.wav	Sesuai
192	Shoba	Sh002822.wav	Sesuai
193	Shoba	Sh003821.wav	Sesuai
194	Shoba	Sh004821.wav	Sesuai
195	Shoba	Sh005821.wav	Sesuai
196	Shoba	Sh006822.wav	Sesuai
197	Shoba	Sh007822.wav	Sesuai
198	Shoba	Sh008821.wav	Sesuai
199	Shoba	Sh009821.wav	Sesuai
200	Shoba	Sh010822.wav	Sesuai
201	Sikah	Si001821.wav	Sesuai
202	Sikah	Si002821.wav	Sesuai
203	Sikah	Si003822.wav	Sesuai
204	Sikah	Si004822.wav	Sesuai
205	Sikah	Si005821.wav	Sesuai
206	Sikah	Si006822.wav	Sesuai
207	Sikah	Si007821.wav	Sesuai
208	Sikah	Si008821.wav	Sesuai
209	Sikah	Si009822.wav	Sesuai

Data Ke	Jenis Irama	Nama File.*wav	Hasil Jenis Irama
210	Sikah	Si010822.wav	Sesuai

C. Data Training Rekaman Qiro'ah Berjumlah 210 Sampel Data.xls

No.	Jenis Irama	Nama File.*wav	Amplitudo Min	Amplitudo Max	Sudut Turun / 10	Sudut Naik / 10	Sudut Turun	Sudut Naik	Amplitudo Max * 1000
1	Bayyati	Ba001433.wav	78	1049,265	309,1	310,5	3091	3105	1049265
2	Bayyati	Ba002432.wav	83	850,624	380,1	356,7	3801	3567	850624
3	Bayyati	Ba003431.wav	4	1024,536	401	399,3	4010	3993	1024536
4	Bayyati	Ba004432.wav	15	891,36	314,7	334,2	3147	3342	891360
5	Bayyati	Ba005432.wav	99	928,94	363,1	372,6	3631	3726	928940
6	Bayyati	Ba006431.wav	100	922,432	359,4	371,1	3594	3711	922432
7	Bayyati	Ba007432.wav	87	858,949	306,7	317,7	3067	3177	858949
8	Bayyati	Ba008432.wav	18	868,534	440,3	459,9	4403	4599	868534
9	Bayyati	Ba009431.wav	71	927,119	336,7	342,2	3367	3422	927119
10	Bayyati	Ba010431.wav	5	935,822	357,5	352,7	3575	3527	935822
11	Hijaz	Hi001433.wav	94	876,828	373,1	381,6	3731	3816	876828
12	Hijaz	Hi002431.wav	70	888,642	467,6	436,8	4676	4368	888642
13	Hijaz	Hi003432.wav	21	925,725	446,6	443,3	4466	4433	925725
14	Hijaz	Hi004431.wav	28	869,812	326,5	348,2	3265	3482	869812
15	Hijaz	Hi005431.wav	43	942,577	435,2	440,9	4352	4409	942577
16	Hijaz	Hi006432.wav	76	961,022	400,1	409	4001	4090	961022
17	Hijaz	Hi007431.wav	87	784,096	314,5	320,7	3145	3207	784096
18	Hijaz	Hi008432.wav	90	850,237	426,5	440,6	4265	4406	850237
19	Hijaz	Hi009431.wav	95	896,327	348,4	360	3484	3600	896327
20	Hijaz	Hi010431.wav	9	854,444	430,8	464	4308	4640	854444
21	Jiharka	Ji001432.wav	59	889,404	340,3	345,9	3403	3459	889404
22	Jiharka	Ji002431.wav	51	791,803	287,6	268,9	2876	2689	791803
23	Jiharka	Ji003431.wav	49	905,624	444,2	463,7	4442	4637	905624
24	Jiharka	Ji004431.wav	38	869,832	256	281,8	2560	2818	869832
25	Jiharka	Ji005432.wav	20	860,927	375	384,8	3750	3848	860927
26	Jiharka	Ji006431.wav	47	871,304	428,8	439	4288	4390	871304
27	Jiharka	Ji007431.wav	31	803,619	391	381,1	3910	3811	803619
28	Jiharka	Ji008431.wav	52	825,342	311,9	327,9	3119	3279	825342
29	Jiharka	Ji009431.wav	43	871,417	349,9	364,6	3499	3646	871417
30	Jiharka	Ji010431.wav	83	1002,182	373	401,2	3730	4012	1002182
31	Nahawan	Na001431.wav	68	858,327	340,6	329,9	3406	3299	858327
32	Nahawan	Na002431.wav	87	849,752	270,4	249,9	2704	2499	849752
33	Nahawan	Na003432.wav	85	941,395	379,5	403	3795	4030	941395
34	Nahawan	Na004432.wav	41	820,307	255,9	284,5	2559	2845	820307
35	Nahawan	Na005431.wav	83	870,395	330,1	341,8	3301	3418	870395
36	Nahawan	Na006431.wav	8	838,052	385	406,5	3850	4065	838052
37	Nahawan	Na007431.wav	83	766,759	344,2	348,2	3442	3482	766759
38	Nahawan	Na008431.wav	60	791,73	356,2	381,2	3562	3812	791730
39	Nahawan	Na009431.wav	26	893,355	354,5	364,7	3545	3647	893355
40	Nahawan	Na010431.wav	24	825,724	371,8	406,1	3718	4061	825724
41	Rost	Ro001431.wav	87	852,445	415,2	421,2	4152	4212	852445
42	Rost	Ro002432.wav	55	913,696	434,4	407,6	4344	4076	913696
43	Rost	Ro003431.wav	7	910,03	430,2	451,2	4302	4512	910030
44	Rost	Ro004432.wav	13	731,461	306,9	333,7	3069	3337	731461
45	Rost	Ro005431.wav	52	823,845	442,7	455,8	4427	4558	823845
46	Rost	Ro006432.wav	48	902,871	441,9	449	4419	4490	902871
47	Rost	Ro007432.wav	24	751,463	315	322	3150	3220	751463
48	Rost	Ro008431.wav	43	873,882	452,8	470,1	4528	4701	873882
49	Rost	Ro009431.wav	11	823,8	480,3	481,5	4803	4815	823800
50	Rost	Ro010432.wav	20	1048,123	380,8	394,3	3808	3943	1048123
51	Shoba	Sh001433.wav	21	898,773	330,5	328,6	3305	3286	898773
52	Shoba	Sh002431.wav	6	817,977	324,1	301,7	3241	3017	817977
53	Shoba	Sh003432.wav	98	921,769	420,8	446,8	4208	4468	921769
54	Shoba	Sh004431.wav	7	920,454	317,7	344,5	3177	3445	920454
55	Shoba	Sh005432.wav	14	891,712	390,5	407,2	3905	4072	891712

No.	Jenis Irama	Nama File. *wav	Amplitudo Min	Amplitudo Max	Sudut Turun / 10	Sudut Naik / 10	Sudut Turun	Sudut Naik	Amplitudo Max * 1000
56	Shoba	Sh006431.wav	80	924,585	428,6	434,5	4286	4345	924585
57	Shoba	Sh007432.wav	40	761,357	298,7	307,5	2987	3075	761357
58	Shoba	Sh008432.wav	69	754,507	369,8	398,9	3698	3989	754507
59	Shoba	Sh009432.wav	95	905,78	381	385	3810	3850	905780
60	Shoba	Sh010432.wav	54	906,393	400,9	436,1	4009	4361	906393
61	Sikah	Si001431.wav	10	865,015	303,8	301,5	3038	3015	865015
62	Sikah	Si002432.wav	79	854,562	250,8	231,9	2508	2319	854562
63	Sikah	Si003432.wav	90	906,537	445,5	466,4	4455	4664	906537
64	Sikah	Si004432.wav	73	823,922	292,4	322,7	2924	3227	823922
65	Sikah	Si005431.wav	68	932,919	369,3	379,7	3693	3797	932919
66	Sikah	Si006432.wav	90	849,991	387,2	397,2	3872	3972	849991
67	Sikah	Si007432.wav	91	848,358	322,3	323,9	3223	3239	848358
68	Sikah	Si008432.wav	35	811,961	336,6	362,7	3366	3627	811961
69	Sikah	Si009431.wav	64	894,138	344,5	353,1	3445	3531	894138
70	Sikah	Si010432.wav	80	871,807	375,2	402,7	3752	4027	871807
71	Bayyati	Ba001531.wav	25	966,367	315,5	317,7	3155	3177	966367
72	Bayyati	Ba002531.wav	41	966,483	257,3	239,7	2573	2397	966483
73	Bayyati	Ba003532.wav	38	972,302	340,9	361,8	3409	3618	972302
74	Bayyati	Ba004532.wav	69	939,652	320,2	346	3202	3460	939652
75	Bayyati	Ba005531.wav	38	970,992	250	273,5	2500	2735	970992
76	Bayyati	Ba006532.wav	60	975,216	324	333,4	3240	3334	975216
77	Bayyati	Ba007532.wav	40	986,288	320,4	323	3204	3230	986288
78	Bayyati	Ba008532.wav	92	975,103	387,1	407,2	3871	4072	975103
79	Bayyati	Ba009532.wav	97	992,395	259	259,2	2590	2592	992395
80	Bayyati	Ba010532.wav	86	1094,393	374,5	412,4	3745	4124	1094393
81	Hijaz	Hi001531.wav	23	917,793	374,2	376	3742	3760	917793
82	Hijaz	Hi002531.wav	66	949,117	357,7	338,5	3577	3385	949117
83	Hijaz	Hi003531.wav	24	1038,729	335,3	354,1	3353	3541	1038729
84	Hijaz	Hi004532.wav	97	929,067	260,8	291,2	2608	2912	929067
85	Hijaz	Hi005531.wav	90	991,915	272,9	293,1	2729	2931	991915
86	Hijaz	Hi006532.wav	18	869,331	364,1	372,4	3641	3724	869331
87	Hijaz	Hi007531.wav	85	884,192	343,4	349,6	3434	3496	884192
88	Hijaz	Hi008532.wav	93	943,245	424,9	445,5	4249	4455	943245
89	Hijaz	Hi009531.wav	97	936,09	283,5	293,5	2835	2935	936090
90	Hijaz	Hi010532.wav	35	1004,418	395,8	425,6	3958	4256	1004418
91	Jiharka	Ji001532.wav	68	935,434	294,2	286,4	2942	2864	935434
92	Jiharka	Ji002531.wav	70	920,922	245,6	218,5	2456	2185	920922
93	Jiharka	Ji003531.wav	25	1039,378	306,9	331,1	3069	3311	1039378
94	Jiharka	Ji004531.wav	41	989,947	240	268,4	2400	2684	989947
95	Jiharka	Ji005531.wav	18	879,471	299,8	312,5	2998	3125	879471
96	Jiharka	Ji006532.wav	25	879,801	341,1	357,1	3411	3571	879801
97	Jiharka	Ji007531.wav	28	852,19	357	352,4	3570	3524	852190
98	Jiharka	Ji008532.wav	54	860,495	345,5	370,5	3455	3705	860495
99	Jiharka	Ji009531.wav	16	959,452	303,4	299,2	3034	2992	959452
100	Jiharka	Ji010531.wav	72	1007,318	360,6	379,4	3606	3794	1007318
101	Nahawan	Na001531.wav	82	894,587	277,3	264,7	2773	2647	894587
102	Nahawan	Na002531.wav	85	905,985	250,5	224,7	2505	2247	905985
103	Nahawan	Na003531.wav	4	985,53	285,1	307,5	2851	3075	985530
104	Nahawan	Na004531.wav	60	960,205	237	267,4	2370	2674	960205
105	Nahawan	Na005531.wav	33	877,559	315	323,7	3150	3237	877559
106	Nahawan	Na006532.wav	40	887,924	307,1	317,9	3071	3179	887924
107	Nahawan	Na007531.wav	34	882,449	330,5	336,3	3305	3363	882449
108	Nahawan	Na008531.wav	28	898,562	302	312,6	3020	3126	898562
109	Nahawan	Na009532.wav	15	941,94	259,4	261,3	2594	2613	941940
110	Nahawan	Na010532.wav	94	948,273	318,8	337	3188	3370	948273
111	Rost	Ro001532.wav	67	908,352	293,5	298,4	2935	2984	908352
112	Rost	Ro002531.wav	100	992,371	337,8	320,1	3378	3201	992371
113	Rost	Ro003531.wav	84	1022,177	399,8	420,9	3998	4209	1022177
114	Rost	Ro004531.wav	4	951,214	292,9	310,8	2929	3108	951214
115	Rost	Ro005531.wav	1	907,992	299,7	312,4	2997	3124	907992
116	Rost	Ro006532.wav	54	872,564	366,8	368,9	3668	3689	872564

No.	Jenis Irama	Nama File. *wav	Amplitudo Min	Amplitudo Max	Sudut Turun / 10	Sudut Naik / 10	Sudut Turun	Sudut Naik	Amplitudo Max * 1000
117	Rost	Ro007532.wav	25	873,851	312,5	317,8	3125	3178	873851
118	Rost	Ro008531.wav	8	934,284	428,4	444,1	4284	4441	934284
119	Rost	Ro009531.wav	43	870,873	364,6	365,5	3646	3655	870873
120	Rost	Ro010531.wav	42	1124,557	352,1	377,6	3521	3776	1124557
121	Shoba	Sh001531.wav	95	882,161	293,6	283,8	2936	2838	882161
122	Shoba	Sh002531.wav	43	880,465	259,6	243,1	2596	2431	880465
123	Shoba	Sh003531.wav	81	1003,059	315,2	336	3152	3360	1003059
124	Shoba	Sh004532.wav	80	951,005	255,7	282,4	2557	2824	951005
125	Shoba	Sh005531.wav	39	940,327	293,5	315,3	2935	3153	940327
126	Shoba	Sh006532.wav	96	915,26	331	343	3310	3430	915260
127	Shoba	Sh007532.wav	9	907,751	333,7	339,4	3337	3394	907751
128	Shoba	Sh008532.wav	11	922,866	276,9	295,7	2769	2957	922866
129	Shoba	Sh009532.wav	26	987,366	275,1	280,9	2751	2809	987366
130	Shoba	Sh010532.wav	56	1025,826	389,8	408,8	3898	4088	1025826
131	Sikah	Si001532.wav	44	883,851	264,1	251	2641	2510	883851
132	Sikah	Si002531.wav	71	923,467	233,7	214,3	2337	2143	923467
133	Sikah	Si003532.wav	41	963,267	312,6	339,1	3126	3391	963267
134	Sikah	Si004531.wav	4	1005,456	275,3	312,4	2753	3124	1005456
135	Sikah	Si005531.wav	35	957,457	310	330,8	3100	3308	957457
136	Sikah	Si006531.wav	60	1024,827	310,4	321,7	3104	3217	1024827
137	Sikah	Si007531.wav	68	905,949	310,1	314,3	3101	3143	905949
138	Sikah	Si008531.wav	46	938,016	279,3	302,4	2793	3024	938016
139	Sikah	Si009531.wav	21	989,572	239,5	250,5	2395	2505	989572
140	Sikah	Si010532.wav	61	941,89	314,7	343,8	3147	3438	941890
141	Bayyati	Ba001821.wav	38	1173,619	310,2	306,3	3102	3063	1173619
142	Bayyati	Ba002821.wav	61	1375,999	258,7	241,2	2587	2412	1375999
143	Bayyati	Ba003822.wav	99	1363,943	252,5	271	2525	2710	1363943
144	Bayyati	Ba004821.wav	97	1194,862	336,5	362,2	3365	3622	1194862
145	Bayyati	Ba005821.wav	63	1347,551	286	315,3	2860	3153	1347551
146	Bayyati	Ba006822.wav	77	1187,414	325,9	335,1	3259	3351	1187414
147	Bayyati	Ba007821.wav	89	1154,582	322,9	327	3229	3270	1154582
148	Bayyati	Ba008821.wav	30	1175,111	312,8	299,3	3128	2993	1175111
149	Bayyati	Ba009822.wav	93	1305,549	292,5	303,4	2925	3034	1305549
150	Bayyati	Ba010822.wav	94	1460,585	222,5	251,5	2225	2515	1460585
151	Hijaz	Hi001821.wav	20	1135,397	317,5	318	3175	3180	1135397
152	Hijaz	Hi002821.wav	91	1366,384	317,9	300,3	3179	3003	1366384
153	Hijaz	Hi003822.wav	23	1384,206	242,8	262,3	2428	2623	1384206
154	Hijaz	Hi004821.wav	37	1163,013	292	317,2	2920	3172	1163013
155	Hijaz	Hi005822.wav	76	1369,088	254,4	267,3	2544	2673	1369088
156	Hijaz	Hi006822.wav	98	1252,812	288,6	289,7	2886	2897	1252812
157	Hijaz	Hi007821.wav	3	1173,552	323,7	339,1	3237	3391	1173552
158	Hijaz	Hi008822.wav	5	1189,963	284,6	274,9	2846	2749	1189963
159	Hijaz	Hi009821.wav	34	1249,917	319,4	317,3	3194	3173	1249917
160	Hijaz	Hi010822.wav	80	1306,648	265,5	291,9	2655	2919	1306648
161	Jiharka	Ji001821.wav	47	1221,915	296,1	281,4	2961	2814	1221915
162	Jiharka	Ji002822.wav	3	1245,042	282,1	257,9	2821	2579	1245042
163	Jiharka	Ji003821.wav	58	1345,871	251,8	272,5	2518	2725	1345871
164	Jiharka	Ji004821.wav	94	1232,747	228,1	258,3	2281	2583	1232747
165	Jiharka	Ji005821.wav	1	1314,288	298,2	313,9	2982	3139	1314288
166	Jiharka	Ji006821.wav	52	1176,32	357,7	359,4	3577	3594	1176320
167	Jiharka	Ji007822.wav	97	1141,403	360,9	361,3	3609	3613	1141403
168	Jiharka	Ji008821.wav	87	1252,053	349,1	342,2	3491	3422	1252053
169	Jiharka	Ji009822.wav	23	1222,144	326,3	324,9	3263	3249	1222144
170	Jiharka	Ji010822.wav	53	1178,537	349,9	378,6	3499	3786	1178537
171	Nahawan	Na001821.wav	67	1188,874	273,4	273,1	2734	2731	1188874
172	Nahawan	Na002821.wav	33	1369,474	284,7	265,1	2847	2651	1369474
173	Nahawan	Na003822.wav	43	1394,551	257,5	276	2575	2760	1394551
174	Nahawan	Na004821.wav	32	1332,826	291,3	314,3	2913	3143	1332826
175	Nahawan	Na005822.wav	93	1294,062	295,6	302,9	2956	3029	1294062
176	Nahawan	Na006822.wav	60	1202,659	344,4	350,7	3444	3507	1202659
177	Nahawan	Na007821.wav	86	1082,932	288,4	283,2	2884	2832	1082932

No.	Jenis Irama	Nama File. *wav	Amplitudo Min	Amplitudo Max	Sudut Turun / 10	Sudut Naik / 10	Sudut Turun	Sudut Naik	Amplitudo Max * 1000
178	Nahawan	Na008821.wav	41	1227,203	353,7	342,7	3537	3427	1227203
179	Nahawan	Na009822.wav	51	1340,237	298,6	303,3	2986	3033	1340237
180	Nahawan	Na010821.wav	8	1225,037	332,4	363,5	3324	3635	1225037
181	Rost	Ro001822.wav	44	1125,077	303,7	308,4	3037	3084	1125077
182	Rost	Ro002821.wav	45	1326,26	290,8	277,9	2908	2779	1326260
183	Rost	Ro003821.wav	44	1374,604	171,8	182,2	1718	1822	1374604
184	Rost	Ro004821.wav	17	1332,826	291,3	314,3	2913	3143	1332826
185	Rost	Ro005821.wav	45	1323,686	260,1	264,4	2601	2644	1323686
186	Rost	Ro006821.wav	81	1234,957	289,7	288,2	2897	2882	1234957
187	Rost	Ro007821.wav	90	1136,96	340,9	355,9	3409	3559	1136960
188	Rost	Ro008821.wav	32	1258,893	239,5	229,3	2395	2293	1258893
189	Rost	Ro009821.wav	13	1209,372	315,9	320,6	3159	3206	1209372
190	Rost	Ro010822.wav	20	1283,484	253,7	269,9	2537	2699	1283484
191	Shoba	Sh001822.wav	74	1203,66	282,1	277,7	2821	2777	1203660
192	Shoba	Sh002822.wav	11	1258,854	271,6	244,9	2716	2449	1258854
193	Shoba	Sh003821.wav	95	1355,33	229,3	241,6	2293	2416	1355330
194	Shoba	Sh004821.wav	68	1214,173	250,3	269,8	2503	2698	1214173
195	Shoba	Sh005821.wav	4	1376,298	296,1	310	2961	3100	1376298
196	Shoba	Sh006822.wav	71	1202,239	324	335,2	3240	3352	1202239
197	Shoba	Sh007822.wav	46	1154,033	337	342,3	3370	3423	1154033
198	Shoba	Sh008821.wav	23	1169,307	370,5	356,3	3705	3563	1169307
199	Shoba	Sh009821.wav	21	1246,044	319,1	324,1	3191	3241	1246044
200	Shoba	Sh010822.wav	74	1215,202	268	294,7	2680	2947	1215202
201	Sikah	Si001821.wav	27	1179,975	274,4	271,8	2744	2718	1179975
202	Sikah	Si002821.wav	29	1309,245	253,6	228,3	2536	2283	1309245
203	Sikah	Si003822.wav	16	1365,873	253,2	277,1	2532	2771	1365873
204	Sikah	Si004822.wav	99	1261,86	266,8	303,6	2668	3036	1261860
205	Sikah	Si005821.wav	34	1345,495	257,6	269,7	2576	2697	1345495
206	Sikah	Si006822.wav	85	1222,578	300,1	307,3	3001	3073	1222578
207	Sikah	Si007821.wav	6	1178,527	323,2	319,4	3232	3194	1178527
208	Sikah	Si008821.wav	95	1249,901	328,2	315,6	3282	3156	1249901
209	Sikah	Si009822.wav	71	1304,786	319,9	320,4	3199	3204	1304786
210	Sikah	Si010822.wav	2	1219,868	332,9	347,3	3329	3473	1219868

D. Source Code Aplikasi

```
package nbayesqori;

import java.io.File;
import java.io.IOException;
import java.util.Arrays;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.UnsupportedAudioFileException;

/**
 *
 * @author Toto Ricky
 */
public class getNaikTurun {
```

```

/**
 * @param args the command line arguments
 */
static int sudutNaik = 0;
static int sudutTurun = 0;

public void getNaikTurun(String path) throws
UnsupportedAudioFormatException, IOException, LineUnavailableException {

    WaveFile2 wav = new WaveFile2(new File(path));

    // int amplitudeExample = wav.getSampleInt(140); // 140th
    amplitude value.
    // System.out.println("sample : " + amplitudeExample);
    int jumlahTot = 0;

    //int naik = (int)
    int panjang = (int) wav.getFramesCount();

    //amplitudeMin=wav.getSampleInt(1);
    int arr[] = new int[panjang];
    int arr_naik[] = new int[panjang];
    int arr_turun[] = new int[panjang];

    for (int i = 0; i < wav.getFramesCount(); i++) {

        arr[i] = wav.getSampleInt(i);
        if (i != wav.getFramesCount() - 1) {
            if (wav.getSampleInt(i) > wav.getSampleInt(i + 1)) {
                int selisih = wav.getSampleInt(i) -
wav.getSampleInt(i + 1);
                arr_turun[i] = selisih;
            }
            if (wav.getSampleInt(i) < wav.getSampleInt(i + 1)) {
                int selisih = wav.getSampleInt(i + 1) -
wav.getSampleInt(i);
                arr_naik[i] = selisih;
            }
        }
    }

    int jumlahArrNaik = 0;
    int panjangArrNaik = 0;
    for(int a =0; a<arr_naik.length;a++){
        if(arr_naik[a]!=0){
            jumlahArrNaik = jumlahArrNaik + arr_naik[a];
            panjangArrNaik++;
        }
    }
    sudutNaik = Math.abs(jumlahArrNaik/panjangArrNaik);

    int jumlahArrTurun = 0;
    int panjangArrTurun = 0;
    for(int a =0; a<arr_turun.length;a++){
        if(arr_turun[a]!=0){
            jumlahArrTurun = jumlahArrTurun + arr_turun[a];
            panjangArrTurun++;
        }
    }
    sudutTurun = Math.abs(jumlahArrTurun/panjangArrTurun);

    System.out.println("panjang " + panjang);
    System.out.println("panjang naik : " + panjangArrNaik);
}

```

```

System.out.println("panjang turun : " + panjangArrTurun);
System.out.println("rata2 naik : " + sudutTurun);
System.out.println("rata2 turun : "+ sudutNaik);

Arrays.sort(arr);

```

```

package nbayesqori;

import java.io.File;
import java.io.IOException;
import java.util.Arrays;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.UnsupportedAudioFileException;

/**
 *
 * @author Toto Ricky
 */
public class getAmplitudo {

    /**
     * @param args the command line arguments
     */
    static int amplitudeMax = 0;
    static int amplitudeMin = 1000000000;
    static int amplitudeAvg = 0;

    public void getAmp(String path) throws UnsupportedAudioFileException,
    IOException, LineUnavailableException {

        WaveFile2 wav = new WaveFile2(new File(path));
        int jumlahTot = 0;

        int panjang = (int) wav.getFramesCount();
        amplitudeMax=panjang;
        int arr[] = new int[panjang];

        for (int i = 0; i < wav.getFramesCount(); i++) {
            int amplitude = wav.getSampleInt(i);

            arr[i] = wav.getSampleInt(i);

            if (amplitude < amplitudeMin && amplitude>0) {
                amplitudeMin = amplitude;
            }
            if (amplitude >= 0) {
                jumlahTot = jumlahTot + amplitude;
            }
        }
        Arrays.sort(arr);

        amplitudeAvg = (int) (jumlahTot / wav.getFramesCount());
        System.out.println("count : " + wav.getFramesCount());
        System.out.println("amplitude max : " + amplitudeMax);
        System.out.println("amplitude min : " + amplitudeMin);
        System.out.println("rata-rata : " + amplitudeAvg);
    }

    public static void main(String[] args) {
        try {
            getAmplitudo GA = new getAmplitudo();
            GA.getAmp("D://Hi002821.wav");
        }
    }
}

```

```

    } catch (Exception e) {
        System.err.println(e);
    }
}

```

```

package nbayesqori;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Arrays;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.UnsupportedAudioFileException;

/**
 *
 * @author Toto Ricky
 */
public class getAmplitudo_1 {

    /**
     * @param args the command line arguments
     */
    static int amplitudoMax = 0;
    static int amplitudoMin = 1000000000;
    static int amplitudoAvg = 0;

    public static void main(String[] args) {
        try {
            getAmplitudo_1 gt = new getAmplitudo_1();
            gt.getAmp();
        } catch (Exception e) {
            System.err.println(e);
        }
    }

    public void getAmp() throws UnsupportedAudioFileException,
        IOException, LineUnavailableException {
        for (int a = 1; a <= 10; a++) {
            String b = "D://Ayat43/Si0" + a + ".wav";

            WaveFile2 wav = new WaveFile2(new File(b));
            int panjang = (int) wav.getFramesCount();
            int arr[] = new int[panjang];

            String c = "D://Ayat43/Si0" + a + ".txt";
            System.out.println(c);
            File file = new File(c);
            file.delete();
            FileWriter writer = new FileWriter(c, true);
            BufferedWriter bw = new BufferedWriter(writer);

            for (int i = 0; i < wav.getFramesCount(); i++) {
                arr[i] = (wav.getSampleInt(i));
            }

            for (int i = 0; i < arr.length; i++) {
                bw.write(String.valueOf(arr[i]));
                bw.newLine();
            }
            bw.close();
        }
    }
}

```

```
}

```

```
public static void main(String[] args) throws IOException {
    }
    public static int hitungJumlah(String[] arrVariabel, String nilaiVar)
    {
        int hitung = 0;
        for (int a = 0; a < arrVariabel.length; a++) {
            if (arrVariabel[a].equals(nilaiVar)) {
                hitung++;
            }
        }
        return hitung;
    }
    public static double hitungMean(double[] atribut) {
        double avg = 0;
        double jumlah = 0;
        for (int a = 0; a < atribut.length; a++) {
            jumlah = jumlah + atribut[a];
        }
        avg = jumlah / atribut.length;
        return avg;
    }
    public static double hitungSDeviasi(double[] atribut, double mean) {
        double avg = mean;
        double sDeviasi = 0;
        double jumlah = 0;
        for (int i = 0; i < atribut.length; i++) {
            atribut[i] = Math.pow((atribut[i] - avg), 2);
            jumlah = jumlah + atribut[i];
        }
        sDeviasi = jumlah / atribut.length;
        return sDeviasi;
    }
    public static double hitungDistribusiNormal(double x, double mean,
        double sDeviasi) {
        double hasil = 5;
        double phi = 3.14159;
        double e = 2.718;
        hasil = (1 / (sDeviasi * Math.sqrt(2 * phi))) * Math.pow(e, -
            (Math.pow(x - mean, 2)) / (2 * Math.pow(sDeviasi, 2)));
        return hasil;
    }
    public static double hitungProbabilitasAtribut_Mean(double[] atribut,
        String[] Irama, String irama) {
        double hitungMean = 0;
        int i = 0;
        double jumlah = 0;
        for (int a = 0; a < Irama.length; a++) {
            if (Irama[a].equals(irama)) {
                jumlah = jumlah + atribut[a];
                i++;
            }
        }
        hitungMean = jumlah / i;
        return hitungMean;
    }
    public static double hitungProbabilitasAtribut_sDeviasi(double[]
        atribut, String[] Irama, String irama, double mean) {

        double avg = mean;
        double sDeviasi = 0;
        double jumlah = 0;
        int i = 0;
        for (int a = 0; a < Irama.length; a++) {

```

```

        if (Irama[a].equals(irama)) {
            atribut[a] = Math.pow((atribut[a] - avg), 2);
            jumlah = jumlah + atribut[a];
            i++;
        }
    }
    sDeviasi = jumlah / atribut.length;
    return sDeviasi;
}

public static double P_irama(int jum_a, int jum_b, int jum_c, int
jum_d, int jum_e, int jum_f, int jum_g) {
    double Pirama = (double) jum_a / (jum_a + jum_b + jum_c + jum_d +
jum_e + jum_f + jum_g);
    return Pirama;
}

public static double P_atribut(int jumlah_a, int jumlah_b) {
    double Pclass = (double) jumlah_a / (jumlah_b);
    return Pclass;
}
}

```

```

package nbayesqori;

/**
 *
 * @author Toto Ricky
 */
import java.io.File;
import java.io.IOException;
import javax.sound.sampled.AudioFormat;
import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.DataLine;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.SourceDataLine;

public class putarWAV {
    private final int BUFFER_SIZE = 128000;
    private File soundFile;
    private AudioInputStream audioStream;
    private AudioFormat audioFormat;
    private SourceDataLine sourceLine;

    /**
     * @param filename the name of the file that is going to be played
     */

    public static void main(String[] args) {
        new putarWAV().playSound("D://Ba001433.wav");
    }

    public void playSound(String filename){

        String strFilename = filename;
        try {
            soundFile = new File(strFilename);
        } catch (Exception e) {
            e.printStackTrace();
            System.exit(1);
        }
        try {
            audioStream = AudioSystem.getAudioInputStream(soundFile);
        } catch (Exception e){
            e.printStackTrace();
            System.exit(1);
        }
        audioFormat = audioStream.getFormat();
    }
}

```

```

        DataLine.Info info = new DataLine.Info(SourceDataLine.class,
audioFormat);
        try {
            sourceLine = (SourceDataLine) AudioSystem.getLine(info);
            sourceLine.open(audioFormat);
        } catch (LineUnavailableException e) {
            e.printStackTrace();
            System.exit(1);
        } catch (Exception e) {
            e.printStackTrace();
            System.exit(1);
        }

        sourceLine.start();
        int nBytesRead = 0;
        byte[] abData = new byte[BUFFER_SIZE];
        while (nBytesRead != -1) {
            try {
                nBytesRead = audioStream.read(abData, 0, abData.length);
            } catch (IOException e) {
                e.printStackTrace();
            }
            if (nBytesRead >= 0) {
                @SuppressWarnings("unused")
                int nBytesWritten = sourceLine.write(abData, 0,
nBytesRead);
            }
            sourceLine.drain();
            sourceLine.close();
        }
    }
}

```