

**PENGEMBANGAN LANJUT APLIKASI SCORING IQRA
BERBASIS ANDROID DENGAN PARAMETER PITCH, VOLUME,
RHYTHM, DAN TIMBRE**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Wahyu Septiadi

NIM: 175150201111015



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2021**



PENGESAHAN

PENGEMBANGAN APLIKASI SCORING IQRA BERBASIS ANDROID DENGAN
PARAMETER *PITCH, VOLUME, RHYTHM, DAN TIMBRE*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Wahyu Septiadi

NIM: 175150201111015

Skripsi ini telah diuji dan dinyatakan lulus pada

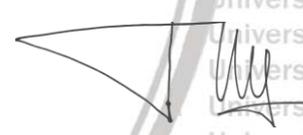
1 Juli 2021

Telah diperiksa dan disetujui oleh:

Pembimbing I

Pembimbing II


Komang Candra Brata, S.Kom., M.T.,
M.Sc.


Adam Hendra Brata, S.Kom., M.T.,
M.Sc.

NIK: 2016078907111001

NIP: 199001052019031009

Mengetahui

Ketua Jurusan Teknik Informatika




Achmad Basuki, S.T., M.MG., Ph.D.

NIP: 197411182003121002

PERNYATAAN ORISINALITAS

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

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

Malang, 12 Juli 2021



Wahyu Septiadi

NIM: 175150201111015

ABSTRAK

Wahyu Septiadi, Pengembangan Lanjut Aplikasi *Scoring* Iqra Berbasis Android dengan Parameter *Pitch, Volume, Rhythm, dan Timbre*

Pembimbing: Komang Candra Brata, S.Kom., M.T., M.Sc. dan Adam Hendra Brata, S.Kom., M.T., M.Sc.

Membaca Al-Qur'an merupakan salah satu kewajiban bagi setiap umat Islam. Sebelum bisa membaca Al-Qur'an dengan lafal dan pengucapan yang benar, diperlukan latihan membaca huruf ataupun kalimat hijaiyah. Salah satu metode belajar Al-Qur'an yaitu menggunakan metode Iqra yang ditemukan oleh KH. As'ad Humam. Pada umumnya kegiatan belajar menggunakan metode Iqra tersebut, memerlukan guru pembimbing (ustaz ataupun ustazah) sebagai pemberi evaluasi murid terhadap ketepatan pengucapan huruf hijaiyah. Dikarenakan dalam proses pembelajaran tersebut dilakukan secara tatap muka di dalam ruangan. Sehingga muncul permasalahan yaitu keterbatasan waktu, tempat, dan pengalaman. Penelitian sebelumnya telah mencoba mengatasi masalah ini yaitu dengan membuat Aplikasi *Scoring* Iqra berbasis Android yang dapat membandingkan suara rekaman murid dengan suara ustaz. Akan tetapi hasil korelasi kemiripan *overall score* antara penilaian Aplikasi dan penilaian ustaz pada iqra1 sebesar 0.034 dalam kategori lemah, iqra2 sebesar 0.51 dalam kategori sedang, dan iqra3 sebesar -0.17 dalam kategori lemah. Untuk mengatasi permasalahan tersebut, pada penelitian ini akan menambah Algoritma *timbre scoring* dan mengubah nilai pembobotan skor dari masing-masing parameter *pitch, volume, rhythm, dan timbre* menjadi 0.55, 0.05, 0.38, dan 0.02. Sehingga hasil korelasi kemiripan *overall score* pada iqra1 sebesar 0.376 dalam kategori sedang, iqra2 sebesar 0.795 dalam kategori kuat, dan iqra3 sebesar 0.255 dalam kategori lemah.

Kata kunci: android, membandingkan suara, iqra, korelasi kemiripan

ABSTRACT

Wahyu Septiadi, *Advanced Development of Android Based Iqra Scoring Application with Pitch, Volume, Rhythm, and Timbre Parameters*

Supervisors: Komang Candra Brata, S.Kom., M.T., M.Sc. and Adam Hendra Brata, S.Kom., M.T., M.Sc.

Reading the Qur'an is an obligation for every Muslim. Before being able to read the Qur'an with correct pronunciation, it is necessary to practice reading letters or hijaiah sentences. One method of learning the Qur'an is using the Iqra method which was discovered by KH. As'ad Humam. In general, learning activities using the Iqra method require a supervising teacher (ustaz or ustazah) as a student to evaluate the accuracy of the pronunciation of the hijaiah letters. Because the learning process is done face-to-face in the room. So that problems arise, namely the limitations of time, place, and experience. Previous research has tried to overcome this problem, namely by making an Android-based Iqra Scoring Application that can compare the recorded voice of students with the voice of the cleric. However, the results of the correlation between the overall score similarity between the application assessment and the ustaz's assessment on iqra1 of 0.034 in the weak category, iqra2 of 0.51 in the medium category, and iqra3 of -0.17 in the weak category. To overcome these problems, this research will add a timbre scoring algorithm and change the score weighting values of each parameter of pitch, volume, rhythm, and timbre to 0.55, 0.05, 0.38, and 0.02. So that the correlation results of the overall score similarity on Iqra1 is 0.376 in the medium category, Iqra2 is 0.795 in the strong category, and Iqra3 is 0.255 in the weak category.

Keywords: android, compare voice, iqra, similarity correlation

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Manfaat	4
1.5 Batasan Masalah	4
1.6 Sistematika Pembahasan	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian Pustaka	6
2.2 Iqra	9
2.3 Android	9
2.4 Room	10
2.5 Pendekatan Agile	10
2.6 User Stories	11
2.7 Class Responsibility Collaborator (CRC) Cards	11
2.8 Feature Extraction	11
2.9 Mel-Frequency Cepstral Coefficient (MFCC)	11
2.10 Gaussian Mixture Model (GMM)	12
2.11 Rumus Slovin	13
2.12 Scoring suara	13



2.12.1	<i>Pitch Based Scoring</i>	14
2.12.2	<i>Volume Based Scoring</i>	14
2.12.3	<i>Rhythm Based Scoring</i>	15
2.12.4	<i>Timbre Based Scoring</i>	15
2.13	Pengujian Sistem	15
2.13.1	Pengujian Unit.....	15
2.13.2	Pengujian Validasi	16
2.13.3	Pengujian Korelasi Kemiripan	16
BAB 3	METODOLOGI	18
3.1	Studi Pustaka	18
3.2	Analisis Kebutuhan	19
3.3	Perancangan	19
3.4	Implementasi	20
3.5	Pengujian Sistem.....	20
3.6	Kesimpulan	20
BAB 4	ANALISIS KEBUTUHAN	21
4.1	<i>Storyboard</i> Aplikasi <i>Scoring</i> Iqra	21
4.2	Arsitektur Sistem	21
4.3	Pengumpulan Data	23
4.4	Kebutuhan Sistem Lama	23
4.4.1	<i>Use Case Diagram</i>	23
4.4.2	<i>User Interface</i>	24
4.5	Analisis Kebutuhan Sistem Baru	25
4.5.1	Identifikasi Aktor	25
4.5.2	Aturan Penomoran.....	25
4.5.3	Kebutuhan Fungsional	26
4.5.4	Kebutuhan Non Fungsional.....	27
4.5.5	<i>Use Case Diagram</i>	28
4.5.6	<i>Use Case Scenario</i>	28
4.6	<i>User Stories</i>	34
BAB 5	PERANCANGAN DAN IMPLEMENTASI	36
5.1	Perancangan	36

5.1.1 Perancangan <i>Sequence Diagram</i>	36
5.1.2 Perancangan <i>Class diagram</i>	38
5.1.3 Perancangan Algoritma	41
5.1.4 Perancangan Basis Data	48
5.1.5 Perancangan <i>CRC Cards</i>	49
5.1.6 Perancangan <i>User Interface (UI)</i>	49
5.2 Implementasi	61
5.2.1 Batasan Implementasi	61
5.2.2 Implementasi Algoritma	62
5.2.3 Implementasi Basis Data	66
5.2.4 Implementasi <i>User Interface (UI)</i>	67
BAB 6 PENGUJIAN DAN ANALISIS	72
6.1 Pengujian Unit	72
6.1.1 Skenario Pengujian <i>Overall Score</i>	72
6.1.2 Analisis Hasil Pengujian Unit	74
6.2 Pengujian Validasi	75
6.2.1 Skenario Pengujian Validasi	75
6.2.2 Analisis Hasil Pengujian Validasi	80
6.3 Pengujian Korelasi Kemiripan	80
BAB 7 PENUTUP	85
7.1 Kesimpulan	85
7.2 Saran	85
DAFTAR REFERENSI	86
LAMPIRAN A DOKUMENTASI FOTO	89
A.1 Ustaz sedang melakukan penilaian	89
A.2 Responden sedang menjawab soal bacaan huruf hijaiyah	90
A.3 Form penilaian <i>pitch, volume, rhythm, dan timbre</i>	91
A.4 Error rate	93



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka Penelitian Sebelumnya	6
Tabel 2.2 Kriteria Nilai r	17
Tabel 4.1 Identifikasi Aktor	25
Tabel 4.2 Keterangan aturan penomoran kebutuhan	26
Tabel 4.3 Kebutuhan fungsional pengguna	26
Tabel 4.4 Kebutuhan non fungsional pengguna	28
Tabel 4.5 <i>Use Case Scenario</i> menampilkan 3 menu jilid iqra	29
Tabel 4.6 <i>Use Case Scenario</i> membunyikan 10 suara bacaan huruf hijaiyah.....	29
Tabel 4.7 <i>Use Case Scenario</i> menampilkan 1 soal bacaan huruf hijaiyah.....	30
Tabel 4.8 <i>Use Case Scenario</i> merekam suara untuk menjawab soal.....	31
Tabel 4.9 <i>Use Case Scenario</i> menghitung skor kemiripan suara rekaman.....	32
Tabel 4.10 <i>Use Case Scenario</i> menampilkan cara penggunaan aplikasi.....	32
Tabel 4.11 <i>Use Case Scenario</i> menampilkan riwayat skor pengerjaan soal.....	33
Tabel 4.12 <i>Use Case Scenario</i> membunyikan riwayat suara rekaman.....	33
Tabel 4.13 Menghapus riwayat skor pengerjaan soal	34
Tabel 4.14 <i>User Stories</i> pada Aplikasi <i>Scoring Iqra</i>	35
Tabel 5.1 Pencarian bobot skor dengan minimum <i>error</i>	42
Tabel 5.2 <i>Pseudocode pitch based scoring</i>	44
Tabel 5.3 <i>Pseudocode volume based scoring</i>	45
Tabel 5.4 <i>Pseudocode rhythm based scoring</i>	46
Tabel 5.5 <i>Pseudocode timbre based scoring</i>	47
Tabel 5.6 <i>Pseudocode overall score</i>	47
Tabel 5.7 Skema Perancangan Basis Data.....	48
Tabel 5.8 Komponen pada UI menu bantuan iterasi pertama	50
Tabel 5.9 Komponen pada UI menu latihan iterasi pertama.....	52
Tabel 5.10 Komponen UI pada menu soal iterasi pertama	54
Tabel 5.11 Komponen UI pada menu bantuan iterasi kedua	56
Tabel 5.12 Komponen UI pada menu latihan iterasi kedua	57
Tabel 5.13 Komponen UI pada menu soal iterasi kedua	59
Tabel 5.14 Komponen UI halaman riwayat skor iterasi kedua.....	60

Tabel 5.15 Spesifikasi sistem perangkat lunak	61
Tabel 5.16 Spesifikasi sistem perangkat keras.....	62
Tabel 5.17 Potongan kode program <i>pitch scoring</i>	62
Tabel 5.18 Potongan kode program <i>volume scoring</i>	63
Tabel 5.19 Potongan kode program <i>rhythm scoring</i>	64
Tabel 5.20 Potongan kode program <i>timbre scoring</i>	65
Tabel 5.21 Potongan kode program <i>overall score</i>	66
Tabel 6.1 Skenario pengujian fungsi <i>getOverallScoreTest</i>	72
Tabel 6.2 Skenario pengujian fungsi <i>isExpectEqualActual</i>	73
Tabel 6.3 Implementasi kode pengujian <i>ScoreTest</i>	73
Tabel 6.4 Skenario pengujian validasi Aplikasi <i>Scoring Iqra</i>	75
Tabel 6.5 Perubahan korelasi kemiripan	81
Tabel 6.6 Perhitungan korelasi kemiripan iqra jilid 1	82
Tabel 6.7 Perhitungan korelasi kemiripan iqra jilid 2	83
Tabel 6.8 Perhitungan korelasi kemiripan iqra jilid 3	84



DAFTAR GAMBAR

Gambar 2.1 Posisi pelafalan makhraj huruf hijaiyah	9
Gambar 2.2 Alur <i>Extreme Programming</i>	10
Gambar 2.3 Skema untuk memperoleh <i>overall score</i>	14
Gambar 3.1 Alur diagram metodologi penelitian.....	18
Gambar 4.1 <i>Storyboard</i> Aplikasi <i>Scoring Iqra</i>	21
Gambar 4.2 Arsitektur Aplikasi <i>Scoring Iqra</i>	22
Gambar 4.3 <i>Use Case Diagram</i> Sistem Lama	24
Gambar 4.4 <i>User Interface</i> Sistem Lama	25
Gambar 4.5 Aturan Penomoran.....	26
Gambar 4.6 <i>Use Case Diagram</i> Sistem Baru	28
Gambar 5.1 <i>Sequence diagram</i> mencari <i>pitch scoring</i>	37
Gambar 5.2 <i>Sequence diagram</i> mencari <i>volume scoring</i>	37
Gambar 5.3 <i>Sequence diagram</i> mencari <i>rhythm scoring</i>	38
Gambar 5.4 <i>Sequence diagram</i> mencari <i>timbre scoring</i>	38
Gambar 5.5 Perancangan <i>class diagram Score</i>	39
Gambar 5.6 Perancangan <i>class diagram</i> Aplikasi <i>Scoring Iqra</i>	40
Gambar 5.7 Skema mendapatkan <i>pitch based scoring</i>	43
Gambar 5.8 Skema mendapatkan <i>volume based scoring</i>	44
Gambar 5.9 Skema mendapatkan <i>rhythm based scoring</i>	45
Gambar 5.10 Skema mendapatkan <i>timbre based scoring</i>	46
Gambar 5.11 Skema perubahan untuk memperoleh <i>overall score</i>	47
Gambar 5.12 Perancangan <i>CRC Cards class Score</i>	49
Gambar 5.13 Perancangan UI menu bantuan iterasi pertama.....	50
Gambar 5.14 Perancangan UI menu latihan iterasi pertama	52
Gambar 5.15 Perancangan UI menu soal iterasi pertama.....	53
Gambar 5.16 Perancangan UI menu bantuan iterasi kedua.....	56
Gambar 5.17 Perancangan UI menu latihan iterasi kedua.....	57
Gambar 5.18 Perancangan UI menu soal iterasi kedua.....	58
Gambar 5.19 Perancangan UI halaman riwayat skor iterasi kedua.....	60
Gambar 5.20 Implementasi Basis data	67

Gambar 5.21 Tampilan implementasi UI menu bantuan 68

Gambar 5.22 Tampilan implementasi menu latihan 69

Gambar 5.23 Tampilan implementasi UI menu soal 70

Gambar 5.24 Tampilan implementasi UI halaman riwayat skor 71

Gambar 6.1 Hasil pengujian unit *overall score* 75



DAFTAR LAMPIRAN

LAMPIRAN A DOKUMENTASI FOTO	89
A.1 Ustaz sedang melakukan penilaian	89
A.2 Responden sedang menjawab soal bacaan huruf hijaiyah	90
A.3 Form penilaian <i>pitch, volume, rhythm, dan timbre</i>	91
A.4 Error rate	93



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Salah satu kewajiban bagi setiap umat Islam adalah membaca, mempelajari, memahami, serta mengamalkan Al-Qur'an. Sebaiknya dalam mempelajari Kitab suci Al-Qur'an diberikan semenjak dini, seperti yang disebutkan pada HR. Thabrani, Anas yang berbunyi "Barangsiapa mengajari anaknya untuk membaca Al-Qur'an maka akan diampuni dosanya, dan barangsiapa yang mengajarkan dengan hafalan di luar kepala, maka Allah SWT akan membangkitkannya kelak di hari kiamat dengan wajah seperti bulan purnama". Hal tersebut ditegaskan secara khusus oleh Rasulullah SAW yang berharap dapat menumbuhkan kepercayaan terhadap Allah SWT sebagai tuhan-Nya dan Al-Qur'an sebagai firman-Nya (Sasongko, 2019). Sebelum bisa membaca Al-Qur'an perlu adanya latihan mempelajari huruf hijaiyah terlebih dahulu (Sulistya, 2016). Jika ingin bisa membaca Al-Qur'an, maka perlu penguasaan tata cara membaca Al-Qur'an dengan lafal dan ucapan yang benar (tajwid) dari latihan membaca huruf, harakat, kalimat, serta ayat-ayat. Maka dari itu belajar menguasai dasar-dasar Al-Qur'an tersebut dapat diberikan sejak usia dini, sehingga pada saat dewasa nanti dapat benar-benar menguasai dalam membaca Al-Qur'an sesuai kaidah-kaidah yang telah ditentukan (Munir & Sudarsono, 1994).

Terdapat metode belajar Al-Qur'an yang populer di Indonesia salah satunya yaitu metode Iqra, metode ini ditemukan oleh KH. As'ad Humam. Dengan adanya metode iqra ini, banyak kalangan yang menilai sangat efektif sebagai bahan ajar. Dikarenakan pada buku Iqra ini sudah terdapat ejaan huruf hijaiyah, potongan kata pendek, hingga juz amma (Fajry, 2019). Pada umumnya dalam kegiatan belajar menggunakan metode Iqra ini murid harus berlatih dengan ustaz. Di sini ustaz bertugas sebagai pembimbing dan memberikan evaluasi terhadap cara pengucapan huruf hijaiyah (makhraj) dari seorang murid secara tatap muka di dalam ruangan (Hartono, 2017). Akan tetapi terdapat beberapa kelemahan dengan dilakukan secara tatap muka di dalam ruangan dengan metode Iqra tersebut yaitu hasil pengajaran membaca Al-Qur'an metode Iqra kualitasnya berkurang. Hal tersebut disebabkan karena ustaz ataupun ustazah yang seadanya, dan semua orang boleh mengajar metode Iqra walaupun belum pernah mengikuti pembinaan metode Iqra, sehingga aturan dalam metode Iqra kurang ketat (Putri, 2017). Selain itu terdapat kelemahan lain yaitu terlalu banyaknya jumlah siswa daripada jumlah guru pembimbing menyebabkan tidak kondusifnya suasana di dalam ruangan belajar, sehingga murid tidak bisa konsentrasi dalam menyerap materi iqra yang diajarkan (Hamidah, 2011).

Beberapa permasalahan yang dijelaskan pada paragraf sebelumnya telah ada beberapa solusi pada penelitian-penelitian sebelumnya, akan tetapi masih belum cukup untuk menyelesaikan masalah tersebut. Pada penelitian (Nur'aini, 2014) menggunakan radio MQ FM Bandung yang mengadakan program magrib mengaji, di mana para pendengar radio dapat menghubungi ustaz yang sedang siaran dari

jam 6 sampai jam 7 malam di program radio tersebut dengan tujuan ustaz dapat langsung mengevaluasi bacaan dari pendengar radio. Akan tetapi terdapat kendala menurut penelitian tersebut yaitu banyaknya penelpon dari pendengar radio yang ingin meminta untuk di evaluasi bacaan Al-Qur'an, menyebabkan kehabisan waktu, di mana waktu program radio tersebut hanya 1 jam saja. Penelitian lainnya yaitu (Fauzan, et al., 2018) mengubah suara pengguna menjadi teks menggunakan bantuan Google *speech recognition* yang kemudian dilakukan penilaian melalui perbandingan teks, hal tersebut memunculkan kendala yaitu karena tidak ada penilaian dengan melakukan perbandingan suara untuk menilai makhraj dari murid, maka penilaian tersebut kurang relevan. Seperti yang disebutkan dalam penelitian (Nasser, et al., 2013) bahwa berkaitan dengan belajar iqra penting dilakukan untuk membandingkan suara dengan suara untuk kejelasan makhraj agar terhindar dari kesalahan cara pengucapan yang memungkinkan bisa berbeda arti. Selain solusi tadi kedua penelitian tersebut, terdapat penelitian lain dari (Nugraha, et al., 2019) yaitu membuat aplikasi yang dapat membandingkan suara murid dengan suara ustaz yang telah direkam dalam ekstensi .wav sehingga murid dalam belajar metode iqra tersebut dapat memperoleh penilaian berupa skor. Penilaian tersebut didapatkan menggunakan konsep karaoke dengan parameter nilai *pitch*, *volume*, dan *rhythm* sehingga diperoleh *overall score* yang dapat digunakan sebagai evaluasi belajar oleh murid. Akan tetapi setelah dilakukan pengujian korelasi kemiripan *overall score* antara penilaian ustaz dengan penilaian dari aplikasi pada iqra 1, iqra 2, dan iqra 3 menghasilkan masing-masing 0.034, 0.51, dan -0.17. Dari hal tersebut dapat diketahui bahwa nilai korelasi kemiripan tersebut berada pada kategori lemah, sedang, dan lemah. Sehingga algoritma dari konsep karaoke dalam menentukan skor masih belum cocok dengan metode iqra.

Solusi yang akan diusulkan dalam penelitian ini yaitu melakukan pengembangan lanjut dari penelitian sebelumnya (Nugraha, et al., 2019) dengan menambah parameter baru yang nantinya digunakan untuk menentukan *overall score* yaitu *timbre*. Menurut (Tsai, et al., 2012) sistem penilaian karaoke dalam bernyanyi harus memanfaatkan berbagai isyarat akustik untuk membuat perbandingan antara lagu uji dan dasar referensi. Dari sudut pandang elemen musik, terdapat 4 dasar isyarat akustik (parameter) yaitu *pitch*, *volume*, *rhythm*, dan *timbre*. Namun pada penelitian (Tsai, et al., 2012) tidak menjelaskan mengenai perhitungan skor *timbre* dikarenakan pada *timbre* melibatkan karakteristik suara alami pada individu, akan tetapi tidak menutup kemungkinan lagu yang dibawakan dengan cara dan nada yang sama tetapi warna suara (*timbre*) yang berbeda dapat menghasilkan skor dari parameter *timbre*, maka perlu melakukan analisis berbasis *timbre*. Pada penelitian ini akan mengimplementasi algoritma untuk menentukan *timbre scoring* menggunakan Algoritma GMM yang pernah digunakan untuk mencari kesamaan dari sebuah lagu berdasarkan warna suara pada beberapa lagu, sehingga dari beberapa lagu tersebut di urutan tingkat kesamaan warna suaranya dengan lagu yang dicari (Fujihara & Goto, 2007). Dari hasil penelitiannya dengan membandingkan warna suara antara kedua lagu

dengan menggunakan beberapa metode salah satunya dengan Algoritma GMM tersebut, dapat dilakukan secara efektif dan menghasilkan tingkat kemiripan yang sesuai. Maka dari itu tingkat kesamaan warna suara tersebut, dapat dijadikan sebagai skor dari parameter *timbre* dan digabungkan dengan ketiga parameter yang lain yaitu *pitch*, *volume*, dan *rhythm* untuk mencari *overall score* pada penelitian ini. Selain itu, perubahan yang dilakukan dari penelitian sebelumnya yaitu pengguna dapat mengevaluasi hasil dari beberapa latihan dalam membaca huruf hijaiyah. Dikarenakan *overall score* tersebut akan disimpan ke halaman riwayat skor yang dapat dilihat oleh pengguna Aplikasi.

Pada paragraf-paragraf sebelumnya telah dijelaskan beberapa masalah dan solusinya, maka dari topik tersebut dapat ditentukan judulnya. Pada penelitian ini berjudul “**Pengembangan Lanjut Aplikasi Scoring Iqra Berbasis Android dengan Parameter Pitch, Volume, Rhythm, dan Timbre**”. Penelitian ini diharapkan dapat mempermudah bagi semua pengguna Aplikasi *Scoring Iqra* dalam belajar membaca Al-Qur’an (mengaji) dengan efektif dan efisien tanpa harus bertatap muka dengan pembimbing mengaji (ustaz). Selain itu dengan adanya pengembangan lanjut Aplikasi *Scoring Iqra* ini, dapat meningkatkan korelasi kemiripan *overall score* antara penilaian yang dilakukan oleh aplikasi dengan penilaian manual oleh ustaz. Sehingga semua pengguna Aplikasi *Scoring Iqra* dapat mengetahui seberapa tingkat pemahaman dalam membaca huruf hijaiyah pada metode Iqra melalui fitur *Scoring* dengan akurat.

1.2 Rumusan Masalah

Adapun permasalahan yang dijelaskan pada latar belakang diatas, dapat diambil beberapa rumusan masalah sebagai berikut:

1. Bagaimana hasil dari perubahan kebutuhan sistem dari penelitian sebelumnya dengan penelitian ini?
2. Bagaimana hasil dari setelah implementasi algoritma GMM dalam menentukan skor *timbre* pada Aplikasi *Scoring Iqra* berbasis Android?
3. Bagaimana perubahan hasil korelasi kemiripan *overall score* antara penilaian manual yang dilakukan ustaz dengan penilaian yang dilakukan oleh aplikasi dalam menilai seseorang membaca huruf hijaiyah pada metode Iqra setelah ditambah *timbre* sebagai parameter baru?

1.3 Tujuan

Tujuan yang ingin dicapai pada penelitian ini adalah:

1. Mengetahui hasil perubahan kebutuhan sistem dari penelitian sebelumnya dengan kebutuhan sistem pada penelitian ini.
2. Mengetahui hasil dari setelah implementasi algoritma GMM dalam menentukan skor *timbre* pada Aplikasi *Scoring Iqra* berbasis Android.

3. Mengetahui perubahan hasil korelasi kemiripan *overall score* antara penilaian manual yang dilakukan ustaz dengan penilaian yang dilakukan oleh aplikasi dalam menilai seseorang membaca huruf hijaiyah pada metode Iqra setelah ditambah *timbre* sebagai parameter baru.

1.4 Manfaat

Adapun manfaat dari penelitian ini yaitu diharapkan dapat bermanfaat untuk orang yang ingin belajar membaca Al-Qur'an dengan metode Iqra, di mana pembelajaran tersebut tidak ada batasan waktu dan tempat. Dikarenakan hanya menggunakan Aplikasi dalam menilai pengucapan huruf hijaiyah. Sehingga Aplikasi ini dapat digunakan secara fleksibel khususnya anak-anak berusia pelajar.

1.5 Batasan Masalah

Adapun batasan masalah dalam perancangan Aplikasi *Scoring Iqra* ini adalah:

1. Aplikasi hanya dibuat pada platform Android dengan minimal SDK versi 21 (*Lollipop*).
2. Aplikasi ini hanya menggunakan bahasa Indonesia.
3. Aplikasi tidak tersedia dalam bentuk tablet, *smart TV*, dan *wearable device*.
4. Aplikasi hanya merepresentasikan metode iqra pada jilid 1 - 3.
5. Suara referensi dari ustaz direkam pada ruangan yang tidak bising dengan menggunakan *smartphone* dan tidak ada proses perubahan pada hasil rekaman tersebut.
6. Pengembangan Aplikasi menggunakan pendekatan *Agile* dengan metode *Extreme Programming (XP)*.

1.6 Sistematika Pembahasan

Untuk dapat mencapai tujuan sesuai yang diharapkan maka sistematika pembahasan yang disusun pada penelitian ini antara lain:

BAB I : PENDAHULUAN

Pada bab ini akan dijelaskan mengenai latar belakang masalah, rumusan masalah, tujuan, manfaat penelitian, batasan masalah, dan sistematika pembahasan dari penelitian "Pengembangan Lanjut Aplikasi *Scoring Iqra* Berbasis Android dengan Parameter *Pitch, Volume, Rhythm, dan Timbre*".

BAB II : LANDASAN KEPUSTAKAAN

Pada bab ini menjelaskan mengenai persamaan dan perbedaan dari penelitian sebelumnya yang sejenis dan teori-teori pendukung penelitian ini. Selain itu juga dijelaskan mengenai rumus *timbre scoring* menggunakan algoritma GMM, di mana algoritma tersebut akan digunakan untuk menentukan skor dari parameter *timbre* yang diperoleh pengguna setelah merekam suara bacaan huruf hijaiyah.

BAB III : METODOLOGI

Pada bab ini berisi mengenai langkah-langkah yang digunakan dalam pengembangan Aplikasi, dan juga menjelaskan alur kerja dalam penulisan penelitian dalam menyelesaikan masalah selama proses penelitian.

BAB IV : ANALISIS KEBUTUHAN

Pada bab ini menuliskan analisis kebutuhan yang mengacu pada penelitian sebelumnya (Nugraha, et al., 2019). Terdapat sedikit perubahan pada bagian ini adanya penambahan *storyboard*. Selain itu pada bagian ini menjelaskan arsitektur sistem, pengumpulan data, kebutuhan pada sistem lama, analisis kebutuhan sistem baru berupa identifikasi aktor, aturan penomoran, kebutuhan fungsional dan non fungsional, *use case diagram* dan *use case scenario*, serta menjelaskan *user stories* dari pengguna aplikasi.

BAB V : PERANCANGAN DAN IMPLEMENTASI

Pada bab ini berisi tentang proses perancangan Aplikasi *Scoring Iqra* berbasis Android yang meliputi: perancangan *sequence diagram*, *class diagram*, algoritma menentukan skor dari 4 parameter, basis data, *CRC Cards*, dan juga *User Interface* (UI). Sedangkan pada proses implementasi meliputi: batasan implementasi, implementasi algoritma, basis data, dan juga implementasi *User Interface* (UI).

BAB VI : PENGUJIAN DAN ANALISIS

Pada bab ini menjelaskan tentang pengujian dan analisis dari pengembangan Aplikasi *Scoring Iqra* yang telah dibuat. Pengujian terdiri dari pengujian unit, validasi dan korelasi kemiripan. Selain dilakukan pengujian, akan dilakukan analisis dari tiap pengujian.

BAB VII : PENUTUP

Pada bab ini menjelaskan kesimpulan yang didapat selama proses pengembangan Aplikasi *Scoring Iqra*. Kesimpulan ini dilandasi sesuai batasan masalah yang dibuat pada bab 1. Selain itu pada bab ini menjelaskan saran-saran, apabila terdapat pengembangan lebih lanjut terhadap Aplikasi *Scoring Iqra*.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Pada subbab ini menjelaskan landasan literatur terhadap penelitian-penelitian sebelumnya yang sejenis. Ada beberapa jurnal yang membahas mengenai cara mendapatkan tingkat kemiripan antara 2 suara. Dimana dari cara atau metode tersebut akan dijadikan acuan / referensi dalam penelitian ini.

Tabel 2.1 Kajian Pustaka Penelitian Sebelumnya

No.	Penulis (tahun)	Persamaan	Perbedaan
1.	Mohamad Handy Nugraha (2019)	1. Mengembangkan Aplikasi <i>Scoring Iqra</i> antara suara uji dengan suara referensi rekaman bacaan huruf hijaiyah. Kemudian dilakukan pencarian korelasi kemiripan antara penilaian otomatis sistem dengan penilaian manual oleh manusia.	1. Pada penelitian sebelumnya dalam menentukan <i>overall score</i> hanya menggunakan 3 parameter suara yaitu <i>pitch, volume, rhythm</i> . Sedangkan pada penelitian ini menggunakan 4 parameter suara yaitu <i>pitch, volume, rhythm, dan timbre</i> . 2. Pada penelitian ini terdapat beberapa pengembangan lanjut terhadap kebutuhan sistem aplikasinya.
2.	Wei-Ho Tsai, Member, IEEE, dan Hsin-Chieh Lee (2012)	1. Mengembangkan sistem <i>scoring</i> suara antara suara uji dengan suara referensi. Kemudian dilakukan pencarian korelasi kemiripan antara penilaian otomatis sistem dengan penilaian manual oleh manusia.	1. Pada penelitian sebelumnya rumus <i>scoring</i> suara digunakan untuk menilai ketepatan penyanyi dalam bernyanyi pada sistem karaoke. Sedangkan pada penelitian ini digunakan untuk menilai ketepatan pengucapan bacaan huruf hijaiyah (tajwid).

			<p>2. Pada penelitian sebelumnya dalam menentukan <i>overall score</i> hanya menggunakan 3 parameter suara yaitu <i>pitch, volume, rhythm</i>. Sedangkan pada penelitian ini menggunakan 4 parameter suara yaitu <i>pitch, volume, rhythm, dan timbre</i>.</p>
<p>3. Hiromasa Fujihara dan Masataka Goto (2007)</p>	<p>1. Menggunakan Algoritma <i>Gaussian Mixture Model (GMM)</i> sebagai penghitung tingkat kesamaan antara kedua suara berdasarkan warna suara / <i>timbre</i>.</p>	<p>1. Pada penelitian sebelumnya dalam mencari <i>Feature extraction</i> suara menggunakan metode <i>LPC-derived mel cepstral coefficients (LPMCC)</i> dan ΔF_0s. Sedangkan pada penelitian ini menggunakan metode <i>Mel-Frequency Cepstral Coefficient (MFCC)</i>.</p> <p>2. Pada penelitian sebelumnya metode <i>clustering</i> pada proses <i>GMM</i> menggunakan <i>Expectation Maximization (EM)</i>. Sedangkan pada penelitian ini menggunakan metode <i>K-Means</i>.</p>	

Adapun referensi jurnal pertama yaitu penelitian dari (Nugraha, et al., 2019). Pada jurnal tersebut menjelaskan pembuatan Aplikasi *Scoring Iqra* yang dapat menampilkan skor ketika pengguna setelah merekam bunyi bacaan huruf hijaiyah yang dipilih. Di mana pada Aplikasi tersebut menerapkan metode evaluasi penyanyi pada lagu karaoke dengan mengimplementasikan algoritma yang

digunakan untuk mendapatkan masing-masing nilai skor dari parameter *pitch*, *volume*, *rhythm* kemudian mengolah ketiga nilai tersebut menjadi nilai *overall score*. Algoritma yang digunakan pada jurnal pertama ini yaitu berasal dari penelitian (Tsai, et al., 2012), yang akan dijelaskan pada referensi jurnal kedua.

Referensi jurnal kedua yaitu penelitian yang dilakukan oleh (Tsai, et al., 2012). Pada jurnal tersebut dijelaskan mengenai langkah-langkah dalam mendapatkan nilai secara otomatis pada lagu karaoke yang dilakukan oleh penyanyi. Adapun parameter penilaiannya berdasarkan nilai *pitch*, *volume*, dan *rhythm* terhadap suara manusia yang sedang menyanyikan lagu mandarin. Dari ketiga parameter nilai tersebut akan di proses menggunakan konsep pembobotan pada setiap parameternya sehingga ketika dijumlahkan akan menghasilkan *overall score*. Ketika didapatkan 2 *overall score* antara penyanyi karaoke yang di nilai oleh sistem dengan penyanyi karaoke yang di nilai secara manual oleh manusia, kemudian dicari tingkat korelasi kemiripan antara kedua *overall score* tersebut menggunakan metode *pearson product-moment correlation coefficient*.

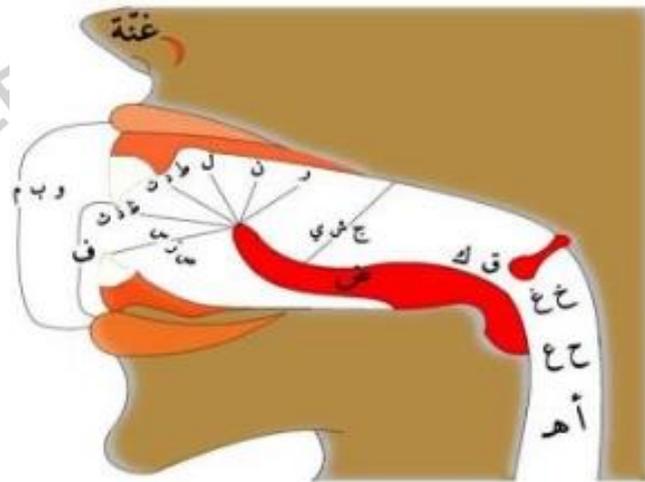
Referensi jurnal ketiga yaitu pada penelitian (Fujihara & Goto, 2007), dimana pada jurnal tersebut menjelaskan *Music Information Retrieval* (MIR) berbasis *timbre* vokal yang bertujuan untuk mencari kesamaan dari sebuah lagu berdasarkan warna suara pada beberapa lagu, sehingga dari beberapa lagu tersebut di urutkan tingkat kesamaan warna suaranya dengan lagu yang dicari tadi. Adapun beberapa metode yang digunakan adalah metode untuk mengekstraksi vektor fitur dari sinyal audio polifonik dengan metode *LPC-derived mel cepstral coefficients* (LPMCC) dan $\Delta F0s$. Sedangkan metode pengukuran tingkat kesamaan vokal *timbre* antara 2 lagu menggunakan metode *Gaussian Mixture Model* (GMM). Pada saat pengujian sistem tersebut menghasilkan 5 dari 6 subjek suara yang menunjukkan keefektifan dalam membandingkan tingkat kemiripan antara kedua lagu. Akan tetapi meskipun efektif, komputasi yang diperlukan sangat tinggi dan memerlukan penyimpanan yang besar. Hal tersebut dikarenakan menggunakan semua vektor fitur untuk menghitung *timbre*.

Dari beberapa referensi diatas, penelitian ini akan mencoba melakukan kombinasi metode yaitu dalam menentukan vektor fitur ekstraksi parameter *timbre* dari suara rekaman ustad dan pengguna, masih tetap menggunakan Algoritma *Mel-Frequency Cepstral Coefficient* (MFCC) seperti menentukan fitur ekstraksi parameter *rhythm* yang ada pada referensi jurnal pertama. Kemudian untuk mendapatkan tingkat kesamaan warna suaranya, diambil dari data vektor fitur ekstraksi kedua suara rekaman tersebut dengan Algoritma GMM sebagai *timbre score*. Sedangkan dalam menentukan *overall score* sesuai konsep pada referensi jurnal kedua, di mana akan terdapat penambahan parameter baru yaitu *timbre* dan juga akan dilakukan perubahan pembobotan nilai dalam setiap parameter skornya. Kemudian untuk mencari total responden pada penelitian ini menggunakan Rumus Slovin, yang nantinya semua responden tersebut akan mencoba Aplikasi dalam menilai ketepatan membaca huruf hijaiyah. Dari semua penilaian yang diperoleh dari sistem dan penilaian ustad, akan dicari korelasi

kemiripannya dengan Rumus *Pearson* seperti pada jurnal pertama dan kedua. Untuk alur lebih detail akan dijelaskan pada bab selanjutnya.

2.2 Iqra

Terdapat metode belajar Al-Qur'an yang populer di Indonesia, salah satunya yaitu metode Iqra yang ditemukan oleh KH. As'ad Humam. Dengan adanya metode iqra ini, banyak kalangan yang menilai sangat efektif sebagai bahan ajar. Dikarenakan pada buku Iqra ini sudah terdapat ejaan huruf hijaiyah, potongan kata pendek, hingga juz amma (Fajry, 2019). Untuk menghasilkan kurikulum pendidikan Al-Qur'an yang baik di dalam menerapkan metode Iqra tersebut, perlu mengutamakan kemampuan pribadi dari masing-masing murid. Dikarenakan kemampuan untuk menerima pelajaran tajwid dari murid satu dengan murid yang lain bisa berbeda, meskipun waktu pembelajaran yang diberikan sama (Srijatun, 2017). Posisi pelafalan makhraj huruf hijaiyah seperti pada gambar 2.1.



Gambar 2.1 Posisi pelafalan makhraj huruf hijaiyah
(Sulistya, 2016)

2.3 Android

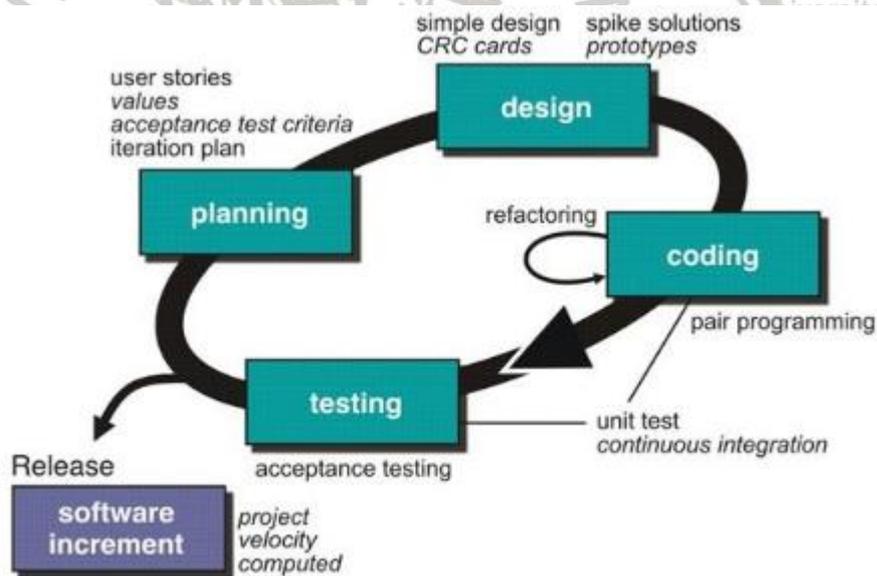
Pada penelitian ini, aplikasi Iqra di kembangkan pada platform *mobile* bersistem operasi Android, dikarenakan pangsa pasar di Indonesia sistem operasi tersebut sebesar 92.14% (Statcounter, 2020). Selain itu Android termasuk sistem operasi yang sumbernya tergolong *open source* dan Google telah merilis kodenya dibawah Lisensi Apache. Kode dengan sumber terbuka dan lisensi perizinan pada Android ini akan memudahkan para pengembang Aplikasi untuk memodifikasi, dan distribusi secara bebas. Dan saat ini perkembangan Android sudah sampai pada versi 11 yang baru diluncurkan pada 8 September 2020 (Hermawan, 2020).

2.4 Room

Room adalah suatu basis data yang menyediakan antarmuka *database* yang dilandasi oleh SQLite. SQLite termasuk juga basis data yang tergolong *open source* dan tertanam dalam sistem operasi Android. Pada *Room* ini menyediakan *repository* ke dalam *Data Access Object* (DAO). Syarat yang harus diterapkan pada *library Room* ini yaitu mengharuskan Aplikasi memiliki satu *instance* ke *database room* yang dapat digunakan untuk mengakses beberapa tabel (*entitas*). Di dalam DAO tersebut berisi pernyataan SQL yang dapat mengakses *repository* seperti memasukkan, mengambil, mengubah dan menghapus data di dalam *database* SQLite (Putra, et al., 2020).

2.5 Pendekatan Agile

Pendekatan ini digunakan untuk membangun aplikasi *scoring* Iqra. Metode pengembangan *Agile* merupakan sekelompok metodologi pengembangan perangkat lunak dengan jangka pendek dengan mengadaptasi secara cepat terhadap perubahan dalam bentuk apapun (Alexandra, 2017). Adapun model yang digunakan yaitu dengan model *Extreme Programming* (XP) yang dapat dilihat pada gambar 2.2.



Gambar 2.2 Alur Extreme Programming

(Sumber Pressman)

Adaptasi yang dilakukan pada fase *planning* yaitu seperti analisis kebutuhan dari aplikasi Iqra yang disesuaikan pada penelitian sebelumnya (Nugraha, et al., 2019). Hanya saja pada penelitian ini menambahkan metode baru dalam melakukan perhitungan skor *timbre* ketika pembacaan huruf hijaiyah. Pada fase *design* ini pasti ada perubahan kebutuhan, maka dari itu yang akan dilakukan pada penelitian ini yaitu membuat perancangan *sequence diagram*, *class diagram*, perubahan *algoritma* dalam menentukan skor dari 4 parameter beserta nilai

pembobotan skor dalam menentukan *overall score*, selain itu membuat perancangan basis data untuk menyimpan data skor, kemudian dilakukan perubahan *Class Responsibilities Collaboration (CRC) Cards*, dan juga perubahan perancangan *User Interface (UI)*.

Pada fase *coding* dijelaskan pada buku (Pressman, 2009), dalam pembuatan kode program perlu adanya pengujian unit terlebih dahulu. Sedangkan pada penelitian ini dilakukan pembuatan kode program langsung berdasarkan perancangan *User Interface (UI)*, *class diagram*, *sequence diagram* yang telah dibuat sebelumnya dan tanpa ada perbaikan *programming*. Adaptasi fase *testing* yang diterapkan pada penelitian ini yaitu melakukan *unit testing* (pengujian unit), pengujian validasi, dan juga pengujian korelasi kemiripan antara kedua penilaian *overall score*.

2.6 User Stories

User stories adalah catatan permintaan dari pengguna dalam pengembangan aplikasi. *User stories* mencakup informasi kebutuhan pengguna secara detail yang digunakan sebagai informasi utama dalam perencanaan dan penjadwalan dalam pengembangan serta sebagai acuan analisis kebutuhan dan pengujian (Ambler, 2002).

2.7 Class Responsibility Collaborator (CRC) Cards

Class Responsibility Collaborator (CRC) Cards merupakan kumpulan kartu indeks yang dibagi menjadi tiga yaitu nama *class*, *responsibilities*, dan juga *collaborator* yang berfungsi mendeklarasikan objek-objek kelas dan juga kelas yang saling berhubungan (Wells, 1999).

2.8 Feature Extraction

Feature Extraction atau ekstraksi fitur adalah salah satu cara yang dapat diimplementasikan untuk dapat mengenali suatu *object* berdasarkan *histogram*. Fitur ekstraksi ini bertujuan untuk menghitung dengan cara membandingkan antar kedua *object* yang dipakai sebagai alat klarifikasi suatu citra berdasarkan ciri-ciri *histogram* yang dimiliki (Religia, 2019). Pada penelitian ini *Feature Extraction* digunakan untuk mencari ciri pada kedua berkas audio antara tajwid pengguna dengan tajwid ustaz. Dimana metode yang digunakan adalah metode *Mel-Frequency Cepstral Coefficient (MFCC)*.

2.9 Mel-Frequency Cepstral Coefficient (MFCC)

MFCC adalah suatu algoritma ekstraksi sinyal suara yang peka terhadap bising (*noise*). MFCC ini dapat menghasilkan akurasi yang tinggi ketika di lingkungan yang bersih atau dapat diartikan sunyi. Sebaliknya ketika di lingkungan yang bising, akurasi akan terjadi penurunan drastis (Prayogi, 2019). Pada penelitian ini MFCC akan digunakan untuk fitur ekstraksi sebagai pembantu proses mencari skor dari

parameter *timbre*, yang pada penelitian sebelumnya sudah menerapkan algoritma ini terhadap penentuan fitur ekstraksi dari parameter *rhythm*.

2.10 Gaussian Mixture Model (GMM)

Algoritma ini termasuk ke dalam tipe *density* model yang terdiri dari komponen fungsi-fungsi *Gaussian*. Untuk menghasilkan *multi model density*, maka dari komponen fungsi tadi perlu adanya *weight* yang berbeda-beda. GMM sering digunakan untuk mencari model dari warna-warna suatu *background* pada setiap *pixel*. *Pixel* adalah sebuah titik terkecil pada suatu gambar atau citra yang dihitung per inci. Dari setiap *pixel* memiliki GMM berbeda-beda, dan dari inputan akan diperoleh data yang dapat diolah berupa warna suatu *pixel*. Untuk mendapatkan model-model dari GMM perlu mengolah warna *pixel* tadi berdasarkan waktu. Biasanya model yang terbentuk akan dibagi menjadi 2 bagian, yang disebut *background model* dan *non-background model*. Semakin banyak jumlah model GMM yang digunakan maka akan mempengaruhi jumlah *background model* dalam suatu *pixel*. Sedangkan metode *clustering* yang sering digunakan pada proses GMM yaitu menggunakan *Expectation Maximization* (EM) dan K-Means. Di mana metode K-Means memiliki kecepatan proses lebih cepat dengan hasil yang cukup baik (Rostianingsih, et al., 2008).

Proses pertama pada metode GMM ini yaitu melakukan pencocokan input terhadap distribusi sampai ditemukan distribusi yang paling cocok. Hal tersebut ditandai dengan nilai suatu *pixel* berada dalam jarak 2,5 standar deviasi dari sebuah distribusi. Di dalam proses pencocokan ini terdapat satu proses lagi yaitu dilakukan proses *update* nilai dari parameter GMM yang nantinya digunakan untuk mengolah input selanjutnya. Nilai tersebut antara lain *weight*, *mean*, dan *varian*. Selanjutnya proses kedua dilakukan pemilihan distribusi yang mencerminkan *background*. Pada proses kedua ini, dilakukan pengurutan dari beberapa model. Sehingga distribusi yang paling mencerminkan *background* akan diletakkan pada posisi atas, sedangkan distribusi yang tidak mencerminkan *background* akan diletakkan di posisi bawah (Rostianingsih, et al., 2008).

Metode GMM ini menjadi peran penting pada penelitian ini yaitu digunakan untuk menentukan tingkat kesamaan antara *timbre* atau warna nada dari kedua audio rekaman bacaan huruf hijaiyah. Dikarenakan metode GMM yang ada pada jurnal referensi (Fujihara & Goto, 2007) tidak menggunakan K-Means sebagai metode *clustering*, maka dari itu algoritma GMM yang digunakan pada penelitian ini menggunakan *library Java Multi Threaded GMM Single Speaker Recognition* (jmtssr) yang telah menerapkan K-Means sebagai metode *clustering*. Di mana *library* ini didapatkan dari *website open source* Github.

2.11 Rumus Slovin

Sebuah rumus yang digunakan untuk menghitung jumlah *minimum sample* dari suatu peristiwa. Dalam penelitian survei ukuran sampel biasanya sangat besar, sehingga diperlukan rumus untuk menghasilkan sampel yang dapat dihitung secara keseluruhan. Rumus slovin dapat dilihat pada persamaan 2.1.

$$n = \frac{N}{1+N(e)^2} \quad (2.1)$$

Dari notasi diatas, n adalah jumlah *minimum sample*, nilai N adalah jumlah populasi, sedangkan nilai e adalah besar persen *margin of error* karena kesalahan pengambilan sampel yang masih dapat ditolerir, misalnya 5 %. Namun, perihal *margin of error* ini dapat ditetapkan sesuai kebutuhan peneliti (Hidayat, 2017).

2.12 Scoring suara

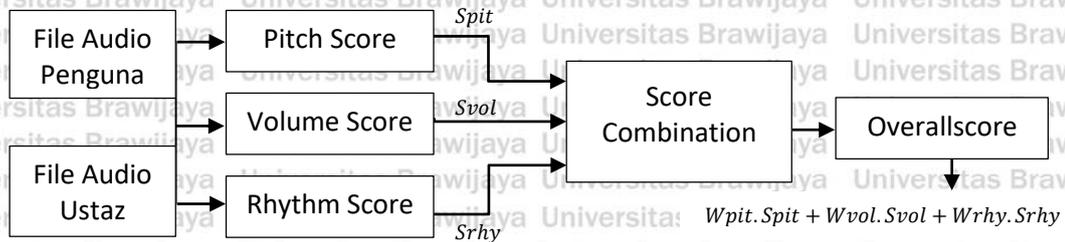
Pada penelitian ini dalam mencari masing-masing skor paramater masih mengacu pada penelitian sebelumnya (Nugraha, et al., 2019). Hanya saja menambahkan satu parameter baru yaitu *Timbre Scoring*. Menurut penelitian (Tsai, et al., 2012) untuk mendapatkan skor harus dilakukan perbandingan antara 2 *file audio* antara suara uji dengan suara referensi, dengan menerapkan perhitungan skor dari 3 dasar akustik suara yaitu *pitch*, *volume*, *rhythm*. Akan tetapi menurut penelitiannya terdapat satu dasar akustik suara lagi yang dapat dimasukkan untuk perhitungan *overall score*, namun sedikit sulit karena melibatkan karakter suara dari individu dan perlu adanya analisis lebih lanjut yaitu *timbre*. Pada penelitian ini, untuk pembahasan mengenai perhitungan perubahan pembobotan skor dari setiap parameter suara dan rumus *overall score* setelah ditambah parameter *timbre* akan dibahas secara detail pada bab perancangan.

Adapun definisi dari parameter *Pitch* adalah tinggi / rendah nada suara. *Volume* adalah tingkat keras atau lembut suara (Budi, 2012). *Rhythm* menjadi unsur serapan kata ritme, di mana ritme dalam bahasa indonesia lazim dihubungkan dengan irama, sedangkan dalam musik ritme berarti waktu atau tempo yang dihasilkan oleh suara (Takari, 2016). Sedangkan *Timbre* sebagai parameter baru yang digunakan pada penelitian ini adalah suatu ciri khas atau warna khas dari suara (Budi, 2012). Menurut penelitian (Tsai, et al., 2012) dari 3 skor akustik suara tersebut akan dikalikan dengan bobot dari masing-masing parameter dan dijumlahkan sehingga akan menghasilkan *overall score* atau total skor. *Overall Score* dapat diperoleh dengan persamaan 2.2.

$$\text{Overall score} = W_{pit}.Spit + W_{vol}.Svol + W_{rhy}.Srhy \quad (2.2)$$

$Spit$, $Svol$, $Srhy$ adalah masing-masing skor dari *pitch*, *volume*, *rhythm*. Sedangkan W_{pit} , W_{vol} , W_{rhy} adalah *weight* atau bobot. Di mana *weight* yang ditetapkan pada penelitian (Tsai, et al., 2012) dan juga penelitian (Nugraha, et al., 2019) yaitu nilai W_{pit} sebesar 0.45, W_{vol} sebesar 0.16, dan W_{rhy} sebesar 0.39 sehingga ketika masing-masing bobot skor ditotal semua akan memperoleh hasil

sama dengan 1. Skema untuk memperoleh *overall score* atau total skor baik pada penelitian (Tsai, et al., 2012) dan penelitian (Nugraha, et al., 2019) dapat dilihat pada gambar 2.3.



Gambar 2.3 Skema untuk memperoleh *overall score*

2.12.1 Pitch Based Scoring

Untuk mendapatkan *pitch scoring* dimulai dari konversi berkas suara rekaman pengguna dengan cara menyampling (pengambilan sampel suara) terlebih dahulu. Sampling tersebut berguna untuk mengubah sinyal analog menjadi sinyal diskrit karena pada dasarnya suara manusia merupakan sinyal analog. *Frame Blocking* digunakan untuk memisah data suara menjadi beberapa *frame*, di mana dari satu *frame* tersebut menghasilkan beberapa sampel. Untuk mengubah sinyal *non-stationer* menjadi sinyal *kuasi-stationer* diperlukan adanya proses tersebut, dengan algoritma *Fast Fourier Transform* (FFT). Proses *windowing signal* suara akan dilakukan pemotongan suara menjadi *frame* yang akan diarahkan ke diskontinuitas pada *signal* awal dan akhir. Hal tersebut akan menyebabkan kesalahan data dalam proses transformasi. Maka dari itu perlu adanya *windowing signal* untuk mengurangi efek diskontinuitas pada sinyal cacah. Pada penelitian ini menggunakan bantuan *library open source* seperti penelitian sebelumnya (Nugraha, et al., 2019) yaitu menggunakan *library* atau kode pustaka dari *music9* versi 1.4.2.0 berbentuk *jar files*. Kode pustaka tersebut dapat mengakomodasi langkah-langkah yang sudah dijelaskan sebelumnya.

Setelah dilakukan sampling, *file* suara ustaz dan pengguna dilakukan konversi menjadi *spectrogram*. Kemudian hasil dari *spectrogram* akan dimasukkan ke dalam *arraylist pitch* dan *arraylist volume*. Setelah itu nilai *array pitch* dan *array volume* tadi dilakukan perbaikan menjadi nilai *harmonicPair* dengan tipe data `Pair` dari *package src.org.ioe.tprsa*. Dari nilai *harmonicPair* akan dihitung menggunakan *pitch scoring* yaitu dengan menghitung *Dynamic Time Wrapping* (DTW). Pada kelas DTW ini digunakan untuk menghitung jarak antar 2 variabel *distance* yaitu *array pitch* antara berkas audio pengguna dan berkas audio ustaz. Dari hasil *distance* tersebut akan dilakukan *mapping* sehingga menghasilkan nilai skor *pitch*.

2.12.2 Volume Based Scoring

Skema untuk mendapatkan *volume scoring* prosesnya sama dengan *pitch scoring*. Setelah mendapatkan *spectrogram* dari kedua *file* audio, selanjutnya mencari nilai *array frequency* maksimum dengan cara menggunakan fungsi dari

library musicg. Kemudian dari kembalian nilai *array* tersebut dihitung menggunakan *class DTW*. Hanya saja pada kelas *DTW* ini digunakan untuk menghitung jarak antar 2 variabel *inbeat* yaitu *array volume* antara berkas audio pengguna dan berkas audio ustaz. Dari hasil *inbeat* tersebut akan dilakukan *mapping* sehingga menghasilkan nilai skor *volume*.

2.12.3 Rhythm Based Scoring

Skema untuk mendapatkan *rhythm scoring* diawali dengan melakukan sampling terhadap kedua berkas audio rekaman pengguna dan ustaz. Setelah itu mencari fitur ekstraksi dari kedua *file* audio tersebut dengan Algoritma MFCC, sehingga didapatkan 2 nilai *array delta energy*. Setelah itu mencari jarak dari nilai *inbeat* dengan cara diproses menggunakan *class DTW*. Dari hasil *inbeat* tersebut akan dilakukan *mapping* sehingga menghasilkan nilai skor *rhythm*.

2.12.4 Timbre Based Scoring

Pada proses perhitungan *timbre scoring* ini hampir sama seperti mencari *rhythm scoring*, yaitu melakukan sampling terhadap 2 *file* audio (audio rekaman pengguna/murid dan audio ustaz). Selanjutnya mencari nilai fitur ekstraksi dari kedua *file* audio rekaman tersebut menggunakan Algoritma MFCC, sehingga didapatkan *array delta energy*. Setelah itu mencari skor dari *timbre* dengan algoritma *Gaussian Mixture Model (GMM)* seperti yang dijelaskan sebelumnya. Setelah melalui proses *GMM* tersebut akan mendapatkan nilai rata-rata (*means*) dari kedua fitur ekstraksi kedua audio rekaman yang dijadikan sebagai nilai skor *timbre*.

2.13 Pengujian Sistem

Pada subbab ini, menjelaskan tahap pengujian dari pengembangan aplikasi *scoring* Iqra, di mana pada penelitian ini akan dilakukan pengujian unit, validasi dan pengujian korelasi kemiripan antara *overall score* penilaian manual ustaz dengan penilaian otomatis dari Aplikasi *Scoring* Iqra.

2.13.1 Pengujian Unit

Pengujian unit digunakan untuk menguji apakah unit terkecil dari hasil kode program sesuai yang diharapkan oleh pengembang Aplikasi. Selain itu digunakan untuk memeriksa apakah algoritma berupa *pseudocode* dan implementasi algoritma dari metode-metode sebelumnya bisa menghasilkan perubahan *output* berupa nilai *pitch score*, *volume score*, dan *rhythm score* serta *timbre score* dapat mempengaruhi perubahan *overall score*. Sedangkan *System Development Life Cycle (SDLC)* yang digunakan untuk pengujian unit ini mengikuti *SDLC* pendekatan *Agile* dengan metode *Extreme Programming (XP)* sesuai buku (Pressman, 2009).

2.13.2 Pengujian Validasi

Pada pengujian ini bertujuan untuk memeriksa kesesuaian antara semua kebutuhan fungsional Aplikasi pada analisis kebutuhan dengan kebutuhan setelah Aplikasi di implementasikan. Pada pengujian ini tidak melihat perancangan algoritma maupun kode program sama sekali, dikarenakan pengujian ini bisa disebut teknik pengujian *black box* atau hanya melihat *input* (masukan) dan *output* (keluaran) dari *interface* aplikasinya saja.

2.13.3 Pengujian Korelasi Kemiripan

Pada pengujian ini dibutuhkan beberapa responden dengan kriteria yaitu pelajar dengan usia 8 – 15 tahun. Rata – rata semua responden tersebut sudah pernah belajar mengaji menggunakan metode iqra pada Taman Pendidikan Al-Qur'an (TPA). Setidaknya dalam kegiatan belajar mengaji tersebut, responden telah lulus pada iqra 1, 2, dan 3. Selain responden diperlukan satu ustaz, di mana ustaz berperan besar yaitu bertugas untuk menilai bacaan dari responden secara manual, di mana ustaz memiliki pengalaman baik dalam mengajar Iqra ataupun Al-Qur'an baik di sekolah maupun TPA di masjid. Selain itu responden juga dinilai oleh sistem pada aplikasi, di mana penilaian dari ustaz dan penilaian sistem pada aplikasi menghasilkan nilai berupa *overall score*. Adapun jumlah responden akan ditentukan menggunakan Rumus Slovin, yang akan dijelaskan lebih detail pada bab pengujian dan analisis korelasi kemiripan.

Pengujian korelasi kemiripan ini akan menguji antara 2 nilai *overall score* dari penilaian manual ustaz dan *overall score* dari sistem aplikasi. Sehingga dari kedua nilai tersebut akan dicari korelasi kemiripan menggunakan rumus *pearson correlation coefficient* yang juga telah digunakan pada penelitian sebelumnya (Nugraha, et al., 2019). Adapun rumus *pearson correlation coefficient* menurut (Chok, 2010) yaitu seperti persamaan 2.3.

$$r = \frac{\sum_{i=1}^n ((x_i - \bar{x})(y_i - \bar{y}))}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (2.3)$$

Dimana nilai \bar{x} dan \bar{y} , dapat dicari seperti pada persamaan 2.4.

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}, \quad \bar{y} = \frac{\sum_{i=1}^n y_i}{n} \quad (2.4)$$

Setelah dihitung menggunakan persamaan 2.3 diatas, selanjutnya didapatkan nilai r . Di mana nilai n adalah jumlah responden. Nilai x_i adalah *overall score* dari masing-masing responden yang dinilai oleh sistem. Sedangkan y_i adalah *overall score* dari masing-masing responden yang dinilai oleh ustaz. Nilai \bar{x} dan \bar{y} adalah nilai rata-rata yang dapat dicari pada persamaan 2.4. Nilai $\sum_{i=1}^n x_i$ adalah jumlah *overall score* dari semua responden yang dinilai oleh sistem. Begitupun dengan $\sum_{i=1}^n y_i$ adalah jumlah *overall score* dari semua responden yang dinilai oleh ustaz.

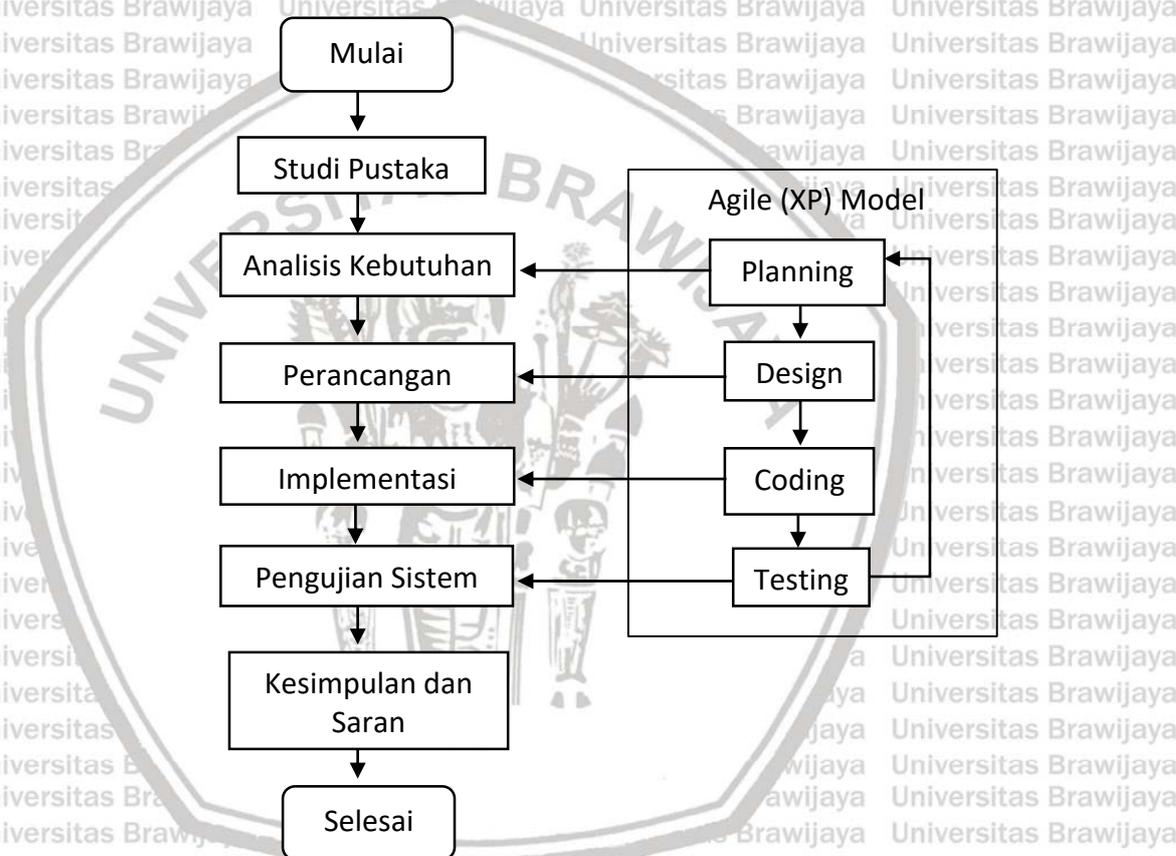
Nilai r tersebut dapat bernilai positif dan dapat bernilai negatif secara teoritis, yang terpenting korelasi kemiripan berada pada interval -1 dan $+1$. Korelasi kemiripan pada metode *pearson correlation coefficient* terjadi konsistensi hasil penilaian antara sistem aplikasi dengan penilaian manual ustaz. Ketika r bernilai positif yaitu menunjukkan hubungan yang searah, di mana pada kasus ini penilaian ustaz naik maka penilaian sistem pada aplikasi juga ikut naik, begitupun sebaliknya. Sedangkan ketika r bernilai negatif yaitu menunjukkan hubungan terbalik antara penilaian manual oleh ustaz naik dan penilaian sistem oleh aplikasi turun, begitupun sebaliknya. Tabel 2.2 menunjukkan kriteria nilai r menurut penelitian (Ratner, 2009).

Tabel 2.2 Kriteria Nilai r

Negatif	Positif	Keterangan
0	0	Tidak ada korelasi antara 2 variabel
Diantara 0 sampai -0.3	Diantara 0 sampai 0.3	Korelasi lemah
Diantara -0.3 sampai -0.7	Diantara 0.3 sampai 0.7	Korelasi sedang/cukup
Diantara -0.7 sampai -1.0	Diantara 0.7 sampai 1.0	Korelasi Kuat
-1.0	1.0	Korelasi Sempurna

BAB 3 METODOLOGI

Pada bab 3 ini akan dijelaskan mengenai alur suatu proses yang digunakan sebagai acuan dalam penelitian ini dalam pengembangan aplikasi *scoring* Iqra berbasis Android menggunakan pendekatan *Agile* dengan model *Extreme Programming* (XP) yang telah dijelaskan pada landasan kepustakaan. Pada penelitian ini akan dibahas beberapa tahapan yaitu analisis kebutuhan (*fase planning*), perancangan (*fase design*), implementasi (*fase coding*), dan pengujian sistem (*fase testing*), serta mendapatkan kesimpulan dan saran, Alur diagram metodologi penelitian ini dapat dilihat pada gambar 3.1.



Gambar 3.1 Alur diagram metodologi penelitian

3.1 Studi Pustaka

Studi pustaka merupakan informasi yang berkaitan mengenai teori-teori dasar yang relevan digunakan untuk memecahkan masalah pada penelitian ini. Sumber informasi tersebut berasal dari jurnal, artikel, buku referensi, ataupun dari *website* yang dapat dipertanggungjawabkan. Informasi tersebut kemudian dijadikan acuan atau rujukan untuk memperkuat argumen yang ada. Studi Pustaka yang digunakan pada penelitian ini adalah sebagai berikut:

1. Kajian Pustaka

2. Iqra
3. Android
4. Room
5. Pendekatan *Agile*
6. *User Stories*
7. *Class Responsibility Collaborator (CRC) Card*
8. *Feature Extraction*
9. *Mel-Frequency Cepstral Coefficient (MFCC)*
10. *Gaussian Mixture Model (GMM)*
11. Rumus Slovin
12. *Scoring* suara
13. Pengujian sistem

3.2 Analisis Kebutuhan

Pada analisis kebutuhan ini merupakan proses awal dalam membuat aplikasi *scoring Iqra*. Pada alur *Extreme Programming* fase analisis kebutuhan disebut juga sebagai *fase planning*. Pada proses ini meliputi: pembuatan *storyboard*, arsitektur sistem, pengumpulan data, menampilkan kebutuhan sistem lama, dan menganalisis kebutuhan sistem baru seperti: identifikasi aktor, analisis kebutuhan fungsional, analisis kebutuhan non fungsional, dan *use case diagram*, serta *use case scenario* yang diambil dari *user stories*. Pada sub-bab arsitektur sistem dijelaskan bagaimana cara kerja aplikasi *scoring Iqra* secara umum beserta pengumpulan datanya yaitu dengan mengunjungi ustaz untuk melakukan rekaman audio Iqra sebagai referensi yang telah dilakukan pada penelitian sebelumnya. Adapun pada penelitian ini masih menggunakan rekaman audio ustaz yang sama sebagai referensi tajwid bacaan huruf hijaiyah.

Adanya kebutuhan fungsional pada sistem baru, maka untuk identifikasi aktor, analisis kebutuhan fungsional, dan kebutuhan non fungsional terdapat perubahan (*refactoring*) dari penelitian sebelumnya, di mana Aplikasi *Scoring Iqra* ini terdapat penambahan fitur riwayat skor pengerjaan soal sebagai evaluasi latihan. Sedangkan pada non fungsionalnya terdapat *reability* aplikasi dalam perhitungan *overall score* yang berasal dari parameter *pitch*, *volume*, *rhythm*, dan *timbre*. Adapun untuk mencari nilai dari *timbre score* menggunakan algoritma GMM. Sedangkan mengenai perubahan *use case diagram* dan *use case scenario* akan menyesuaikan dari kebutuhan fungsional pada sistem baru.

3.3 Perancangan

Pada proses ini bertujuan untuk membuat perancangan sistem setelah proses analisis kebutuhan. Proses ini sering disebut *fase design* pada alur model *Extreme*

Programming. Bahasa pemodelan pada model *Extreme Programming* yang digunakan pada proses ini tidak menjelaskan secara detail mengenai *Unified Modelling Language* (UML), akan tetapi langsung merancang Algoritma yang digunakan untuk menghitung *score*. Jadi pada penelitian ini hanya fokus menjelaskan perancangan algoritma yang digunakan untuk mencari skor dari 4 nilai parameter yaitu nilai *pitch*, *volume*, *rhythm*, dan *timbre* yang nantinya akan menghasilkan satu nilai yaitu *overall score*. Sedangkan untuk perancangan *User Interface* (UI) terdapat modifikasi dari UI penelitian sebelumnya (Nugraha, et al., 2019). Di mana pada UI aplikasi ini dibuat lebih interaktif lagi terhadap penggunaanya.

3.4 Implementasi

Pada proses ini termasuk proses yang terpenting, di mana algoritma yang dirancang pada proses perancangan diterapkan pada proses ini. Proses implementasi sering disebut *fase coding* pada alur model *Extreme Programming*. Pada proses ini mengimplementasikan algoritma *timbre scoring* ke dalam penelitian sebelumnya. Sehingga diharapkan setelah menambahkan algoritma tersebut pada aplikasi *scoring Iqra* ini terdapat perubahan hasil korelasi kemiripan ketika membandingkan antara *overall score* dari penilaian manual ustaz dengan penilaian aplikasi. Sedangkan *software* yang digunakan untuk merancang aplikasi ini yaitu menggunakan *software* *Android Studio 4.2.1*. Menggunakan bahasa pemrograman Java dan XML untuk membuat tampilan antarmuka Aplikasi.

3.5 Pengujian Sistem

Pada fase pengujian pada penelitian ini yaitu melakukan pengujian unit, pengujian validasi dan pengujian korelasi kemiripan saja. Dikarenakan penelitian ini fokus pada batasan masalah yang dibuat. Pada proses ini disebut juga sebagai *fase testing* pada alur model *Extreme Programming*, di mana *unit testing* yang digunakan yaitu menggunakan *library Junit4 Testing*.

Pengujian korelasi kemiripan akan dilakukan dengan 14 responden, di mana semua responden merekam suara bacaan 3 konten huruf hijaiyah dari masing-masing jilid *Iqra* yang diberikan oleh Aplikasi. Setelah itu, rekaman dinilai oleh aplikasi dan ustaz yang nantinya *overall score* yang diperoleh dari masing-masing responden akan dibandingkan sehingga diperoleh korelasi kemiripan antara penilaian ustaz dengan aplikasi menggunakan *pearson correlation coefficient* (Ratner, 2009). Di mana rumus metode *pearson correlation coefficient* telah dijelaskan pada bab sebelumnya.

3.6 Kesimpulan

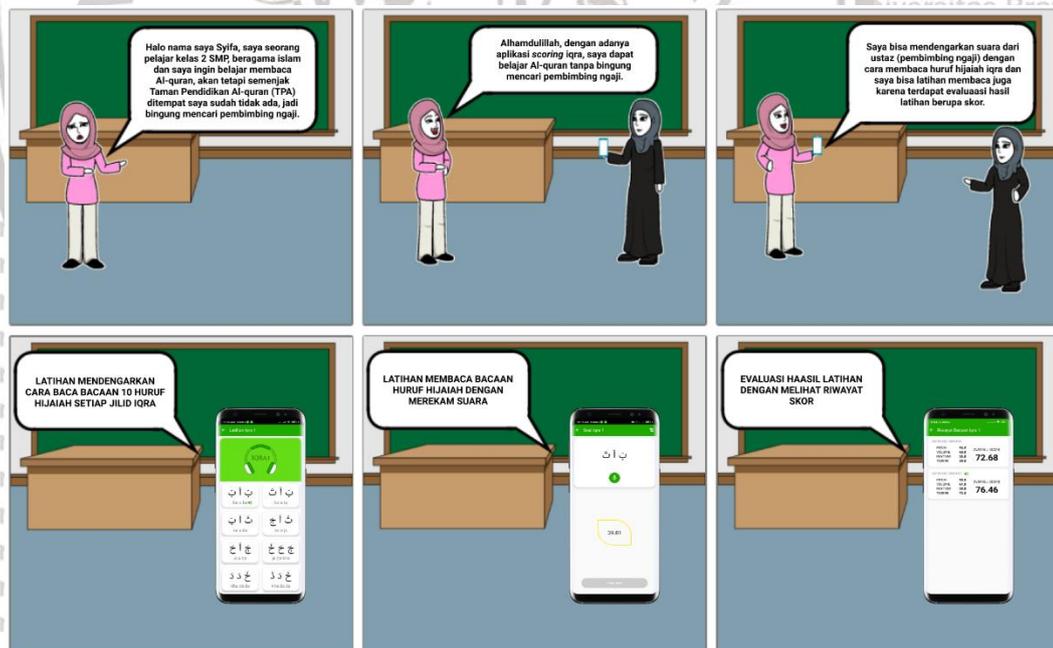
Pada proses ini bertujuan untuk menarik kesimpulan yang didapat setelah melakukan penelitian ini. Selain itu pada proses ini juga memberikan saran-saran yang belum bisa tercapai pada penelitian ini, untuk pengembangan lebih lanjut pada penelitian selanjutnya.

BAB 4 ANALISIS KEBUTUHAN

Dalam penelitian ini akan dilakukan pengembangan lanjut pada aplikasi *Scoring Iqra*, di mana kebutuhan fungsional dan beberapa fitur yang ada pada aplikasi sebelumnya masih tetap ada. Hanya saja pada penelitian ini terdapat beberapa kebutuhan fungsional baru. Pada bab ini akan menjelaskan gambaran dari aplikasi berupa *storyboard*, arsitektur sistem, pengumpulan data, menampilkan kebutuhan sistem yang ada pada sistem lama dan juga analisis dari kebutuhan sistem baru yang diambil dari *user stories*.

4.1 Storyboard Aplikasi Scoring Iqra

Storyboard adalah suatu papan cerita singkat berupa gambar yang disusun secara berurutan. Pada *storyboard* ini digunakan untuk menjelaskan secara umum Aplikasi *Scoring Iqra*. Ilustrasi cerita dimulai dari seorang pelajar kelas 2 SMP yang tidak bisa belajar Al-quran dikarenakan tidak memiliki pembimbing ngaji (membaca al-quran) di lingkungannya. Kemudian setelah mengetahui terdapat aplikasi yang dapat membimbing untuk berlatih membaca Al-quran dengan metode iqra, pelajar tersebut dapat mengevaluasi dari latihan belajarnya. Ilustrasi cerita tersebut digambarkan seperti yang ada pada gambar 4.1.



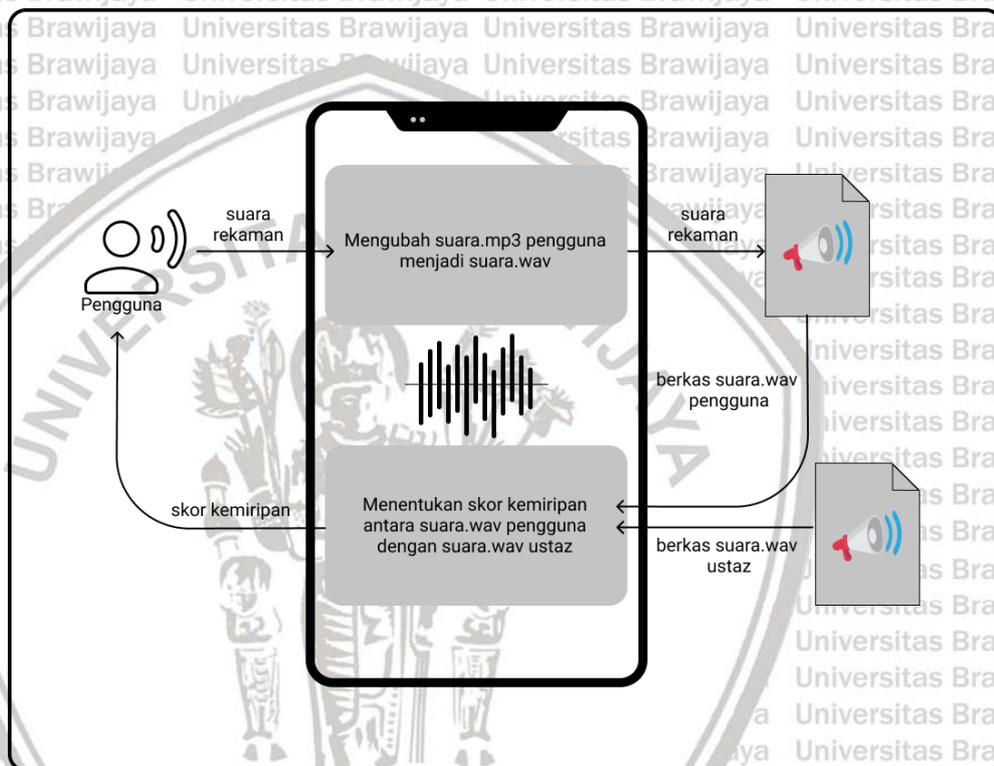
Gambar 4.1 Storyboard Aplikasi Scoring Iqra

4.2 Arsitektur Sistem

Pada arsitektur sistem ini akan menjelaskan secara detail mengenai cara kerja Aplikasi *Scoring Iqra*. Secara garis besar aplikasi ini menyediakan 3 menu pilihan yaitu pengguna dapat mendengarkan cara baca huruf hijaiyah dari suara ustaz, pengguna dapat merekam suaranya untuk menjawab soal bacaan huruf hijaiyah,

dan pengguna dapat memilih bantuan atau tata cara penggunaan Aplikasi *Scoring Iqra*. Pada aplikasi ini masih mempertahankan tingkat kesulitan dari bacaan Iqra sebanyak 3 jilid atau semakin tinggi jilid yang dipilih oleh pengguna, maka tingkat kesulitan bacaan huruf hijaiyah semakin sulit.

Adapun Penyimpanan suara rekaman yang dilakukan oleh pengguna, akan disimpan pada penyimpan lokal perangkat. Sedangkan suara referensi dari ustaz sudah termasuk *resource* dari aplikasi, jadi pengguna tidak perlu mengunduh terlebih dahulu. Arsitektur sistem dari Aplikasi *Scoring Iqra* dapat dilihat pada gambar 4.2.



Gambar 4.2 Arsitektur Aplikasi *Scoring Iqra*

Berikut adalah gambaran kasar mengenai alur kerja menghitung skor kemiripan suara antara pengguna dengan suara referensi dari ustaz pada Aplikasi *Scoring Iqra* pada gambar 4.2 yaitu antara lain:

1. Sebelum merekam pengucapan bacaan huruf hijaiyah yang dipilih, pengguna harus menekan tombol dengan *icon microphone* on terlebih dahulu. Kemudian ketika bacaan huruf hijaiyah sudah selesai, pengguna menekan tombol dengan *icon microphone* yang disilang.
2. Proses perekaman suara pengguna, secara otomatis telah direkam menjadi berkas suara berekstensi .wav dan tersimpan pada penyimpanan lokal perangkat milik pengguna. Kemudian muncul *pop up* notifikasi bahwa berkas suara pengguna tersimpan.

3. Setelah muncul *pop up* notifikasi rekaman telah tersimpan, pengguna dapat menekan tombol “lihat skor” untuk menghitung skor kemiripan antara suara referensi dari ustaz dengan suara rekaman pengguna.

4. Kemudian nilai skor akan muncul pada layar perangkat Android pengguna. Apabila pengguna ingin merekam suaranya lagi, maka dapat mengulangi tahap 1 – 4 diatas.

5. Pada proses pengolahan suara dari kedua berkas berentensi .wav antara suara ustaz dengan suara pengguna akan di hitung kemiripannya dengan 4 kriteria parameter yaitu *pitch, volume, rhythm, dan timbre*. Di mana berkas suara .wav pengguna akan terus di simpan dengan nama yang berbeda-beda, dikarenakan akan dijadikan sebagai riwayat skor latihan pengguna.

4.3 Pengumpulan Data

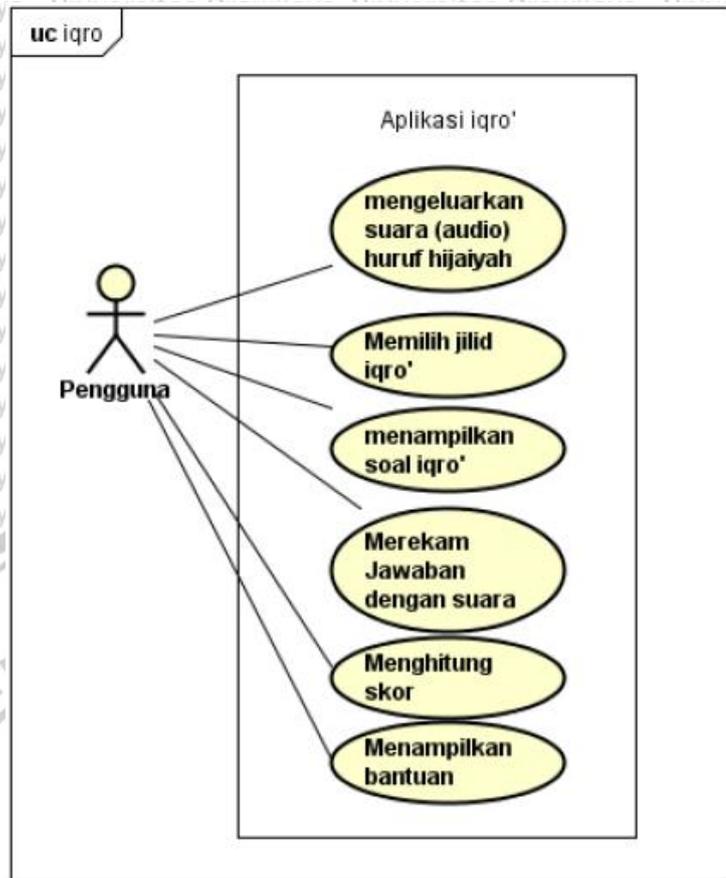
Pada proses pengumpulan data ini di tujukan pada pengumpulan data suara referensi atau *sample* rekaman bacaan Iqra yang dilakukan oleh ustaz pada jilid 1, 2, dan 3. Masing-masing jilid terdapat 10 berkas suara rekaman dengan bacaan Iqra yang berbeda-beda sesuai metode Iqra. Pada penelitian ini masih menggunakan sumber data pada penelitian sebelumnya. Di mana dari total 30 berkas suara rekaman tersebut dilakukan pada ruang yang cukup sunyi dan tidak ada yang di *edit* menggunakan *software* apapun.

4.4 Kebutuhan Sistem Lama

Pada subbab ini akan dijelaskan mengenai kebutuhan sistem dan hasil implementasi kebutuhan sistem penelitian sebelumnya (Nugraha, et al., 2019).

4.4.1 Use Case Diagram

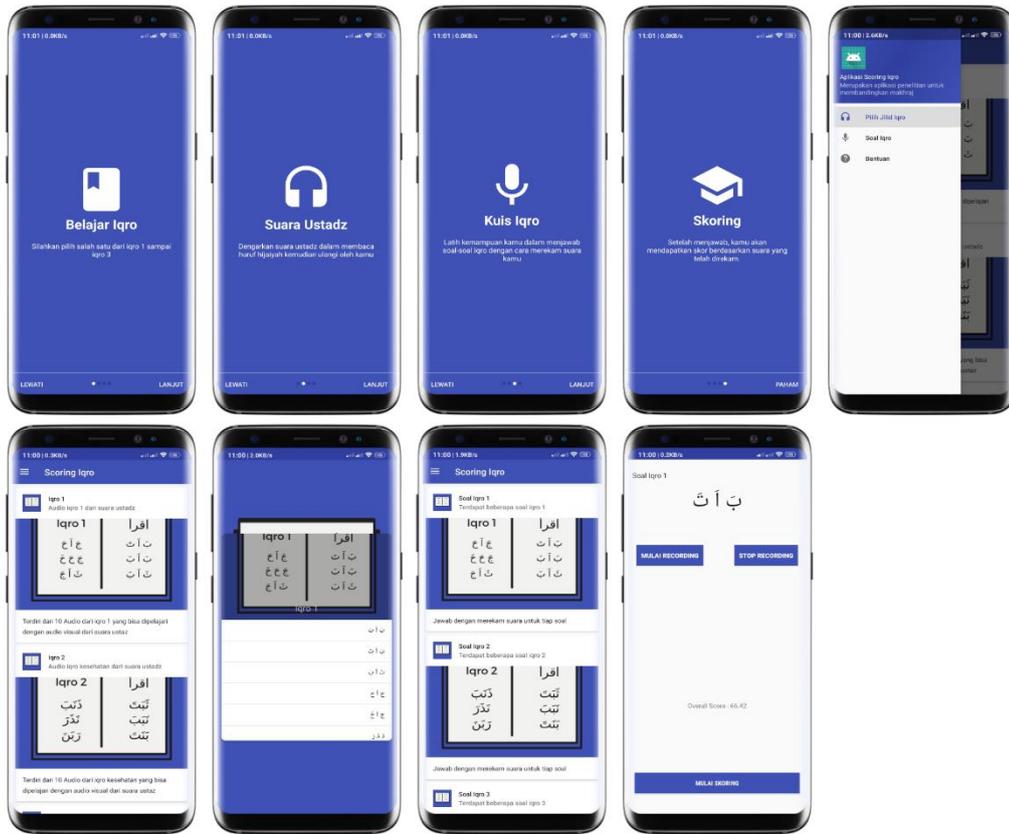
Pada kebutuhan sistem pada penelitian sebelumnya, terdapat 6 kebutuhan fungsional yang ada di Aplikasi *Scoring Iqro'*. Adapun fungsional tersebut adalah mengeluarkan suara (audio) huruf hijaiyah, memilih jilid iqro', menampilkan soal iqro', merekam jawaban dengan suara, menghitung skor, dan menampilkan bantuan. *Use case diagram* kebutuhan sistem pada penelitian sebelumnya dapat dilihat pada gambar 4.3.



Gambar 4.3 Use Case Diagram Sistem Lama

4.4.2 User Interface

Dari 6 kebutuhan fungsional sistem lama pada gambar 4.3, telah diimplementasikan ke dalam *User Interface* Aplikasi *Scoring Iqro'*. Hasil tampilan Aplikasi sesuai kebutuhan sistem tersebut dapat dilihat pada gambar 4.4.



Gambar 4.4 User Interface Sistem Lama

4.5 Analisis Kebutuhan Sistem Baru

4.5.1 Identifikasi Aktor

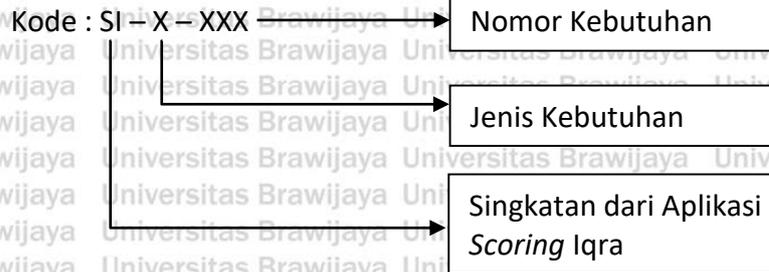
Pada bagian ini akan dijelaskan mengenai aktor yang akan menggunakan Aplikasi *Scoring Iqra* berbasis Android, dimana aktor yang dipilih difokuskan kepada pelajar yang ingin belajar mengaji melalui Aplikasi *Scoring Iqra* seperti yang dijelaskan pada Tabel 4.1.

Tabel 4.1 Identifikasi Aktor

No	Aktor	Deskripsi
1	Pengguna	Pengguna adalah sebutan bagi orang yang akan menggunakan Aplikasi <i>Scoring Iqra</i> . Pengguna ini difokuskan kepada pelajar yang ingin belajar membaca Al-Qur'an dengan metode Iqra.

4.5.2 Aturan Penomoran

Skema aturan penomoran dapat dilihat pada gambar 4.5.



Gambar 4.5 Aturan Penomoran

Keterangan dari aturan penomoran dapat dilihat pada Tabel 4.2.

Tabel 4.2 Keterangan aturan penomoran kebutuhan

Kode	Aturan Penomoran	Keterangan
SI	SI	Nama Aplikasi <i>Scoring Iqra</i> (SI)
X	1	Kebutuhan Fungsional
X	2	Kebutuhan Non-Fungsional
XXX	001..n	Kode Kebutuhan Pengguna

4.5.3 Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang menjelaskan kemampuan yang dimiliki oleh aplikasi. Adapun Kebutuhan fungsional pada sistem baru dapat dilihat pada Tabel 4.3.

Tabel 4.3 Kebutuhan fungsional pengguna

No	Kode Fungsi	Nama Fungsi	Deskripsi
1.	SI – 1 – 001	Menampilkan 3 menu jilid iqra	Sistem dapat menyediakan 3 jilid iqra yaitu jilid 1 sampai jilid 3. Adapun pemilihan jilid ini dapat dilakukan pada menu soal maupun menu latihan.
2.	SI – 1 – 002	Membunyikan 10 suara bacaan huruf hijaiyah	Sistem dapat membunyikan / mengeluarkan suara bacaan huruf hijaiyah yang dipilih oleh pengguna. Pada menu latihan ini terdapat 3 jilid iqra, dimana setiap jilid terdapat 10 bacaan huruf hijaiyah. Fungsi ini dapat dilakukan pada halaman latihan.
3.	SI – 1 – 003	Menampilkan 1 soal bacaan huruf hijaiyah	Sistem dapat menampilkan soal berupa bacaan huruf hijaiyah, di mana terdapat 3 jilid iqra, dimana setiap jilid terdapat 1 soal bacaan huruf hijaiyah yang harus

			dijawab dengan cara merekam suara cara baca huruf hijaiyah tersebut. Fungsi ini dapat dilakukan pada halaman soal.
4.	SI – 1 – 004	Merekam suara untuk menjawab soal	Sistem dapat merekam suara pengguna dengan menyediakan fitur rekam suara, yang nantinya pengguna dapat menjawab soal bacaan huruf hijaiyah yang telah dipilih.
5.	SI – 1 – 005	Menghitung skor kemiripan suara rekaman	Sistem dapat menghitung skor kemiripan antara suara rekaman pengguna dengan suara rekaman ustaz, setelah pengguna merekam suara bunyi bacaan huruf hijaiyah yang dipilih.
6.	SI – 1 – 006	Menampilkan cara penggunaan aplikasi	Sistem dapat menyediakan halaman bantuan, supaya membantu pengguna yang belum mengerti cara penggunaan aplikasi <i>scoring iqra</i> .
7.	SI – 1 – 007	Menampilkan riwayat skor pengerjaan soal	Sistem dapat menyediakan halaman riwayat latihan pengerjaan soal berupa daftar skor dari 4 parameter (<i>pitch, volume, rhythm, dan timbre</i>) dan juga <i>overall score</i> .
8.	SI – 1 – 008	Membunyikan riwayat suara rekaman	Sistem dapat mengeluarkan suara bacaan huruf hijaiyah dari latihan mengerjakan soal bacaan huruf hijaiyah, ketika menekan salah satu daftar riwayat skor latihan.
9.	SI – 1 – 009	Menghapus riwayat skor pengerjaan soal	Sistem dapat menyediakan fitur untuk menghapus daftar riwayat skor yang diperoleh dari hasil latihan mengerjakan soal oleh pengguna.

4.5.4 Kebutuhan Non Fungsional

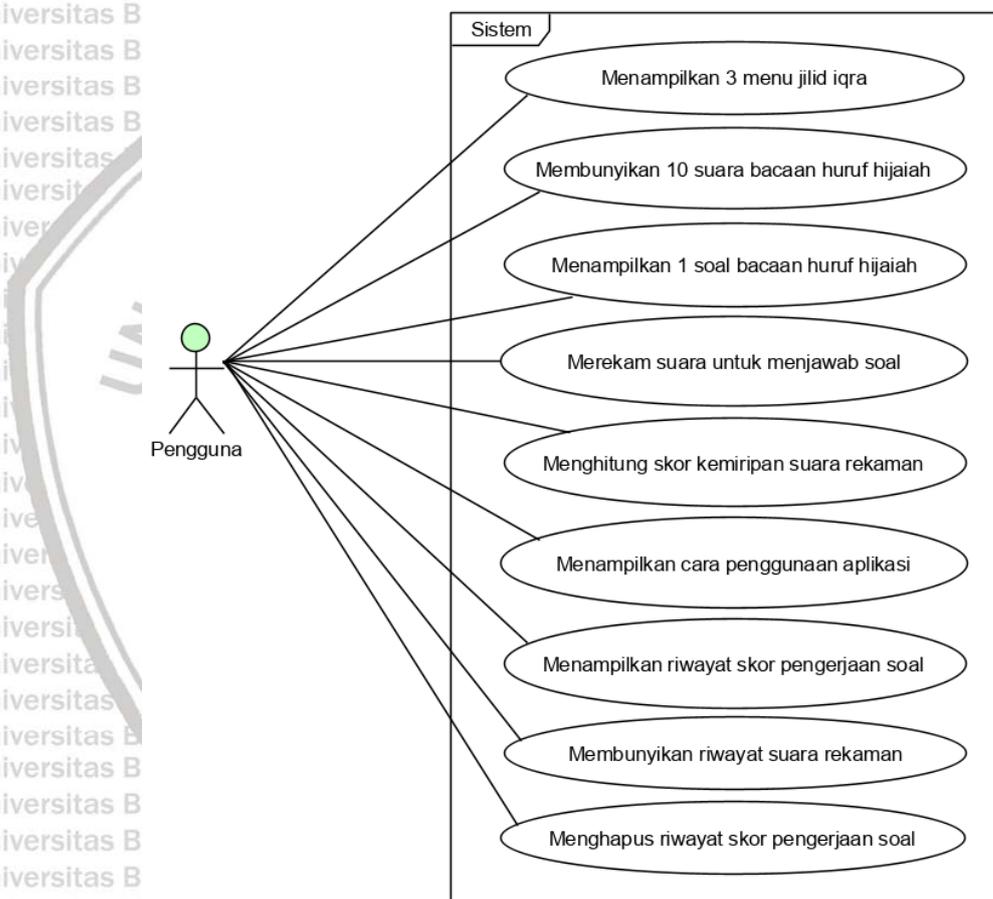
Kebutuhan non fungsional pada penelitian ini hanya mencakup kemampuan dalam kategori *Reliability* (Kehandalan) yaitu mampu mempertahankan tingkat kinerja / *performance* berupa akurasi, konsistensi, ataupun toleransi kesalahan dari Aplikasi. Tabel 4.4 merupakan kebutuhan non fungsional Aplikasi *Scoring Iqra*.

Tabel 4.4 Kebutuhan non fungsional pengguna

No	Kode Fungsi	Deskripsi
1.	SI – 2 – 001	Sistem dapat menghitung <i>overall score</i> yang berasal dari parameter <i>pitch</i> , <i>volume</i> , <i>rhythm</i> , dan <i>timbre</i> . <i>Overall Score</i> memiliki rentang nilai antara 0,00 sampai 100,00.

4.5.5 Use Case Diagram

Use case diagram adalah diagram yang menunjukkan interaksi apa saja yang dapat diberikan oleh sistem kepada aktor. Use case diagram Aplikasi Scoring Iqra dapat dilihat pada gambar 4.4.



Gambar 4.6 Use Case Diagram Sistem Baru

4.5.6 Use Case Scenario

Pada bagian ini akan menjelaskan secara detail mengenai masing-masing use case diagram pada gambar 4.6. Adapun use case scenario dari setiap fungsionalnya dapat dilihat pada tabel 4.5 sampai 4.13. Pada Tabel 4.5 merupakan skenario untuk menjalankan fungsional sistem dalam menampilkan 3 menu jilid iqra.

Tabel 4.5 Use Case Scenario menampilkan 3 menu jilid iqra

<i>Actor</i>	Pengguna
<i>Pre-Condition</i>	Pengguna sudah berada pada halaman utama aplikasi <i>Scoring Iqra</i> .
<i>Main Flow</i>	1. Jika pengguna ingin memilih jilid iqra pada menu latihan, maka pengguna sudah bisa memilih jilid yang telah disediakan oleh sistem pada halaman utama aplikasi.
	2. Jika pengguna ingin memilih jilid iqra pada menu soal, maka pengguna harus masuk pada menu soal terlebih dahulu. Dengan cara pengguna menekan tombol <i>navigation drawer header</i> pada pojok kiri atas aplikasi.
	3. Pengguna memilih menu soal.
<i>Alternative Flow</i>	1. Ketika pengguna berada pada menu soal dan ingin kembali ke menu latihan, maka perlu menekan tombol <i>navigation drawer header</i> pada pojok kiri atas aplikasi.
	2. Pengguna memilih menu latihan
<i>Post-Condition</i>	Sistem dapat menampilkan semua jilid yang ada, baik dalam menu latihan ataupun menu soal. Adapun jilid tersebut bermaksud sebagai tingkat kesulitan bacaan iqra, di mana semakin tinggi jilid maka semakin sulit bacaan iqra tersebut. Sehingga pengguna dapat memilih dari ketiga jilid yang telah dibuat oleh sistem.

Tabel 4.6 merupakan skenario untuk menjalankan fungsional sistem dalam membunyikan 10 suara bacaan huruf hijaiyah dari suara ustaz yang ada pada setiap jilid iqra.

Tabel 4.6 Use Case Scenario membunyikan 10 suara bacaan huruf hijaiyah

<i>Actor</i>	Pengguna
<i>Pre-Condition</i>	Pengguna sudah berada pada halaman utama aplikasi <i>Scoring Iqra</i> .
<i>Main Flow</i>	1. Pengguna memilih salah satu jilid dari ketiga jilid yang ada.
	2. Pengguna memilih salah satu huruf hijaiyah.

<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>navigation drawer header</i> yang ada pada pojok kiri atas aplikasi. 2. Pengguna memilih menu latihan, kemudian sistem menampilkan daftar ketiga jilid iqra. 3. Pengguna memilih salah satu jilid iqra, sistem akan menampilkan 10 kata / huruf hijaiyah. 4. Pengguna memilih salah satu huruf hijaiyah tersebut.
<i>Post-Condition</i>	Sistem akan membunyikan rekaman suara dari ustaz mengenai cara baca bacaan kalimat hijaiyah yang dipilih oleh pengguna. Huruf yang mengeluarkan bunyi ditandai dengan munculnya <i>icon speaker</i> .

Tabel 4.7 merupakan skenario untuk menjalankan fungsional sistem dalam menampilkan 1 soal bacaan huruf hijaiyah yang ada pada setiap jilid iqra.

Tabel 4.7 Use Case Scenario menampilkan 1 soal bacaan huruf hijaiyah

<i>Actor</i>	Pengguna
<i>Pre-Condition</i>	Pengguna sudah berada pada halaman utama aplikasi <i>Scoring Iqra</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>navigation drawer header</i> pada pojok kiri atas aplikasi. 2. Pengguna memilih menu soal, kemudian sistem akan menampilkan daftar ketiga jilid iqra. 3. Pengguna memilih salah satu jilid iqra, dan sistem akan menampilkan 1 soal bacaan huruf hijaiyah untuk dikerjakan oleh pengguna.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan menampilkan soal berdasarkan jilid yang dipilih oleh pengguna.

Tabel 4.8 merupakan skenario untuk menjalankan fungsional sistem dalam merekam suara untuk menjawab soal bacaan huruf hijaiyah yang ada pada setiap jilid iqra.

Tabel 4.8 Use Case Scenario merekam suara untuk menjawab soal

<i>Actor</i>	Pengguna
<i>Pre-Condition</i>	Pengguna sudah berada pada halaman utama aplikasi <i>Scoring Iqra</i> .
<i>Main Flow</i>	1. Pengguna menekan tombol <i>navigation drawer header</i> pada pojok kiri atas aplikasi.
	2. Pengguna memilih menu soal, kemudian sistem akan menampilkan daftar ketiga jilid iqra.
	3. Pengguna memilih salah satu jilid iqra, dan sistem akan menampilkan soal untuk dikerjakan oleh pengguna.
	4. Sebelum merekam suaranya, pengguna harus memilih tombol <i>icon microphone on</i> terlebih dahulu. Dan sistem akan mulai merekam suara yang ditandai muncul animasi <i>microphone</i> .
	5. Kemudian pengguna harus mengakhiri rekaman suaranya, dengan cara menekan tombol <i>icon microphone off</i> . Kemudian sistem akan menyimpan berkas rekaman suara pengguna ke dalam penyimpanan lokal dari perangkat pengguna berupa berkas audio yang berformat (.wav). Sistem juga akan memberikan <i>feedback</i> kepada pengguna berupa notifikasi bahwa rekaman telah berhasil disimpan.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan menyimpan berkas rekaman suara pengguna ke dalam penyimpanan lokal dari perangkat pengguna berupa berkas audio yang berformat (.wav). Sistem juga akan memberikan <i>feedback</i> kepada pengguna berupa notifikasi bahwa rekaman telah berhasil disimpan.

Tabel 4.9 merupakan skenario untuk menjalankan fungsional sistem dalam menghitung skor kemiripan antara kedua suara rekaman pengguna dan rekaman ustaz berupa *overall score*.

Tabel 4.9 Use Case Scenario menghitung skor kemiripan suara rekaman

Actor	Pengguna
Pre-Condition	Pengguna telah berhasil menyimpan rekaman suara dari soal bacaan huruf hijaiyah yang dipilih ke dalam penyimpanan lokal perangkat pengguna berupa berkas audio yang berformat (.wav). Pengguna juga telah menerima <i>feedback</i> dari sistem berupa notifikasi bahwa rekaman telah berhasil disimpan.
Main Flow	<ol style="list-style-type: none"> 1. Pengguna menekan tombol “lihat skor” yang ada pada halaman untuk merekam suara. Sistem akan memulai proses kalkulasi <i>overall score</i> antara rekaman suara dari pengguna dengan suara referensi dari ustaz. 2. Pengguna akan menerima hasil skor pada tengah layar perangkat pengguna yang telah ditampilkan oleh sistem. Skor yang ditampilkan pada rentang 0,00 sampai 100,00.
Alternative Flow	-
Post-Condition	Sistem akan menghitung skor kemiripan suara rekaman antara milik pengguna dengan milik ustaz, dan dapat menampilkan skor tersebut pada perangkat pengguna. Sehingga pengguna dapat mengetahui tingkat ketepatan dalam membaca huruf hijaiyah yang dipilih oleh pengguna.

Tabel 4.10 merupakan skenario untuk menjalankan fungsional sistem dalam menampilkan cara penggunaan aplikasi berupa *Onboarding Screen*.

Tabel 4.10 Use Case Scenario menampilkan cara penggunaan aplikasi

Actor	Pengguna
Pre-Condition	Pengguna sudah berada pada halaman utama aplikasi <i>Scoring Iqra</i> .
Main Flow	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>navigation drawer header</i> pada pojok kiri atas aplikasi. 2. Pengguna memilih menu bantuan, kemudian sistem akan menampilkan <i>Onboarding Screen</i>. 3. Pengguna dapat menekan tombol “lanjut” untuk mengetahui cara penggunaan aplikasi selanjutnya atau bisa juga dengan cara <i>swipe/</i>

	geser ke kanan dan geser ke kiri untuk kembali ke cara penggunaan aplikasi sebelumnya. Serta pengguna dapat menekan tombol “lewati” untuk keluar dari menu bantuan.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan menampilkan bantuan mengenai cara penggunaan aplikasi <i>Scoring Iqra</i> berupa <i>Onboarding Screen</i> . Sehingga pengguna dapat mengetahui fungsional utama dari aplikasi tersebut.

Tabel 4.11 merupakan skenario untuk menjalankan fungsional sistem dalam menampilkan riwayat skor pengerjaan soal yang ada pada setiap jilid iqra.

Tabel 4.11 Use Case Scenario menampilkan riwayat skor pengerjaan soal

<i>Actor</i>	Pengguna
<i>Pre-Condition</i>	Pengguna sudah berada pada halaman mengerjakan soal bacaan huruf hijaiyah.
<i>Main Flow</i>	1. Pengguna menekan tombol <i>icon history</i> pada pojok kanan atas aplikasi.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan menampilkan halaman riwayat skor pengerjaan soal pengguna dalam bentuk <i>list</i> yang menampilkan 4 skor parameter suara (<i>pitch, volume, rhythm, dan timbre</i>), dan <i>overall score</i> , beserta tanggal dan waktu pengerjaan soal.

Tabel 4.12 merupakan skenario untuk menjalankan fungsional sistem dalam membunyikan riwayat suara rekaman pengguna sebagai evaluasi latihan.

Tabel 4.12 Use Case Scenario membunyikan riwayat suara rekaman

<i>Actor</i>	Pengguna
<i>Pre-Condition</i>	Pengguna sudah berada pada halaman mengerjakan soal bacaan huruf hijaiyah.
<i>Main Flow</i>	1. Pengguna menekan tombol <i>icon history</i> pada pojok kanan atas aplikasi. 2. Pengguna menekan salah satu daftar riwayat skor pengerjaan soal
<i>Alternative Flow</i>	-

<i>Post-Condition</i>	Sistem akan membunyikan rekaman suara pengguna, yang dapat ditandai dengan munculnya <i>icon speaker</i> pada daftar riwayat yang dipilih.
-----------------------	--

Tabel 4.13 merupakan skenario untuk menjalankan fungsional sistem dalam menghapus riwayat skor pengerjaan soal yang dapat dilakukan pada daftar riwayat skor semua jilid iqra.

Tabel 4.13 Menghapus riwayat skor pengerjaan soal

<i>Actor</i>	Pengguna
<i>Pre-Condition</i>	Pengguna sudah berada pada halaman mengerjakan soal bacaan huruf hijaiyah.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>icon history</i> pada pojok kanan atas aplikasi. 2. Pengguna melakukan menekan dan tahan pada salah satu daftar riwayat yang ingin dihapus. 3. Sistem akan menampilkan <i>pop up</i> konfirmasi ke pengguna, untuk memastikan bahwa pengguna benar-benar ingin menghapus daftar riwayat tersebut. 4. Pengguna memilih tombol "IYA" untuk melanjutkan hapus riwayat skor pengerjaannya, atau tombol "TIDAK" untuk membatalkan menghapus daftar riwayat skor pengerjaannya.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan menghapus daftar riwayat skor pengerjaan dari lokal database.

4.6 User Stories

Pada bagian ini dilakukan karena mematuhi pedoman yang ada pada *Ebook* ditulis oleh (Ambler, 2002). Pada buku tersebut mengatakan bahwa *user stories* adalah bagian terpenting dari metode *Extreme Programming (XP)*. Di mana *User Stories* dijadikan sebagai alternatif dari *use case diagram* dan *use case scenario*. Adapun *User Stories* dapat dilihat pada Tabel 4.14.

Tabel 4.14 *User Stories* pada Aplikasi *Scoring Iqra*

No.	<i>User Stories</i>
1.	“Sebagai pengguna, saya ingin dapat belajar membaca huruf hijaiyah secara <i>audio</i> dan <i>visual</i> dengan menekan salah satu bacaan Iqra, saya dapat mendengarkan cara baca huruf hijaiyah tersebut dari suara ustaz.”
2.	“Sebagai pengguna, saya ingin dapat melihat soal dengan huruf hijaiyah yang telah disiapkan oleh aplikasi.”
3.	“Sebagai pengguna, saya ingin bisa merekam suara untuk menjawab soal huruf hijaiyah yang diberikan oleh aplikasi.”
4.	“Sebagai pengguna, saya ingin dapat melihat skor akhir atau <i>overall score</i> setelah saya melakukan rekaman suara bacaan dari salah satu soal huruf hijaiyah yang saya pilih dan menekan tombol “lihat skor” pada aplikasi.”
5.	“Sebagai pengguna, saya ingin dapat mengetahui tata cara penggunaan aplikasi <i>Scoring Iqra</i> dengan UI yang mudah dipahami.”
6.	“Sebagai pengguna, saya ingin dapat memilih jilid Iqra seperti pada buku fisik Iqra pada umumnya, sehingga dapat dicari dengan mudah.”
7.	“Sebagai pengguna, saya ingin dapat melihat riwayat skor pengerjaan soal bacaan Iqra, sehingga saya dapat mengetahui skor-skor sebelumnya.”
8.	“Sebagai pengguna, saya ingin dapat mengevaluasi riwayat pengucapan pengerjaan soal bacaan Iqra.”
9.	“Sebagai pengguna, saya ingin dapat menghapus daftar riwayat skor pengerjaan soal.”

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini menjelaskan 2 tahap sekaligus yaitu proses perancangan dan proses implementasi dari pengembangan Aplikasi *Scoring Iqra* berbasis Android. Pada subbab perancangan nanti terdapat beberapa bagian perancangan yaitu perancangan *sequence diagram*, *class diagram*, perancangan algoritma perhitungan 4 skor dan pembobotan skor dalam perhitungan *overall score*, selain itu dilakukan perancangan basis data, *CRC Cards*, dan juga perancangan *User Interface* (UI). Kemudian setelah subbab perancangan selesai dibuat, akan dilanjutkan ke tahap proses implementasi, di mana pada proses implementasi ini akan menjelaskan implementasi tahap perancangan yang telah dibuat sebelumnya. Adapun proses implementasi meliputi batasan implementasi yang menjelaskan spesifikasi perangkat lunak dan perangkat keras dalam pengembangan Aplikasi, implementasi algoritma, basis data, dan *User Interface* (UI).

5.1 Perancangan

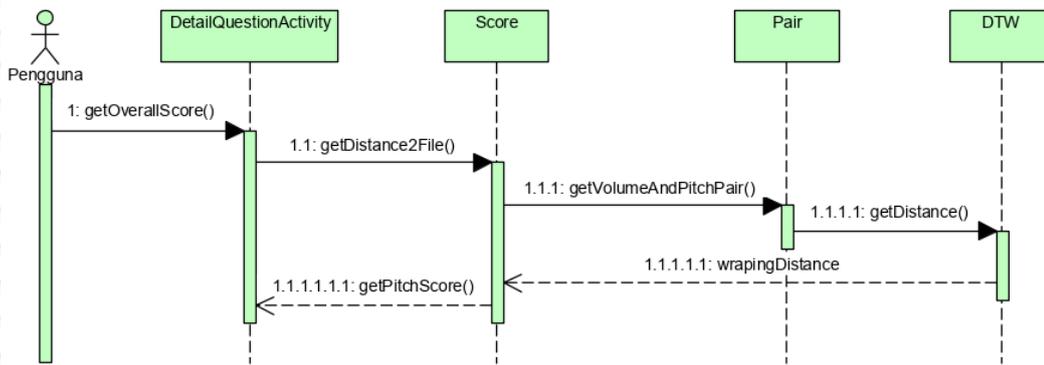
Setelah analisis kebutuhan selesai, maka proses perancangan dapat dimulai. Hasil dari proses analisis kebutuhan berupa kebutuhan fungsional dan hasil dari pemodelan kebutuhan tersebut berupa *use case diagram* dan *use case scenario* yang menjadi dasar perancangan sistem ini.

5.1.1 Perancangan *Sequence Diagram*

Pada subbab ini menjelaskan bagaimana hubungan antara pengguna dengan aplikasi *scoring iqra*. *Sequence diagram* ini berguna untuk memperjelas input dari pengguna dan output dari sistem. Di mana akan dijelaskan 4 *sequence diagram* dalam mencari nilai *overall score*.

5.1.1.1 *Sequence Diagram Pitch Scoring*

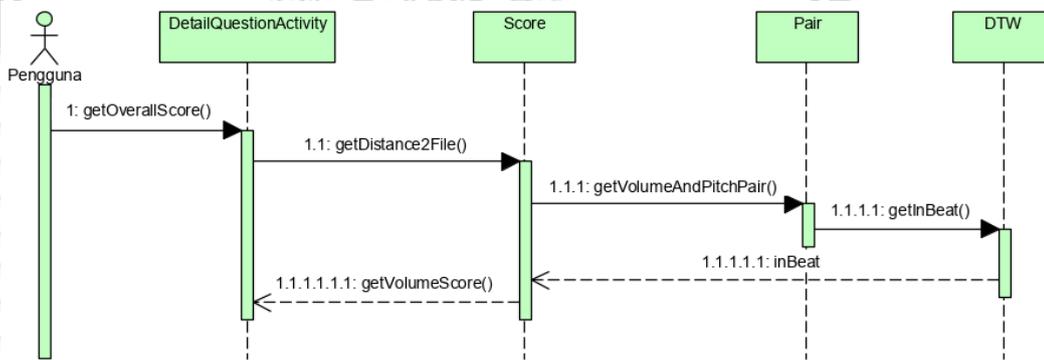
Alur untuk menentukan *pitch scoring* yaitu pengguna harus menekan tombol “lihat skor” untuk memicu pemanggilan fungsi `getOverallScore()`, di dalam fungsi tersebut terdapat beberapa proses, salah satunya yaitu memanggil fungsi `getDistance2File()` di *class Score*, selanjutnya fungsi ini akan memanggil fungsi `getVolumeAndPitchPair()` yaitu algoritma yang dibuat oleh Ganesh Tiwari yang akan mengembalikan nilai *harmonicPair*, kemudian akan dilanjutkan ke proses selanjutnya yaitu memanggil fungsi `getDistance()` yang ada pada *class DTW*. Pada fungsi `getDistance()` ini akan mengembalikan nilai dari variabel *wrappingDistance*, kemudian diproses pada fungsi `getPitchScore()` untuk mencari nilai *score* akhir yang dijadikan sebagai *Pitch Score*. *Sequence diagram pitch scoring* dapat dilihat pada gambar 5.1.



Gambar 5.1 Sequence diagram mencari pitch scoring

5.1.1.2 Sequence Diagram Volume Scoring

Alur untuk menentukan *volume scoring* yaitu pengguna harus menekan tombol “lihat skor” untuk memicu pemanggilan fungsi `getOverallScore()`, di dalam fungsi tersebut terdapat beberapa proses, salah satunya yaitu memanggil fungsi `getDistance2File()` di *class Score*, selanjutnya fungsi ini akan memanggil fungsi `getVolumeAndPitchPair()` yaitu algoritma yang dibuat oleh Ganesh Tiwari yang akan mengembalikan nilai *harmonicPair*, kemudian akan dilanjutkan ke proses selanjutnya yaitu memanggil fungsi `getInBeat()` yang ada pada *class DTW*. Pada fungsi `getInBeat()` ini akan mengembalikan nilai dari variabel *inBeat*, kemudian diproses pada fungsi `getVolumeScore()` untuk mencari nilai *score* akhir yang dijadikan sebagai *Volume Score*. Sequence diagram *volume scoring* dapat dilihat pada gambar 5.2.

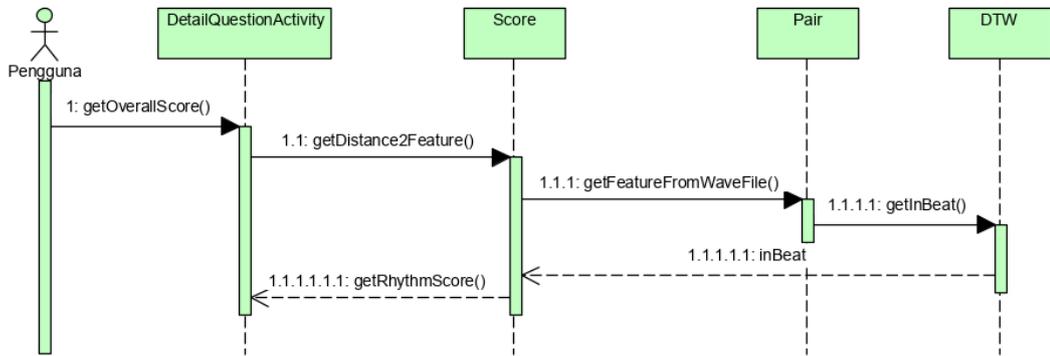


Gambar 5.2 Sequence diagram mencari volume scoring

5.1.1.3 Sequence Diagram Rhythm Scoring

Alur untuk menentukan *volume scoring* yaitu pengguna harus menekan tombol “lihat skor” untuk memicu pemanggilan fungsi `getOverallScore()`, di dalam fungsi tersebut terdapat beberapa proses, salah satunya yaitu memanggil fungsi `getDistance2Feature()` di *class Score*, selanjutnya fungsi ini akan memanggil fungsi `getFeatureFromWaveFile()` yaitu algoritma yang dibuat oleh Ganesh Tiwari yang akan mengembalikan nilai *arrayFeature*, kemudian akan dilanjutkan ke proses selanjutnya yaitu memanggil fungsi `getInBeat()` yang ada

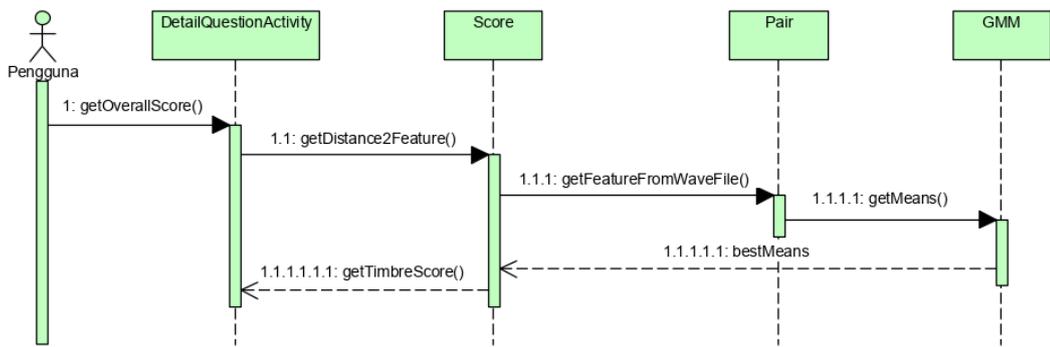
pada class DTW. Pada fungsi `getInBeat()` ini akan mengembalikan nilai dari variabel `inBeat`, kemudian diproses pada fungsi `getRhythmScore()` untuk mencari nilai `score` akhir yang dijadikan sebagai *Rhythm Score*. Sequence diagram *rhythm scoring* dapat dilihat pada gambar 5.3.



Gambar 5.3 Sequence diagram mencari *rhythm scoring*

5.1.1.4 Sequence Diagram Timbre Scoring

Alur untuk menentukan *volume scoring* yaitu pengguna harus menekan tombol “lihat skor” untuk memicu pemanggilan fungsi `getOverallScore()`, di dalam fungsi tersebut terdapat beberapa proses, salah satunya yaitu memanggil fungsi `getDistance2Feature()` di class `Score`, selanjutnya fungsi ini akan memanggil fungsi `getFeatureFromWaveFile()` yaitu algoritma yang dibuat oleh Ganesh Tiwari yang akan mengembalikan nilai `arrayFeature`, kemudian akan dilanjutkan ke proses selanjutnya yaitu memanggil fungsi `getTimbreScore()`. Di dalam fungsi `getTimbreScore()` ini terdapat kalkulasi menghitung rata-rata *feature extraction* dari kedua data audio yaitu pada fungsi `getMeans()` pada class `GMM` yang akan mengembalikan nilai dari variabel `bestMeans` yang akhirnya dijadikan sebagai *Timbre Score*. Sequence diagram *timbre scoring* dapat dilihat pada gambar 5.4.



Gambar 5.4 Sequence diagram mencari *timbre scoring*

5.1.2 Perancangan Class diagram

Class diagram adalah diagram yang menunjukkan seluruh objek pada sistem yang dihubungkan menggunakan relasional antara objek-objek kelas tersebut.

Pada perancangan *class diagram* yang dibuat hanya terdapat pada nama *class* tanpa adanya atribut dan method dari *class* tersebut. Adapun *library* yang digunakan adalah *library musicg* versi 1.4.2.0 yang dikembangkan oleh Google dengan ekstensi berkas (.jar). *Library* tersebut berfungsi untuk mendapatkan skor atau nilai dari *pitch* dan *volume*. Sedangkan untuk mendapatkan nilai dari *rhythm* dan *timbre* menggunakan *library* yang dibuat oleh Ganesh Tiwari. Adapun perbedaan perancangan pada penelitian sebelumnya dengan penelitian ini yaitu terletak pada struktur pengembangan pada program aplikasi, salah satunya pada penelitian ini menggunakan *library* androidx menyesuaikan perkembangan android itu sendiri. Selain itu terdapat perubahan penamaan nama pada *class*, baik file berekstensi (.java) maupun (.xml).

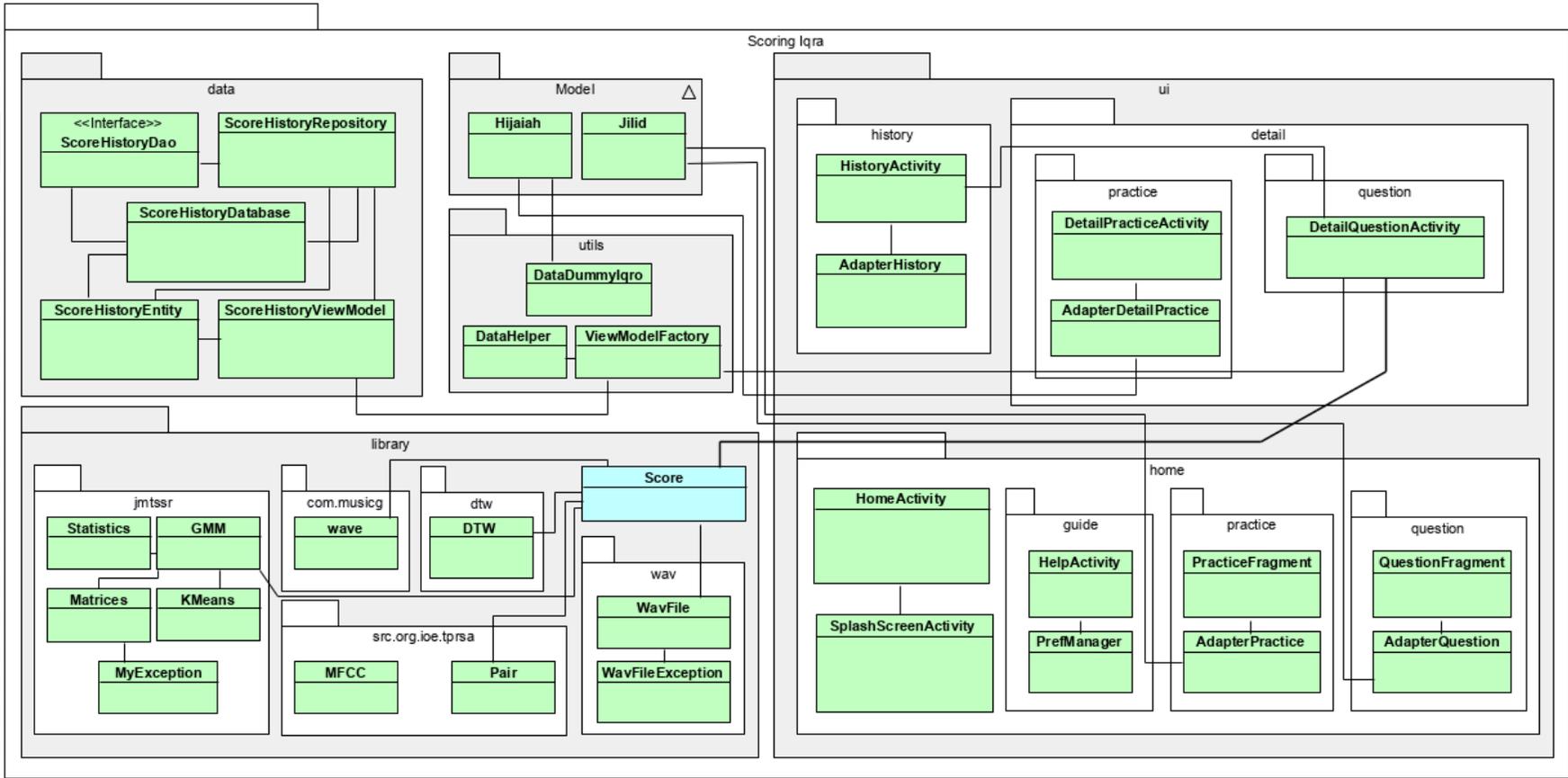
Perancangan *class diagram* secara keseluruhan dapat dilihat pada gambar 5.6. Sedangkan perancangan *class diagram* pada *class Score* akan dijelaskan lebih detail yaitu terdapat atribut dan fungsi yang ada pada *class* tersebut. Perancangan *class diagram Score* dapat dilihat pada gambar 5.5.

```

classDiagram
    class Score {
        +Score()
        +getVolumeScore(dist : double, exp : double) : double
        +getRhythmScore(inBeat : double) : double
        +getPitchScore(dist : double, exp : double) : double
        +getTimbreScore(smp : double[], tmp : double[]) : double
        +getOverallScore(Sv : double, Sp : double, Sr : double, St : double) : double
        +map(x : double, in_min : double, in_max : double, out_min : double, out_max : double) : double
        +getVolumeAndPitchPair(filename : String, arrVol : float[], arrPitc : float[]) : Pair<float[], float[]>
        +isEmptySound(files : String) : boolean
        +getFeatureFromWaveFile(files : String, arrFeat : float[]) : Pair<float[], float[]>
        +getDistance2Feature(file1 : String, file2 : String) : double
        +getDistance2File(file1 : String, file2 : String) : double
    }
    
```

Gambar 5.5 Perancangan *class diagram Score*





Gambar 5.6 Perancangan class diagram Aplikasi Scoring Iqra

5.1.3 Perancangan Algoritma

5.1.3.1 Algoritma Pembobotan Score

Adanya penambahan parameter penilaian baru yaitu *Timbre*, maka terdapat perubahan untuk menentukan *overall score* suara. Perubahan terletak pada rumus menentukan *Overall Score* dan nilai pembobotan dari setiap parameternya. Rumus persamaan dapat dilihat pada persamaan 5.1.

$$\text{Overall score} = W_{pit}.Spit + W_{vol}.Svol + W_{rhy}.Srhy + W_{tim}.Stim \quad (5.1)$$

Spit, *Svol*, *Srhy*, dan *Stim* tersebut berarti skor dari parameter *pitch*, *volume*, *rhythm*, dan *timbre*. Sedangkan *Wpit*, *Wvol*, *Wrhy*, *Wtim* adalah *weight* atau bobot, di mana rumus *overall score* ini disesuaikan pada penelitian (Tsai, et al., 2012). Di mana pada penelitiannya *Wpit*, *Wvol*, dan *Wrhy* sudah ditentukan. Sedangkan *Wtim* tidak dijelaskan, maka dari itu perlu adanya perubahan *weight*. Pada awal perancangan pembobotan sementara menggunakan *Wpit*, *Wvol*, *Wrhy*, dan *Wtim* dengan masing-masing bobot bernilai 0.47, 0.12, 0.39, 0.02 seperti yang diblok kuning pada tabel 5.1. Akan tetapi dari pembobotan tersebut masih belum optimal. Maka perlu adanya perubahan bobot sampai mendapatkan rata-rata *error rate* paling minimum (optimal).

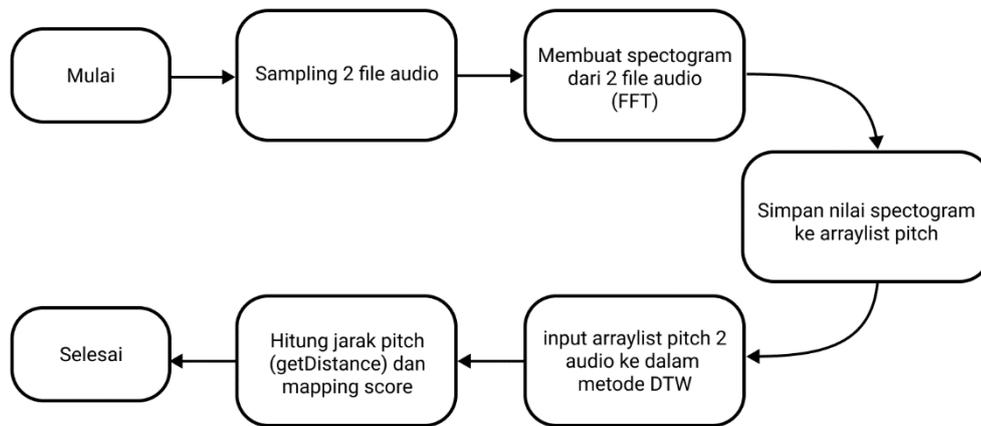
Menurut penelitian (Nuari, et al., 2019) bahwa kriteria, bobot, dan nilai kriteria dapat disesuaikan jika ada perubahan kebijakan, supaya dapat menghasilkan sistem yang baik. Maka penentuan pembobotan pada penelitian ini, akan dilakukan pengujian terhadap 20 bobot skor yang berbeda-beda terhadap 4 parameter *Pitch* (P), *Volume* (V), *Rhythm* (R), dan *Timbre* (T) pada 3 Jilid Iqra. Sehingga dari hasil pengujian tersebut, ditemukan rata-rata *error rate* paling minimum pada Iqra 1 dan Iqra 2 dengan bobot yang sama yaitu *Wpit* sebesar 0.55, *Wvol* sebesar 0.05, *Wrhy* sebesar 0.38, dan *Wtim* sebesar 0.02. Sedangkan pada Iqra 3, bobot tersebut tidak menghasilkan rata-rata *error rate* yang paling minimum. Maka pembobotan pada Iqra 3 akan disamakan dan disesuaikan dengan hasil pembobotan pada Iqra 1 dan Iqra 2. Hasil rata-rata *error rate* paling minimum dari 3 jilid Iqra dapat dilihat pada tabel 5.1 yang diblok hijau.



Tabel 5.1 Pencarian bobot skor dengan minimum error

No.	PEMBOBOTAN PADA IQRA 1						PEMBOBOTAN PADA IQRA 2						PEMBOBOTAN PADA IQRA 3					
	Total	P	V	R	T	MEAN ERROR RATE	Total	P	V	R	T	MEAN ERROR RATE	Total	P	V	R	T	MEAN ERROR RATE
1	1	0.37	0.07	0.51	0.05	8.246656371	1	0.37	0.07	0.51	0.05	8.02853514	1	0.37	0.07	0.51	0.05	5.172012077
2	1	0.41	0.03	0.54	0.02	6.73914665	1	0.41	0.03	0.54	0.02	6.868861133	1	0.41	0.03	0.54	0.02	5.081552129
3	1	0.42	0.12	0.4	0.06	7.29854389	1	0.42	0.12	0.4	0.06	6.535420086	1	0.42	0.12	0.4	0.06	4.096859434
4	1	0.42	0.11	0.41	0.06	7.32422135	1	0.42	0.11	0.41	0.06	6.643570698	1	0.42	0.11	0.41	0.06	4.161192497
5	1	0.42	0.1	0.42	0.06	7.34990927	1	0.42	0.1	0.42	0.06	6.75166628	1	0.42	0.1	0.42	0.06	4.225576405
6	1	0.45	0.08	0.41	0.06	6.769920588	1	0.45	0.08	0.41	0.06	6.315488598	1	0.45	0.08	0.41	0.06	3.97302044
7	1	0.45	0.08	0.43	0.04	6.273561359	1	0.45	0.08	0.43	0.04	5.892862998	1	0.45	0.08	0.43	0.04	4.106359715
8	1	0.45	0.07	0.43	0.05	6.546706823	1	0.45	0.07	0.43	0.05	6.211482017	1	0.45	0.07	0.43	0.05	4.105903872
9	1	0.46	0.05	0.44	0.05	6.38769398	1	0.46	0.05	0.44	0.05	6.210042987	1	0.46	0.05	0.44	0.05	4.171673772
10	1	0.47	0.12	0.39	0.02	5.26811686	1	0.47	0.12	0.39	0.02	4.617173418	1	0.47	0.12	0.39	0.02	3.843316888
11	1	0.37	0.12	0.42	0.09	9.117493896	1	0.37	0.12	0.42	0.09	8.557391704	1	0.37	0.12	0.42	0.09	5.38951533
12	1	0.45	0.05	0.48	0.02	5.84987642	1	0.45	0.05	0.48	0.02	5.78989625	1	0.45	0.05	0.48	0.02	4.578334816
13	1	0.38	0.03	0.55	0.04	7.884329755	1	0.38	0.03	0.55	0.04	7.959732143	1	0.38	0.03	0.55	0.04	5.346365798
14	1	0.36	0.09	0.45	0.1	9.669243036	1	0.36	0.09	0.45	0.1	9.43254047	1	0.36	0.09	0.45	0.1	6.080563127
15	1	0.33	0.09	0.53	0.05	9.063864119	1	0.33	0.09	0.53	0.05	8.850676616	1	0.33	0.09	0.53	0.05	5.759003091
16	1	0.43	0.05	0.48	0.04	6.767222796	1	0.43	0.05	0.48	0.04	6.646759848	1	0.43	0.05	0.48	0.04	4.537352578
17	1	0.4	0.1	0.4	0.1	8.773692782	1	0.4	0.1	0.4	0.1	8.292016316	1	0.4	0.1	0.4	0.1	5.097719021
18	1	0.52	0.08	0.33	0.07	5.572335183	1	0.52	0.08	0.33	0.07	5.029332439	1	0.52	0.08	0.33	0.07	3.209697783
19	1	0.45	0.14	0.31	0.1	7.606831808	1	0.45	0.14	0.31	0.1	6.65086909	1	0.45	0.14	0.31	0.1	3.966249866
20	1	0.55	0.05	0.38	0.02	4.059764156	1	0.55	0.05	0.38	0.02	3.69912862	1	0.55	0.05	0.38	0.02	3.79169744

5.1.3.2 Algoritma *Pitch Based Scoring*



Gambar 5.7 Skema mendapatkan *pitch based scoring*

Gambar 5.7 adalah alur untuk mencari nilai skor *pitch*, diawali dengan memasukkan kedua file audio rekaman pengguna dan ustaz dengan mengambil sampel (sampling) ke dalam fungsi `getDistance2File()` di class `Score`, di dalam fungsi ini akan memanggil fungsi `getVolumeAndPitchPair()` yaitu algoritma yang dibuat oleh Ganesh Tiwari. Fungsi `getVolumeAndPitchPair()` dipanggil setiap file audio, di mana kedua file audio tersebut dikonversi menjadi *spectrogram* menggunakan algoritma *Fast Fourier Transform (FFT)* pada *library musicg* versi 1.4.2.0. Hasil dari *spectrogram* akan dimasukkan ke dalam *arraylist pitch* dan *arraylist volume*. Setelah itu memanggil `getHarmonicProbability()` yang akan mengembalikan *array pitch* dan *array volume* tadi sebagai nilai *harmonicPair* dengan tipe data `Pair` dari *package src.org.ioe.tprsa*.

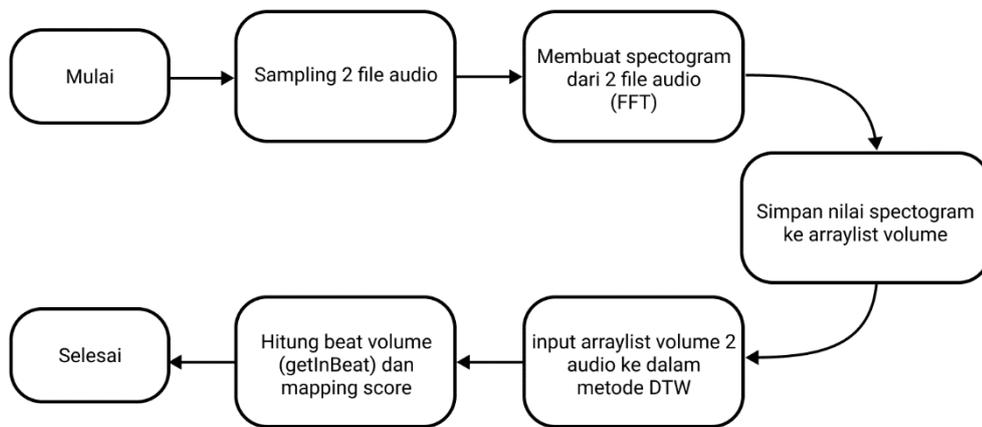
Kemudian lanjut ke proses selanjutnya yaitu memanggil fungsi `getDistance()` yang ada pada class *Dynamic Time Wrapping (DTW)*. Pada kelas DTW ini digunakan untuk menghitung jarak antar 2 variabel yaitu *array pitch* antara berkas audio pengguna dan berkas audio ustaz. Pada fungsi `getDistance()` ini akan mengembalikan nilai dari variabel *wrappingDistance*, kemudian diproses pada fungsi `getPitchScore()` untuk mencari nilai *score* akhir yang dijadikan sebagai *Pitch Score*. Adapun *pseudocode pitch based scoring* dapat dilihat pada tabel 5.2.

Tabel 5.2 Pseudocode pitch based scoring

```

Mulai
    Panggil method getDistance2File
    Instansiasi objek DTW
    Instansiasi object Score
    Panggil method pitch based scoring (getDistance)
Selesai
    
```

5.1.3.3 Algoritma Volume Based Scoring



Gambar 5.8 Skema mendapatkan volume based scoring

Gambar 5.8 adalah alur untuk mencari nilai skor *volume*, diawali dengan memasukkan kedua file audio rekaman pengguna dan ustaz dengan mengambil sampel (sampling) ke dalam fungsi `getDistance2File()` di *class Score*, di dalam fungsi ini akan memanggil fungsi `getVolumeAndPitchPair()` yaitu algoritma yang dibuat oleh Ganesh Tiwari. Fungsi `getVolumeAndPitchPair()` dipanggil setiap file audio, di mana kedua file audio tersebut dikonversi menjadi *spectrogram* menggunakan algoritma *Fast Fourier Transform (FFT)* pada *library musicg* versi 1.4.2.0. Hasil dari *spectrogram* akan dimasukkan ke dalam *arraylist pitch* dan *arraylist volume*. Setelah itu memanggil `getHarmonicProbability()` yang akan mengembalikan *array pitch* dan *array volume* tadi sebagai nilai *harmonicPair* dengan tipe data `Pair` dari *package src.org.ioe.tprsa*.

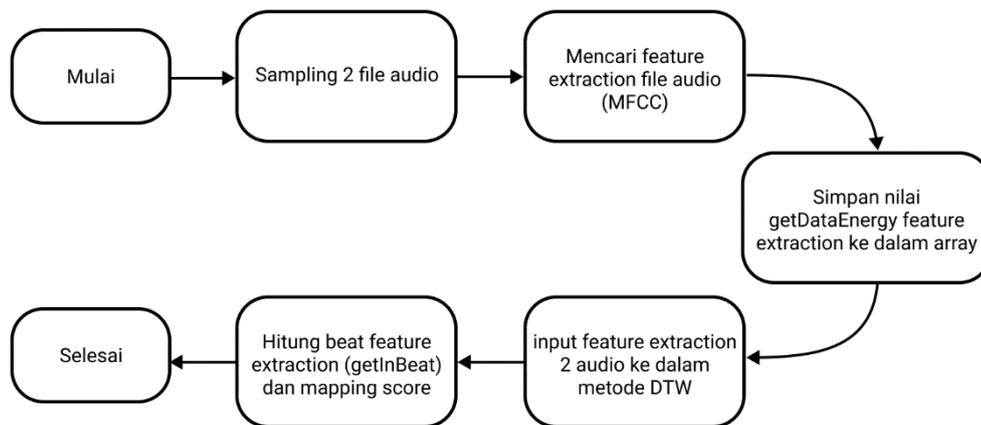
Kemudian lanjut ke proses selanjutnya yaitu memanggil fungsi `getInBeat()` yang ada pada *class Dynamic Time Wrapping (DTW)*. Pada kelas DTW ini digunakan untuk menghitung jarak antar 2 variabel yaitu *array volume* antara berkas audio pengguna dan berkas audio ustaz. Pada fungsi `getInBeat()` ini akan mengembalikan nilai dari variabel *inBeat*, kemudian diproses pada fungsi `getVolumeScore()` untuk mencari nilai *score* akhir yang dijadikan sebagai *Volume Score*. Adapun *pseudocode volume based scoring* dapat dilihat pada tabel 5.3.

Tabel 5.3 Pseudocode volume based scoring

```

Mulai
    Panggil method getDistance2File
    Instansiasi objek DTW
    Instansiasi object Score
    Panggil method volume based scoring (getInBeat)
Selesai
    
```

5.1.3.4 Algoritma *Rhythm Based Scoring*



Gambar 5.9 Skema mendapatkan *rhythm based scoring*

Gambar 5.9 adalah alur untuk mencari nilai skor *volume*, diawali dengan memasukkan kedua file audio rekaman pengguna dan ustaz dengan mengambil sampel (sampling) ke dalam fungsi `getDistance2Feature()` di *class Score*, di dalam fungsi ini akan memanggil fungsi `getFeatureFromWaveFile()` yaitu algoritma yang dibuat oleh Ganesh Tiwari. Fungsi `getFeatureFromWaveFile()` dipanggil setiap file audio, di mana kedua file audio akan dicari *feature extraction* dengan algoritma *Mel-Frequency Cepstral Coefficient* (MFCC), sehingga didapatkan nilai dari *delta energy* dalam bentuk *array*. Hasil dari *delta energy* tersebut akan dijadikan nilai kembalian sebagai nilai *featurePair* dengan tipe data *Pair* dari *package src.org.ioe.tprsa*.

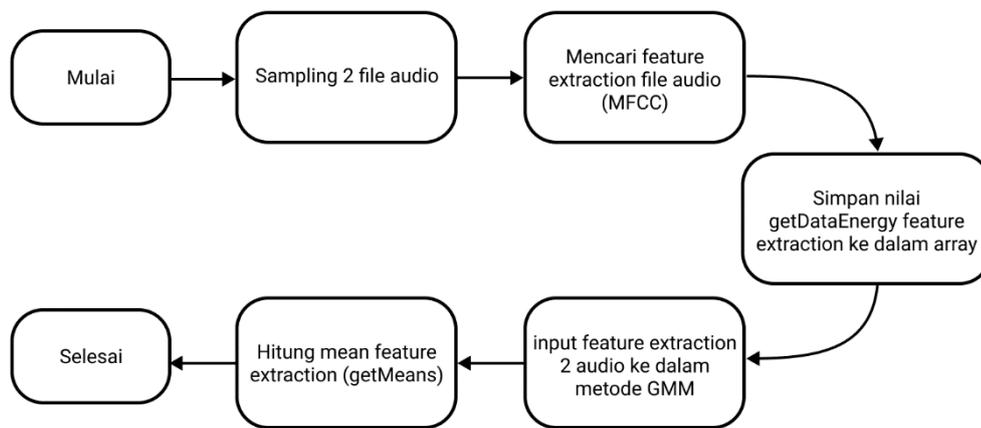
Kemudian lanjut ke proses selanjutnya yaitu memanggil fungsi `getInBeat()` yang ada pada *class Dynamic Time Wrapping* (DTW). Pada kelas DTW ini digunakan untuk menghitung jarak antar 2 variabel yaitu *array feature* antara berkas audio pengguna dan berkas audio ustaz. Pada fungsi `getInBeat()` ini akan mengembalikan nilai dari variabel *inBeat*, kemudian diproses pada fungsi `getRhythmScore()` untuk mencari nilai *score* akhir yang dijadikan sebagai *Rhythm Score*. Adapun *pseudocode rhythm based scoring* dapat dilihat pada tabel 5.4.

Tabel 5.4 Pseudocode rhythm based scoring

```

Mulai
    Panggil method getDistance2Feature
    Instansiasi objek DTW
    Instansiasi object Score
    Panggil method rhythm based scoring (getInBeat)
Selesai
    
```

5.1.3.5 Algoritma *Timbre Based Scoring*



Gambar 5.10 Skema mendapatkan *timbre based scoring*

Gambar 5.10 adalah alur untuk mencari nilai skor *volume*, diawali dengan memasukkan kedua file audio rekaman pengguna dan ustaz dengan mengambil sampel (sampling) ke dalam fungsi `getDistance2Feature()` di *class Score*, di dalam fungsi ini akan memanggil fungsi `getFeatureFromWaveFile()` yaitu algoritma yang dibuat oleh Ganesh Tiwari. Fungsi `getFeatureFromWaveFile()` dipanggil setiap file audio, di mana kedua file audio akan dicari *feature extraction* dengan algoritma *Mel-Frequency Cepstral Coefficient* (MFCC), sehingga didapatkan nilai dari *delta energy* dalam bentuk *array*. Hasil dari *delta energy* tersebut akan dijadikan nilai kembalian sebagai nilai *featurePair* dengan tipe data *Pair* dari *package src.org.ioe.tprsa*.

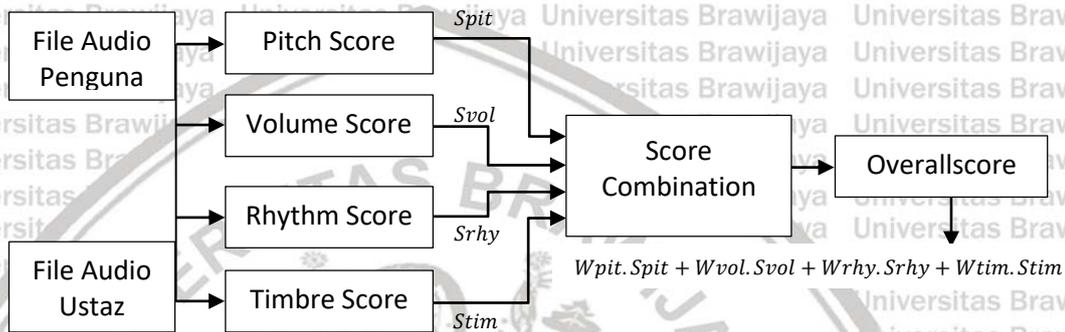
Kemudian lanjut ke proses selanjutnya yaitu memanggil fungsi `getMeans()` yang ada pada *class Gaussian Mixture Model* (GMM). Pada kelas GMM ini digunakan untuk menghitung jarak antar 2 variabel yaitu *array feature* antara berkas audio pengguna dan berkas audio ustaz. Pada fungsi `getMeans()` ini akan mengembalikan nilai dari variabel *bestMeans* atau nilai rata-rata dari kedua data fitur ekstraksi audio rekaman, yang akhirnya dijadikan sebagai *Timbre Score*. Adapun *pseudocode timbre based scoring* dapat dilihat pada tabel 5.5.

Tabel 5.5 Pseudocode timbre based scoring

```

Mulai
    Panggil method getDistance2Feature
    Instansiasi objek GMM
    Instansiasi object Score
    Panggil method timbre based scoring (dataFeature2Audio)
Selesai
    
```

5.1.3.6 Algoritma Overall Score



Gambar 5.11 Skema perubahan untuk memperoleh overall score

Pada gambar 5.11 adalah skema dalam mendapatkan *overall score*, yang terdapat perubahan skema dari penelitian (Tsai, et al., 2012) yaitu menambahkan *timbre* sebagai parameter baru dalam menghitung *overall score*. Di mana rumus *overall score* ini akan dijalankan terakhir setelah mendapatkan semua nilai skor dari 4 parameter. Parameter tersebut adalah *pitch*, *volume*, *rhythm*, dan *timbre*. Untuk mendapatkan nilai dari *overall score* memerlukan pembobotan skor yang telah ditetapkan yaitu bobot skor *pitch* sebesar 0.55, bobot skor *volume* sebesar 0.05, bobot skor *rhythm* sebesar 0.38, dan bobot skor *timbre* sebesar 0.02. Dari bobot tersebut akan dikalikan terhadap nilai skor dari tiap parameternya dan kemudian dijumlahkan seperti persamaan 5.1 pada algoritma pembobotan *score*. Adapun *pseudocode* mendapatkan *overall score* dapat dilihat pada tabel 5.6.

Tabel 5.6 Pseudocode overall score

```

Mulai
    Merekam suara pengguna
    Panggil method pitch based scoring
    Panggil method volume based scoring
    Panggil method rhythm based scoring
    Panggil method timbre based scoring
    Panggil method overall score
Selesai
    
```

5.1.4 Perancangan Basis Data

Pada penelitian ini akan membahas perancangan basis data yang digunakan untuk penyimpanan data dalam membangun Aplikasi *Scoring Iqra*. Adapun Media yang digunakan untuk menyimpan datanya yaitu internal memori. Penyimpanan data secara internal ini hanya digunakan untuk menyimpan data *history* dari skor penilaian dari pengerjaan soal bacaan huruf hijaiyah. Untuk mempermudah dalam pembuatan basis data secara internal maka pada penelitian ini perlu menggunakan bantuan pihak ketiga yaitu *Room*. Seperti yang dijelaskan pada bab sebelumnya, pihak ketiga ini kita membutuhkan *Entitas*, *Repository*, dan *Data Access Object (DAO)*.

Pada *repository* berisi beberapa fungsi untuk mengatur data pada tabel *database*. *Repository* ini bertugas untuk meneruskan *request* ke DAO. Adapun fungsi yang dibutuhkan pada DAO Aplikasi ini yaitu *insert* sebagai input ke database, *query* untuk mengambil data dari database, dan *delete* untuk menghapus data dari database.

Syarat yang harus diterapkan pada *library Room* ini yaitu mengharuskan Aplikasi memiliki satu *instance* ke *database room* yang dapat digunakan untuk mengakses beberapa tabel (*entitas*). Sehingga untuk pengaturan pembuatan database perlu meng-*extends RoomDatabase* dan dipisah dengan pembuatan tabelnya, dimana nama database akan di definisikan di *class* ini. Sedangkan pada *entitas*-nya kita perlu mendefinisikan skema untuk membuat tabel berupa nama tabel, nama atribut, jenis data, dan atribut yang menjadi *primary key*. Adapun skema *entitas* basis data Aplikasi *Scoring Iqra* dapat dilihat pada Tabel 5.7.

Tabel 5.7 Skema Perancangan Basis Data

History (<i>database name</i>)		
score_history (<i>table name</i>)		
id	int	<i>Primary key (auto generated)</i>
jilid	String	-
date	String	-
file_name	String	-
pitch	String	-
volume	String	-
rhythm	String	-
timbre	String	-
overall_score	String	-

5.1.5 Perancangan CRC Cards

Pada gambar 5.12 adalah diagram CRC Cards pada class Score. Pada Perancangan ini terdapat 5 bagian yaitu *Class name*, *Description*, *Responsibilities*, *Name*, dan *Collaborator*. Pada gambar ini dijelaskan bahwa terdapat 3 *library* yang digunakan sebagai kolaborasi untuk menentukan *overallscore* yaitu musicg, tprsa, dan jmtssr. Adapun musicg bertujuan untuk mengatur *spectrogram* dari Audio. *Library* tprsa digunakan untuk mencari *feature* dari Audio. Sedangkan jmtssr adalah *library open source* GMM (*Gaussian Mixture Models*) untuk mencari nilai kesamaan 2 Audio berdasarkan warna suara (*timbre*). Dimana pengambilan sampling file Audio dilakukan pada *DetailQuestionActivity* yaitu halaman untuk menjawab soal bacaan huruf hijaiyah.

Score	
Description: menghitung skor pitch, volume, rhythm, dan timbre untuk dijadikan parameter pencarian overallscore	
Responsibilities:	
Name	Collaborator
sampling file Audio	DetailQuestionActivity
menggunakan library musicg	musicg
menggunakan class ganesh tiwari	tprsa
menggunakan library GMM speaker recognition	jmtssr
menampilkan overallscore	DetailQuestionActivity

Gambar 5.12 Perancangan CRC Cards class Score

5.1.6 Perancangan User Interface (UI)

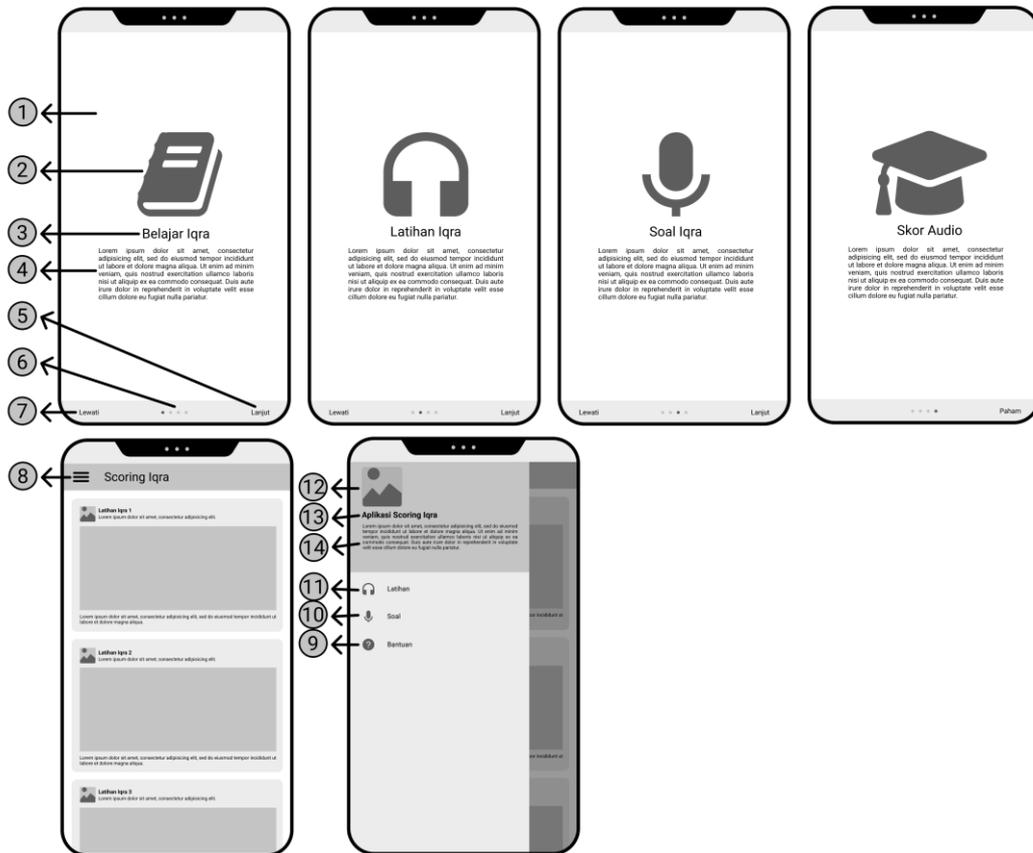
Pada subbab perancangan *User Interface* (antarmuka) ini akan dibahas mengenai komponen-komponen yang digunakan beserta alur kerja atau jalannya aplikasi scoring iqra dalam bentuk *screen flow*. *Screen flow* adalah menampilkan desain aplikasi secara berurutan. Pada subbab ini terdapat 2 iterasi perancangan perubahan dari desain awal yang ada pada penelitian sebelumnya. Untuk mempermudah penjelasan dalam setiap komponennya, maka diberikan penomoran pada halaman antarmuka yang dibahas.

5.1.6.1 Perancangan UI Iterasi Pertama

Perubahan pada iterasi pertama ini tidak begitu signifikan dengan tampilan yang ada pada aplikasi penelitian sebelumnya, perubahannya yaitu mengubah desain yang ada pada halaman detail soal, di mana semua sudut tombol dibentuk oval, dan penambahan *topbar* pada halaman detail soal dan detail latihan. Pada masing-masing *topbar* tersebut memiliki tombol kembali ke menu sebelumnya dan judul sebagai penanda jilid sesuai yang dipilih pengguna.

a. Perancangan UI menu bantuan iterasi pertama

Pada perancangan menu bantuan bertujuan untuk menjelaskan kepada pengguna bagaimana cara penggunaan aplikasi dan memberikan informasi menu apa saja yang ada pada aplikasi *scoring iqra* dalam bentuk *onboarding screen*. Perancangan UI menu bantuan dapat dilihat pada gambar 5.13.



Gambar 5.13 Perancangan UI menu bantuan iterasi pertama

Pada tabel 5.8 akan menjelaskan penomoran bagian-bagian dari gambar 5.13 secara rinci mengenai komponen-komponen beserta keterangannya.

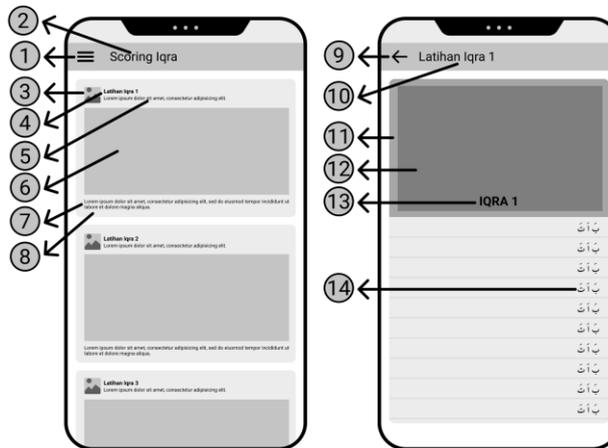
Tabel 5.8 Komponen pada UI menu bantuan iterasi pertama

No.	Nama Komponen	Keterangan
1.	ViewPager	Pertama kali <i>install</i> aplikasi <i>scoring iqra</i> , sistem akan menampilkan <i>Onboarding Screen</i> , tampilan tersebut hanya digunakan sebagai bantuan berupa tata cara penggunaan dan menunjukkan menu apa saja yang ada pada aplikasi <i>scoring iqra</i> . Fungsi dari komponen <i>ViewPager</i> yaitu jika pengguna ingin berpindah pada

		<i>slide</i> lain pada tampilan <i>Onboarding Screen</i> , maka dapat dipicu dengan <i>swipe</i> ke kanan/kiri pada layar.
2.	<i>ImageView</i>	Pada komponen ini digunakan sebagai icon dari tiap-tiap <i>slide</i> pada tampilan <i>Onboarding Screen</i> .
3.	<i>TextView</i>	Penggunaan <i>TextView</i> ini bertujuan sebagai judul dari tiap-tiap <i>slide</i> pada tampilan <i>Onboarding Screen</i> .
4.	<i>TextView</i>	Penggunaan <i>TextView</i> yang kedua ini bertujuan sebagai subjudul yang menjelaskan kegunaan menu dari tiap-tiap <i>slide</i> pada tampilan <i>Onboarding Screen</i> .
5.	<i>Button</i>	Komponen ini berada pada <i>layout activity_help</i> yang berfungsi untuk memicu perubahan <i>slide</i> selanjutnya pada tampilan <i>Onboarding Screen</i> .
6.	<i>LinearLayout</i>	Berfungsi sebagai indikator posisi <i>slide</i> pada tampilan <i>Onboarding Screen</i> .
7.	<i>Button</i>	Komponen ini berada pada <i>layout activity_help</i> yang berfungsi untuk melewati tampilan <i>Onboarding Screen</i> .
8.	<i>Hamburger button</i>	Komponen ini berfungsi sebagai pemicu munculnya <i>navigation drawer header</i> .
9.	<i>Menu Bantuan</i>	Salah satu dari komponen <i>menu navigation drawer header</i> , yang berfungsi sebagai alternatif ketika pengguna ingin membuka kembali menu bantuan.
10.	<i>Menu Soal</i>	Salah satu dari komponen <i>menu navigation drawer header</i> , yang berfungsi sebagai alternatif ketika pengguna ingin membuka fragment menu soal.
11.	<i>Menu Latihan</i>	Salah satu dari komponen <i>menu navigation drawer header</i> , yang berfungsi sebagai alternatif ketika pengguna ingin membuka fragment menu latihan.
12.	<i>ImageView</i>	Pada bagian atas <i>navigation drawer header</i> terdapat 3 komponen. Komponen pertama ini berfungsi sebagai Logo aplikasi <i>scoring iqra</i> .
13.	<i>TextView</i>	Komponen ini digunakan sebagai nama dari aplikasi.
14.	<i>TextView</i>	Komponen terakhir yaitu berfungsi sebagai penjelas singkat aplikasi <i>scoring iqra</i> secara umum.

b. Perancangan UI menu latihan iterasi pertama

Pada perancangan menu latihan bertujuan untuk memberikan pelatihan cara membaca huruf hijaiyah berupa suara dari ustaz yang dapat didengarkan oleh pengguna berulang-ulang. Perancangan UI menu latihan dapat dilihat pada gambar 5.14.



Gambar 5.14 Perancangan UI menu latihan iterasi pertama

Pada tabel 5.9 akan menjelaskan penomoran bagian-bagian dari gambar 5.14 secara rinci mengenai komponen-komponen beserta keterangannya.

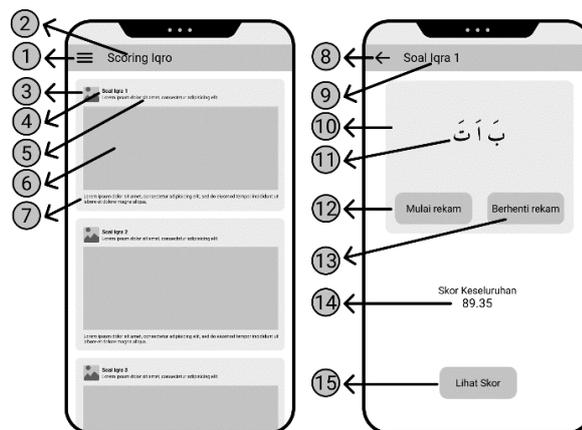
Tabel 5.9 Komponen pada UI menu latihan iterasi pertama

No.	Nama Komponen	Keterangan
1.	<i>Hamburger button</i>	Pada topbar aplikasi terdapat 2 komponen, komponen pertama ini berfungsi sebagai pemicu munculnya <i>navigation drawer header</i> .
2.	<i>TextView</i>	Komponen kedua ini sebagai nama aplikasi.
3.	<i>ImageView</i>	Pada <i>RecyclerView</i> tiap item terdiri beberapa komponen, di mana tiap item dibungkus dengan <i>CardView</i> . Komponen ini digunakan sebagai jilid iqra. Berupa angka dalam Bahasa Arab.
4.	<i>TextView</i>	Pada komponen ini sebagai judul dari latihan setiap jilid iqra.
5.	<i>TextView</i>	Pada komponen ini sebagai subjudul dari latihan setiap jilid iqra.
6.	<i>ImageView</i>	Komponen ini digunakan sebagai <i>icon</i> latihan iqra, yaitu berupa gambar <i>headphone</i> sesuai jilidnya. Pemilihan gambar <i>headphone</i> dikarenakan dalam latihan cara baca huruf hijaiyah, pengguna dapat

		menggunakan <i>headphone</i> supaya dapat mendengarkan mahraj lebih jelas.
7.	<i>TextView</i>	Komponen ini digunakan sebagai penjelas secara umum dari tiap-tiap latihan jilid iqra.
8.	<i>CardView</i>	Pada setiap <i>list</i> jilid iqra ini, semua komponen dibungkus oleh <i>CardView</i> . Tujuannya yaitu supaya item terlihat timbul dan memiliki efek bayangan.
9.	<i>ImageView</i>	Komponen yang dijadikan sebagai tombol kembali ke halaman sebelumnya.
10.	<i>TextView</i>	Digunakan sebagai judul yang disamakan dengan komponen judul <i>list</i> latihan sebelumnya.
11.	<i>View</i>	Digunakan untuk memberikan <i>opacity</i> (efek transparansi) pada komponen yang dibawahnya.
12.	<i>ImageView</i>	Digunakan untuk <i>icon</i> latihan iqra, yang diambil dari komponen gambar pada <i>list</i> latihan sebelumnya.
13.	<i>TextView</i>	Sebagai judul dari jilid yang dipilih oleh pengguna.
14.	<i>ListView</i>	Pada <i>ListView</i> ini hanya menampilkan daftar huruf hijaiyah.

c. Perancangan UI menu Soal iterasi pertama

Pada perancangan menu soal bertujuan untuk menguji kemampuan cara membaca huruf hijaiyah dari rekaman pengguna yang nantinya pengguna dapat memperoleh skor dengan membandingkan seberapa mirip dengan suara referensi dari ustaz. Perancangan UI menu soal dapat dilihat pada gambar 5.15.



Gambar 5.15 Perancangan UI menu soal iterasi pertama

Pada tabel 5.10 akan menjelaskan penomoran bagian-bagian dari gambar 5.15 secara rinci mengenai komponen-komponen beserta keterangannya.

Tabel 5.10 Komponen UI pada menu soal iterasi pertama

No.	Nama Komponen	Keterangan
1.	<i>Hamburger button</i>	Pada topbar aplikasi terdapat 2 komponen, komponen pertama ini berfungsi sebagai pemicu munculnya <i>navigation drawer header</i> .
2.	<i>TextView</i>	Komponen kedua ini sebagai nama aplikasi.
3.	<i>ImageView</i>	Pada <i>RecyclerView</i> tiap item terdiri beberapa komponen, di mana tiap item dibungkus dengan <i>CardView</i> . Komponen ini digunakan sebagai jilid iqra. Berupa angka dalam Bahasa Arab.
4.	<i>TextView</i>	Pada komponen ini sebagai judul dari soal pada setiap jilid iqra.
5.	<i>TextView</i>	Pada komponen ini sebagai subjudul dari soal pada setiap jilid iqra.
6.	<i>ImageView</i>	Komponen ini digunakan sebagai <i>icon</i> soal iqra, yaitu berupa gambar <i>microphone</i> sesuai jilidnya. Pemilihan gambar ini dikarenakan dalam menjawab soal pengguna harus merekam suaranya terlebih dahulu, yang identik merekam dengan <i>microphone</i> .
7.	<i>TextView</i>	Komponen ini digunakan sebagai penjelas secara umum dari tiap-tiap soal jilid iqra.
8.	<i>ImageView</i>	Komponen yang dijadikan sebagai tombol kembali ke halaman sebelumnya.
9.	<i>TextView</i>	Digunakan sebagai judul yang disamakan dengan komponen judul <i>list</i> soal sebelumnya.
10.	<i>CardView</i>	Dalam konten utama pada halaman untuk menjawab soal ini, beberapa komponen dibungkus menggunakan <i>CardView</i> . Selain dijadikan <i>background</i> juga supaya terlihat timbul dan bisa memberi efek bayangan.
11.	<i>TextView</i>	Komponen ini digunakan untuk menampilkan soal huruf hijaiah yang harus dijawab pengguna.
12.	<i>Button</i>	Digunakan sebagai tombol untuk mulai merekam suara bacaan dari soal huruf hijaiah.
13.	<i>Button</i>	Digunakan sebagai tombol untuk berhenti merekam suara bacaan dari soal huruf hijaiah

14.	<i>TextView</i>	Pada komponen ini berfungsi untuk menampilkan skor yang diperoleh oleh pengguna.
15.	<i>Button</i>	Komponen ini dijadikan pemicu untuk pemanggilan fungsi kalkulasi perhitungan skor keseluruhan.

5.1.6.2 Perancangan UI Iterasi Kedua

Perubahan pada iterasi kedua ini terdapat perubahan yaitu mengubah desain yang ada pada halaman menjawab soal, di mana semua sudut tombol dibentuk oval, dan mengubah *button* untuk merekam audio. Perubahan kedua yaitu pada halaman latihan, yaitu mengubah *List* huruf hijaiyah dari *ListView* menjadi *RecyclerView* supaya pada setiap item huruf hijaiyah dapat di kustom. Alasan lain yaitu supaya pada aplikasi ini memiliki tema selaras yaitu berbentuk oval / lingkaran, dan terlihat interaktif terhadap pengguna. Selain perubahan tampilan halaman, terdapat perubahan warna Aplikasi dari biru ke hijau, dikarenakan warna hijau adalah warna yang identik dengan islam sekaligus warna kesukaan Rasulullah Muhammad SAW (Saefullah, 2017).

Selain itu terdapat tambahan halaman *Splashscreen* pada awal pembukaan aplikasi, yang berisi logo dan *copyright* tahun pembuatan aplikasi. Kemudian terdapat tambahan halaman riwayat skor dari pengerjaan soal huruf hijaiyah. Alasan menambah halaman baru ini yaitu sebagai pengembangan dari aplikasi pada penelitian sebelumnya supaya dapat lebih interaktif dengan pengguna.

a. Perancangan UI menu bantuan iterasi kedua

Pada perancangan menu bantuan bertujuan untuk menjelaskan kepada pengguna bagaimana cara penggunaan aplikasi dan memberikan informasi menu apa saja yang ada pada aplikasi *scoring iqra* dalam bentuk *onboarding screen*. Pada perancangan menu bantuan iterasi kedua ini masih tetap seperti iterasi pertama, hanya saja terdapat tambahan halaman *Splashscreen*. Kegunaan *Splashscreen* ini yaitu untuk tampilan awal setiap pengguna membuka aplikasi *scoring iqra*. Selain itu terdapat perubahan bentuk logo aplikasi dari kotak ke bentuk lingkaran. Perancangan UI menu bantuan dapat dilihat pada gambar 5.16.



Gambar 5.16 Perancangan UI menu bantuan iterasi kedua

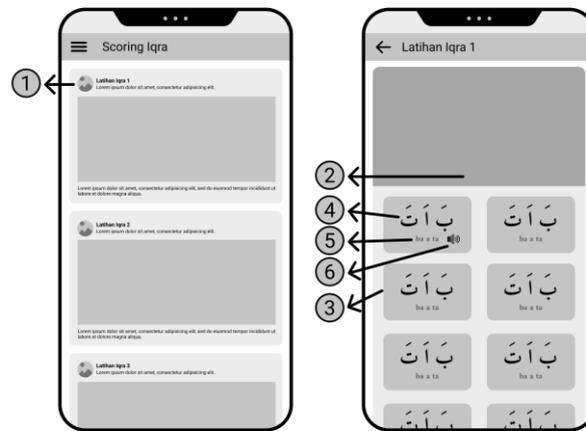
Pada tabel 5.11 akan menjelaskan penomoran bagian-bagian dari gambar 5.16 secara rinci mengenai komponen-komponen beserta keterangannya.

Tabel 5.11 Komponen UI pada menu bantuan iterasi kedua

No.	Nama Komponen	Keterangan
1.	<i>Lottie (.json)</i>	Komponen pertama pada <i>Splashscreen</i> berupa animasi dari <i>library lottie</i> .
2.	<i>TextView</i>	Komponen kedua pada <i>Splashscreen</i> digunakan sebagai nama aplikasi, komponen ini juga dibuat bergerak supaya terlihat menarik pengguna.
3.	<i>TextView</i>	Komponen ketiga pada <i>Splashscreen</i> digunakan sebagai tanda <i>copyright</i> pembuatan aplikasi pada tahun 2021.
4.	<i>CircleImageView</i>	Perubahan bentuk logo aplikasi dari kotak ke lingkaran, bertujuan untuk mempertahankan tema aplikasi yaitu oval.
5.	<i>CircleImageView</i>	Perubahan bentuk logo aplikasi dari kotak ke lingkaran, bertujuan untuk mempertahankan tema aplikasi yaitu oval.

b. Perancangan UI menu latihan iterasi kedua

Pada perancangan menu latihan bertujuan untuk memberikan pelatihan cara membaca huruf hijaiyah berupa suara dari ustaz yang dapat didengarkan oleh pengguna berulang-ulang. Pada perancangan menu latihan pada iterasi kedua ini masih sama seperti iterasi pertama, hanya saja terdapat perubahan pada halaman detail latihan yaitu mengubah *List* huruf hijaiyah dari *ListView* menjadi *RecyclerView* supaya pada setiap item huruf hijaiyah dapat di kustom, maksudnya pada setiap *list* huruf hijaiyah terdapat beberapa komponen dan dapat disesuaikan desainnya. Perancangan UI menu latihan dapat dilihat pada gambar 5.13.



Gambar 5.17 Perancangan UI menu latihan iterasi kedua

Pada tabel 5.12 akan menjelaskan penomoran bagian-bagian dari gambar 5.13 secara rinci mengenai komponen-komponen beserta keterangannya.

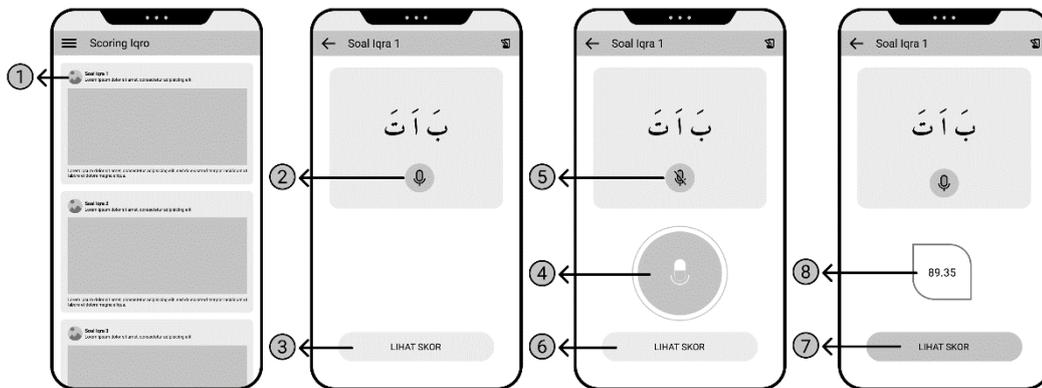
Tabel 5.12 Komponen UI pada menu latihan iterasi kedua

No.	Nama Komponen	Keterangan
1.	<i>CircleImageView</i>	Bentuk <i>icon</i> jilid iqra ini diubah dari berbentuk kotak menjadi bentuk lingkaran.
2.	<i>ImageView</i>	Pada <i>icon</i> latihan iqra hanya menampilkan gambar saja, dan menghapus judul jilid pada iterasi pertama dikarenakan terdapat duplikasi judul dengan judul di bagian topbar.
3.	<i>RecyclerView</i>	Pada setiap huruf hijaiyah yang ditampilkan dibungkus menggunakan <i>CardView</i> . Hanya saja disini mengubah <i>List</i> yang dipakai dari <i>ListView</i> menjadi <i>RecyclerView</i> supaya 3 komponen yang terdapat didalamnya dapat dikustom letaknya.
4.	<i>TextView</i>	Komponen pertama yaitu digunakan sebagai huruf hijaiyah, dengan merubah ukuran <i>font</i> menjadi lebih

		besar. Hal tersebut bertujuan untuk memfokuskan pengguna pada huruf hijaiyah sembari mendengarkan cara baca huruf tersebut.
5.	<i>TextView</i>	Komponen kedua yaitu digunakan sebagai tanda baca dari huruf hijaiyah dalam tulisan latin, yang ditulis sesuai aturan pada (Wikipedia, 2020). Penambahan huruf latin ini bertujuan jika pengguna tidak jelas mendengarkan suara ustaz, maka dapat membaca huruf latin tersebut.
6.	<i>ImageView</i>	Komponen terakhir yaitu gambar <i>speaker</i> sebagai tanda ketika pengguna klik salah satu item <i>RecyclerView</i> maka akan muncul <i>icon</i> ini. Hal tersebut bertujuan supaya pengguna dapat melihat huruf mana yang berbunyi.

c. Perancangan UI menu soal iterasi kedua

Pada perancangan menu soal bertujuan untuk menguji kemampuan cara membaca huruf hijaiyah dari rekaman pengguna yang nantinya dapat memperoleh skor yang dibandingkan seberapa mirip dengan suara ustaz. Perubahan pada iterasi kedua ini terletak pada *button* untuk merekam suara dan terdapat penambahan animasi ketika pengguna sedang merekam. Perancangan UI menu soal dapat dilihat pada gambar 5.18.



Gambar 5.18 Perancangan UI menu soal iterasi kedua

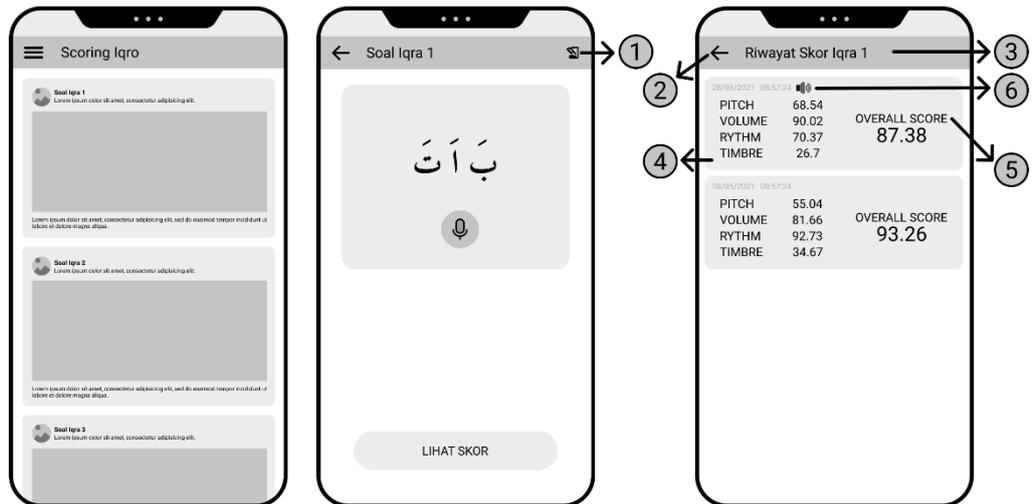
Pada tabel 5.13 akan menjelaskan penomoran bagian-bagian dari gambar 5.14 secara rinci mengenai komponen-komponen beserta keterangannya.

Tabel 5.13 Komponen UI pada menu soal iterasi kedua

No.	Nama Komponen	Keterangan
1.	<i>CircleImageView</i>	Bentuk <i>icon</i> jilid iqra ini diubah dari berbentuk kotak menjadi bentuk lingkaran.
2.	<i>ImageView</i>	Pada iterasi sebelumnya tombol untuk merekam suara pengguna yaitu menggunakan <i>button</i> dan pada iterasi kedua ini diubah menjadi <i>ImageView</i> supaya dapat menampilkan secara simpel dengan <i>icon microphone on</i> .
3.	<i>Button</i>	Pada iterasi ini tombol untuk memicu kalkulasi skor tidak bisa diklik selama pengguna belum merekam jawabannya. Hal tersebut dikarenakan tombol di beri <i>state enable = false</i> .
4.	<i>Lottie .json</i>	Ketika tombol untuk merekam (nomer 2) ditekan maka akan memunculkan animasi <i>microphone</i> sebagai tanda bahwa pengguna sedang merekam.
5.	<i>ImageView</i>	Ketika tombol nomer 2 ditekan maka <i>visibility</i> dari sistem akan bekerja. Di mana tombol untuk merekam (nomer 2) akan di sembunyikan dan menampilkan tombol dengan <i>icon microphone off</i> sebagai tombol untuk mengakhiri rekaman suara.
6.	<i>Button</i>	Tombol lihat skor tersebut masih <i>enable = false</i> , dikarenakan proses merekam belum selesai.
7.	<i>Button</i>	Ketika tombol nomer 5 ditekan, maka animasi (nomer 4) dan tombol <i>microphone off</i> (nomer 5) akan di sembunyikan. Kemudian menampilkan tombol <i>microphone on</i> kembali. Hal tersebut menandakan bahwa proses rekam suara pengguna telah selesai dan berhasil disimpan di penyimpanan lokal dari perangkat pengguna. Maka dari itu tombol lihat skor akan mengubah <i>state</i> menjadi <i>enable = true</i> , sehingga dapat ditekan untuk melihat skor.
8.	<i>TextView</i>	Perubahan tampilan nilai skor disini yaitu dari sebelumnya tidak ada <i>background</i> , dan pada iterasi kedua ini terdapat penambahan <i>background</i> seperti belah ketupat. Dimana terdapat 3 warna <i>background</i> untuk membagi kategori nilai dari warna merah (nilai < 30), warna kuning (nilai antara 30 - 70), dan warna hijau (nilai > 70).

d. Perancangan UI halaman riwayat skor

Pada perancangan halaman riwayat skor ini bertujuan untuk memenuhi kebutuhan sesuai *user stories*. Alasan lainnya supaya pengguna dapat mengevaluasi skor hasil latihan dari mengerjakan soal bacaan huruf hijaiyah. Perancangan UI halaman riwayat dapat dilihat pada gambar 5.19.



Gambar 5.19 Perancangan UI halaman riwayat skor iterasi kedua

Pada tabel 5.14 akan menjelaskan penomoran bagian-bagian dari gambar 5.19 secara rinci mengenai komponen-komponen beserta keterangannya.

Tabel 5.14 Komponen UI halaman riwayat skor iterasi kedua

No.	Nama Komponen	Keterangan
1.	<i>ImageView</i>	Dengan adanya perubahan <i>user stories</i> terkait penambahan fungsional menampilkan riwayat skor pengerjaan soal, maka pada iterasi ini terdapat <i>icon history</i> ini, ketika di tekan oleh pengguna maka akan menuju ke halaman riwayat skor.
2.	<i>ImageView</i>	Komponen yang dijadikan sebagai tombol kembali ke halaman sebelumnya.
3.	<i>TextView</i>	Digunakan sebagai judul riwayat skor sesuai jilid iqra yang dipilih.
4.	<i>CardView</i>	Dalam setiap <i>RecyclerView</i> daftar riwayat skor ini, beberapa komponen dibungkus oleh <i>CardView</i> ini. Hal ini bertujuan supaya terlihat timbul dan bisa memberi efek bayangan pada setiap <i>list</i> nya.
5.	<i>TextView</i>	Komponen ini digunakan untuk menampilkan beberapa skor antara lain skor dari 4 parameter suara (<i>pitch</i> , <i>volume</i> , <i>rhythm</i> , <i>timbre</i>) dan <i>overall</i>

		score, beserta tanggal dan waktu pengguna mengerjakan soal bacaan huruf hijaiyah tersebut.
6.	<i>ImageView</i>	Komponen terakhir yaitu gambar <i>speaker</i> sebagai tanda ketika pengguna klik salah satu item <i>RecyclerView</i> maka akan muncul <i>icon</i> ini. Hal tersebut bertujuan supaya pengguna dapat melihat huruf mana yang berbunyi.

5.2 Implementasi

Pada tahap implementasi ini dilakukan menurut proses perancangan yang telah dibuat sebelumnya. Adapun tahap implementasi disini dimulai dengan mendefinisikan batasan implementasi berupa spesifikasi perangkat lunak dan perangkat keras yang digunakan, kemudian implementasi algoritma dan basis data ke kode program dan yang terakhir yaitu implementasi *user interface* (antarmuka) Aplikasi *Scoring Iqra*.

5.2.1 Batasan Implementasi

Pada subbab ini menjelaskan batasan-batasan secara rinci ketika proses implementasi aplikasi *scoring iqra*. Adapun batasannya dibedakan menjadi 2 bagian yaitu dari sisi perangkat lunak dan sisi perangkat keras.

5.2.1.1 Spesifikasi Sistem Perangkat Lunak

Dalam proses pengembangan Aplikasi *Scoring Iqra* menggunakan beberapa perangkat lunak yang dijelaskan pada tabel 5.15.

Tabel 5.15 Spesifikasi sistem perangkat lunak

Nama Komponen	Spesifikasi
Sistem Operasi	Windows 10 Pro
Editor dokumentasi	Microsoft Office Word 2016
Editor perancangan	Visual Paradigm 16.0
Editor pemrograman	Android Studio 4.2.1
Editor desain UI	Figma
Bahasa pemrograman	Java, XML

5.2.1.2 Spesifikasi Sistem Perangkat Keras

Dalam proses pengembangan Aplikasi *Scoring Iqra* menggunakan perangkat keras yang dijelaskan pada tabel 5.16.

Tabel 5.16 Spesifikasi sistem perangkat keras

Nama Komponen	Spesifikasi
Laptop	<ol style="list-style-type: none"> 1. Asus A556U (15,6 inch, 1TB, 64-bit) 2. Processor 2,5 GHz Intel Core i5 3. Memory RAM 16GB 2133 MHz DDR3 4. Graphic Nvidia 940mx
Smartphone	<ol style="list-style-type: none"> 1. Redmi Note 8 (6,3 inch, 64GB) 2. Processor Octa Core 3. Memory RAM 4GB 4. Versi Android 10
Earphone	<ol style="list-style-type: none"> 1. Samsung Original White 2. Cable type 1.2 m 3. Frequency Response to 20 KHz 4. Driver type 3.5 mm

5.2.2 Implementasi Algoritma

Implementasi algoritma dilakukan berdasarkan tahap perancangan algoritma. Di mana perancangan *pseudocode* yang dibuat sebelumnya, diterjemahkan ke dalam Bahasa pemrograman supaya dapat dijalankan sesuai dengan kebutuhan. Pada proses ini menggunakan IDE Android Studio 4.2.1 dan bahasa pemrograman Java dan XML.

5.2.2.1 Implementasi Algoritma *Pitch Scoring*

Pada implementasi algoritma *pitch scoring* ini akan menampilkan skor *pitch* dari kedua berkas audio .wav antara berkas audio pengguna dengan berkas audio ustaz. Di mana perancangan algoritma *pitch scoring* telah diimplementasikan ke kode program seperti pada tabel 5.17.

Tabel 5.17 Potongan kode program *pitch scoring*

```

1 public double getPitchScore(double dist) {
2     double powValue = dist <= 0.2 ? map(dist, 0.2, 0.00,
3     0.7, 0.99999) : map(dist, 0.2, 1.0, 0.5, 0.0);
4     return Math.round(100 * Math.sqrt(Math.pow(powValue,
5     2)));
6 }
7
8 public static double[] getDistance2File(String file1,
9 String file2) {
10 Pair<float[], float[]> featurePair;
11 double[] ret = new double[]{0.0, 0.0};
12

```

```

13 //inisialisasi
14 float[] arrPitch = new float[1];
15 float[] arrVolume = new float[1];
16
17 //memanggil method getVolumeAndPitchPair
18 featurePair = getVolumeAndPitchPair(file1, arrVolume,
19 arrPitch);
20 arrPitch = featurePair.second;
21
22 float[] arrPitch2 = new float[1];
23 float[] arrVolume2 = new float[1];
24
25 featurePair = getVolumeAndPitchPair(file2, arrVolume2,
26 arrPitch2);
27 arrPitch2 = featurePair.second;
28
29 //instansiasi objek DTW
30 DTW dtw = new DTW(arrPitch, arrPitch2);
31 //instansiasi objek Score
32 Score score = new Score();
33
34 //memanggil method getInBeat dengan parameter pemanggilan
35 method getDistance
36 double pitch =
37 score.getPitchScore(Math.exp(dtw.getDistance()));
38
39 ret[0] = pitch;
40
41 return ret;
42 }

```

5.2.2.2 Implementasi Algoritma *Volume Scoring*

Pada implementasi algoritma *volume scoring* ini akan menampilkan skor *volume* dari kedua berkas audio .wav antara berkas audio pengguna dengan berkas audio ustaz. Di mana perancangan algoritma *volume scoring* telah diimplementasikan ke kode program seperti pada tabel 5.18.

Tabel 5.18 Potongan kode program *volume scoring*

```

1 public double getVolumeScore(double inBeat) {
2     double powValue = map(inBeat, -1, 1, 0.6, 0.99999);
3     return Math.round(100 *
4     Math.sqrt(Math.pow(powValue, 2)));
5 }
6
7 public static double[] getDistance2File(String file1,
8 String file2) {
9     Pair<float[], float[]> featurePair;
10    double[] ret = new double[]{0.0, 0.0};
11
12    //inisialisasi
13    float[] arrPitch = new float[1];
14    float[] arrVolume = new float[1];
15
16    //memanggil method getVolumeAndPitchPair
17

```

```

18 featurePair = getVolumeAndPitchPair(file1, arrVolume,
19 arrPitch);
20 arrVolume = featurePair.first;
21
22 float[] arrPitch2 = new float[1];
23 float[] arrVolume2 = new float[1];
24
25 featurePair = getVolumeAndPitchPair(file2, arrVolume2,
26 arrPitch2);
27 arrVolume2 = featurePair.first;
28
29 //instansiasi objek DTW
30 DTW dtw = new DTW(arrVolume, arrVolume2);
31 //instansiasi objek Score
32 Score score = new Score();
33
34 //memanggil method getPitchScore dengan parameter
35 pemanggilan method getDistance
36 double volume =
37 score.getVolumeScore(Math.exp(dtw.getInBeat()));
38
39 ret[1] = volume;
40
41 return ret;
    }
    
```

5.2.2.3 Implementasi Algoritma *Rhythm Scoring*

Pada implementasi algoritma *rhythm scoring* ini akan menampilkan skor *rhythm* dari kedua berkas audio .wav antara berkas audio pengguna dengan berkas audio ustaz. Di mana perancangan algoritma *rhythm scoring* telah diimplementasikan ke kode program seperti pada tabel 5.19.

Tabel 5.19 Potongan kode program *rhythm scoring*

```

1 public double getRhythmScore(double inBeat) {
2     double powValue = map(inBeat, -0.999, 0.2, 0.0,
3     0.99999);
4     return Math.round(100 * Math.sqrt(Math.pow(powValue,
5     2)));
6 }
7
8 public static double[] getDistance2Feature(String file1,
9 String file2) {
10 Pair<float[], float[]> featurePair;
11 double[] ret = new double[]{0.0, 0.0};
12
13 float[] featureFile1;
14
15 //memanggil method getFeatureFromWaveFile
16 featurePair = getFeatureFromWaveFile(file1);
17 featureFile1= featurePair.first;
18
19 float[] featureFile2;
20
21 featurePair = getFeatureFromWaveFile(file2);
22 featureFile2 = featurePair.first;
    
```

```

23
24 //instansiasi objek DTW
25 DTW dtw = new DTW(featureFile1, featureFile2);
26 //instansiasi objek Score
27 Score score = new Score();
28
29 //memanggil method getRhyScore dengan parameter pemanggilan
30 method getInBeat
31 double rhythm = score.getRhythmScore(dtw.getInBeat());
32
33 ret[0] = rhythm;
34
35 return ret;
36 }

```

5.2.2.4 Implementasi Algoritma *Timbre Scoring*

Pada implementasi algoritma *timbre scoring* ini akan menampilkan skor *timbre* dari kedua berkas audio .wav antara berkas audio pengguna dengan berkas audio ustaz. Di mana perancangan algoritma *timbre scoring* telah diimplementasikan ke kode program seperti pada tabel 5.20.

Tabel 5.20 Potongan kode program *timbre scoring*

```

1 public double getTimbreScore(double bestMeans) {
2     return Math.round(bestMeans);
3 }
4
5 public static double[] getDistance2Feature(String file1,
6 String file2) {
7     Pair<float[], float[]> featurePair;
8     double[] ret = new double[]{0.0, 0.0};
9
10    float[] featureFile1;
11
12    //memanggil method getFeatureFromWaveFile
13    featurePair = getFeatureFromWaveFile(file1);
14    featureFile1= featurePair.first;
15
16    float[] featureFile2;
17
18    featurePair = getFeatureFromWaveFile(file2);
19    featureFile2 = featurePair.first;
20
21    int length = Math.min(featureFile1.length,
22    featureFile2.length);
23
24    // length - 1 ==> karena nilai index terakhir pasti 0,0
25    (supaya sama, jadi index terakhir dibuang saja)
26    double[] featureExtractionFile1 = new
27    double[length - 1];
28    for (int i = 0; i < length - 1; i++) {
29        featureExtractionFile1[i] =
30    Math.abs(featureFile1[i]); // di absolute supaya nilai
31    akhir tidak minus
32    }
33

```

```

34         double[] featureExtractionFile2 = new
35         double[length - 1];
36         for (int i = 0; i < length - 1; i++) {
37             featureExtractionFile2[i] =
38             Math.abs(featureFile2[i]);
39         }
40
41         // input feature dari 2 audio ke array double 2D
42         double[][] dataFeature2Audio =
43         {featureExtractionFile1, featureExtractionFile2};
44
45         // set fitur ekstraksi ke metode GMM
46         GMM gmm = new GMM(dataFeature2Audio, 1);
47         double timbre =
48         score.getTimbreScore(gmm.getMeans());
49
50         ret[1] = timbre;
51
52         return ret;
53     }

```

5.2.2.5 Implementasi Algoritma Overall Score

Pada implementasi algoritma *overall score* ini akan menampilkan skor penggabungan dari keempat parameter *pitch*, *volume*, *rhythm*, dan *timbre*. Di mana perancangan algoritma *overall score* telah diimplementasikan ke kode program seperti pada tabel 5.21.

Tabel 5.21 Potongan kode program *overall score*

```

1     public double getOverAllScore(double Sv, double Sp, double
2     Sr, double St) {
3         double spRate = 0.55 * Sp;
4         double svRate = 0.05 * Sv;
5         double srRate = 0.38 * Sr;
6         double stRate = 0.02 * St;
7         double overallScore = svRate + spRate + srRate +
8         stRate;
9
10        return Double.isNaN(Sv) || Double.isNaN(Sp) ||
11        Double.isNaN(Sr) || Double.isNaN(St) ? 0 : overallScore;
12    }

```

5.2.3 Implementasi Basis Data

Pada bagian ini akan menampilkan hasil implementasi dari perancangan basis data Aplikasi *Scoring Iqra*. Adapun atribut id dapat dijadikan sebagai *primary key* dan dibuat secara *auto generated*. Sedangkan nama database, nama *entitas*, dan nama atribut *entitas*-nya dapat di implementasikan sesuai tahap perancangannya. Gambar 5.20 di ambil pada fitur *Database Inspector* Android Studio 4.2.1.



id	jilid	date	file_name	pitch	volume	rhythm	timbre	overall_score
1	1	09/06/2021 07:45:	/VoiceRecord1-95	48.0	66.0	54.0	80.0	54.66
2	2	09/06/2021 07:45:	/VoiceRecord2-95	96.0	61.0	53.0	75.0	73.93
3	3	09/06/2021 07:45:	/VoiceRecord2-95	91.0	62.0	53.0	69.0	71.56

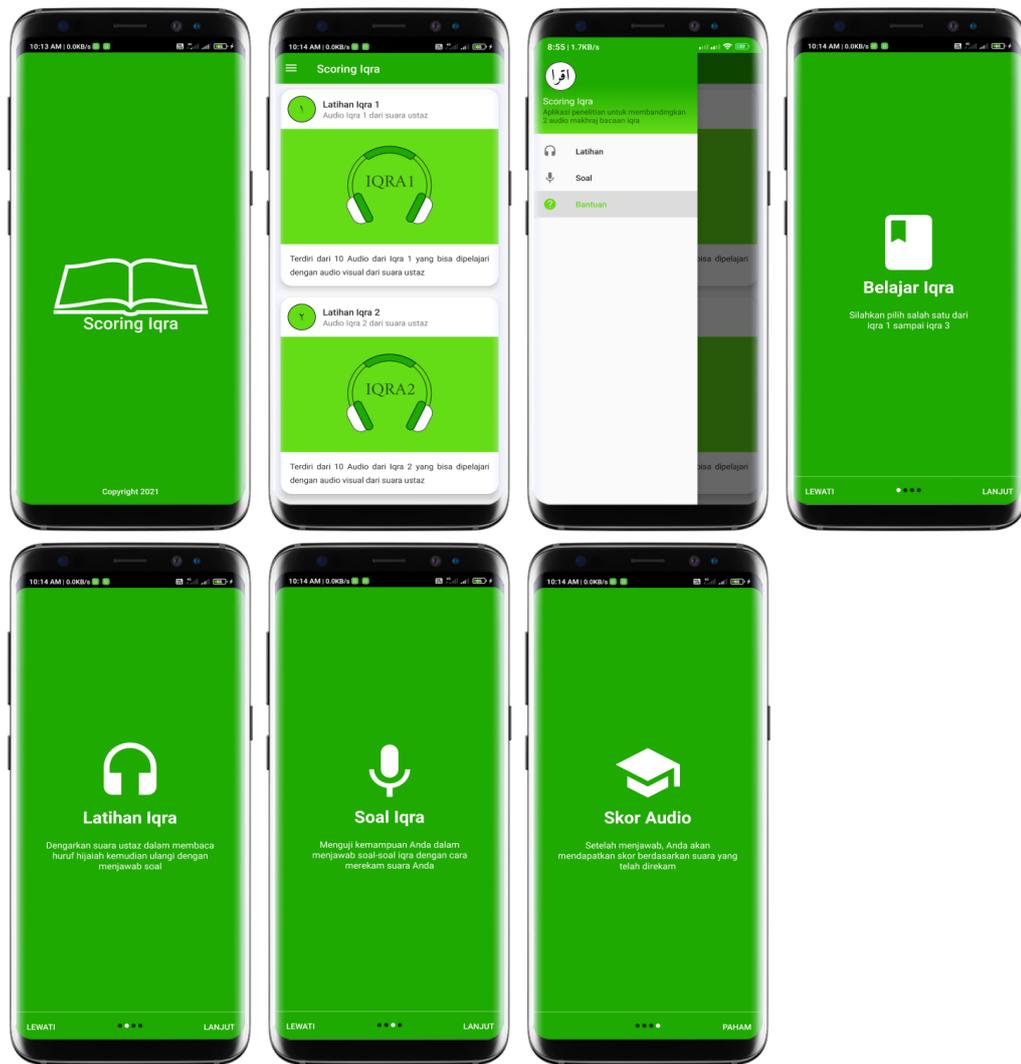
Gambar 5.20 Implementasi Basis data

5.2.4 Implementasi User Interface (UI)

Pada implementasi UI atau antarmuka aplikasi *scoring iqra* ini berdasarkan perancangan UI pada subbab sebelumnya. Pada tahapan implementasi ini dibuat dengan semirip-miripnya dengan tahap perancangan UI-nya dan juga akan dijelaskan secara detail mengenai tampilan dari aplikasi *scoring iqra*. Adapun yang akan dijelaskan pada dokumentasi ini yaitu hanya pada iterasi terakhir (sesuai perancangan UI iterasi kedua). Pada penulisan dokumentasi implementasi UI ini akan dibuat secara *screen flow* menu utama dan halaman baru yaitu halaman riwayat skor pengerjaan soal bacaan huruf hijaiyah.

5.2.4.1 Implementasi UI menu bantuan

Pada *screen flow* ini, akan menampilkan alur untuk menuju ke halaman bantuan yang terdiri dari beberapa halaman yaitu *splashscreen*, halaman utama, *navigation drawer header*, dan menu bantuan. Hasil implementasinya dapat dilihat pada gambar 5.21.



Gambar 5.21 Tampilan implementasi UI menu bantuan

5.2.4.2 Implementasi UI menu latihan

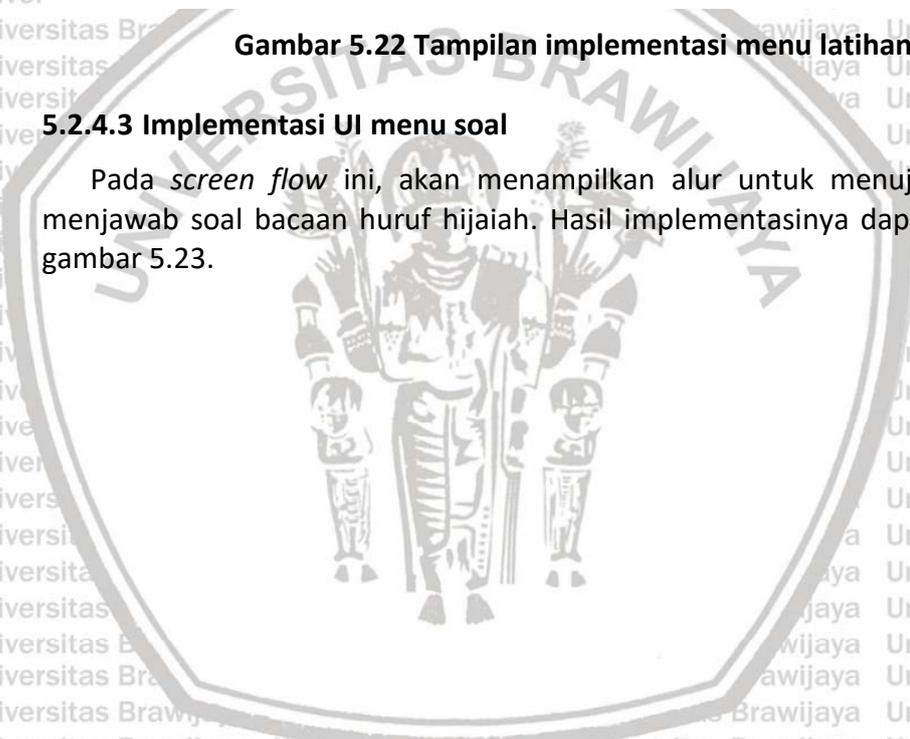
Pada *screen flow* ini, akan menampilkan alur untuk menuju ke halaman latihan cara baca membaca beberapa huruf hijaiyah. Hasil implementasinya dapat dilihat pada gambar 5.22.

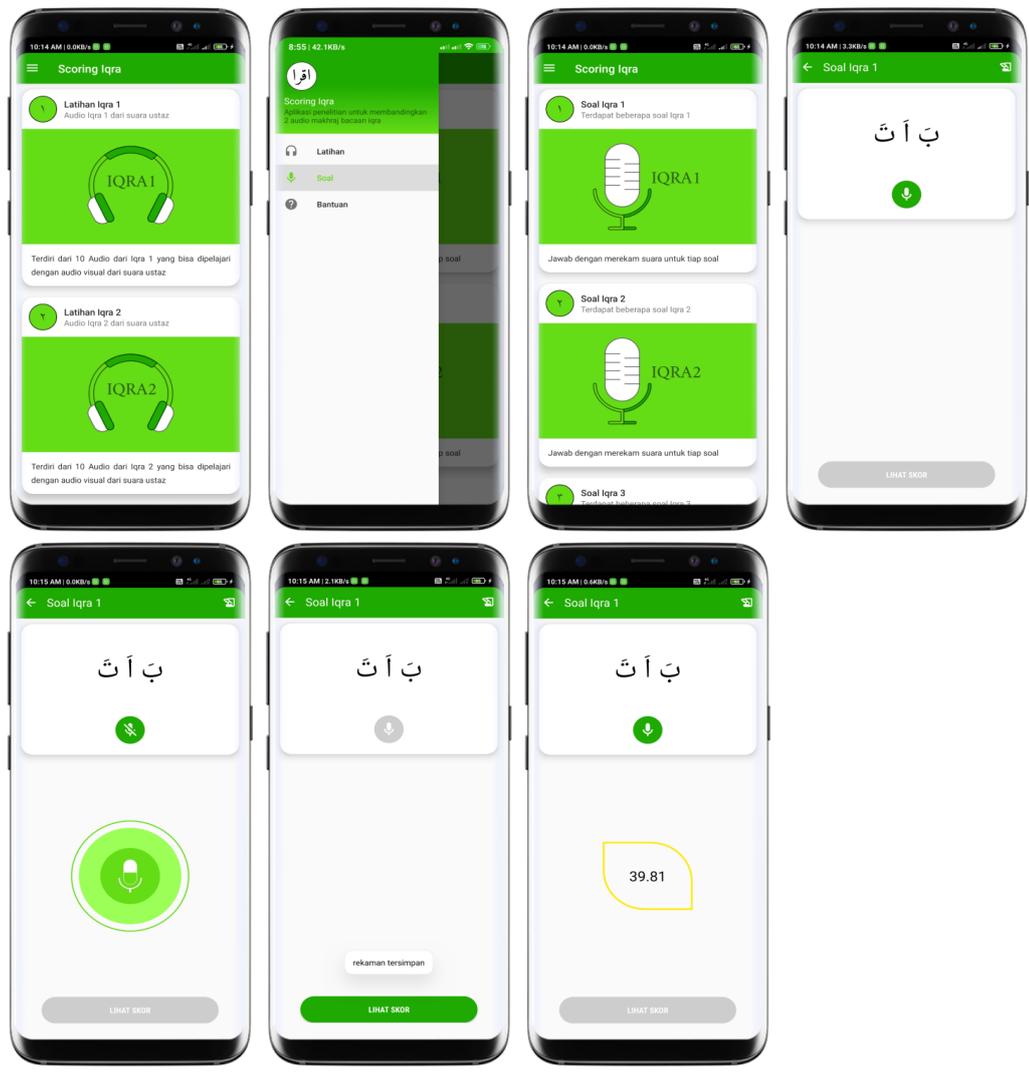


Gambar 5.22 Tampilan implementasi menu latihan

5.2.4.3 Implementasi UI menu soal

Pada *screen flow* ini, akan menampilkan alur untuk menuju ke halaman menjawab soal bacaan huruf hijaiyah. Hasil implementasinya dapat dilihat pada gambar 5.23.

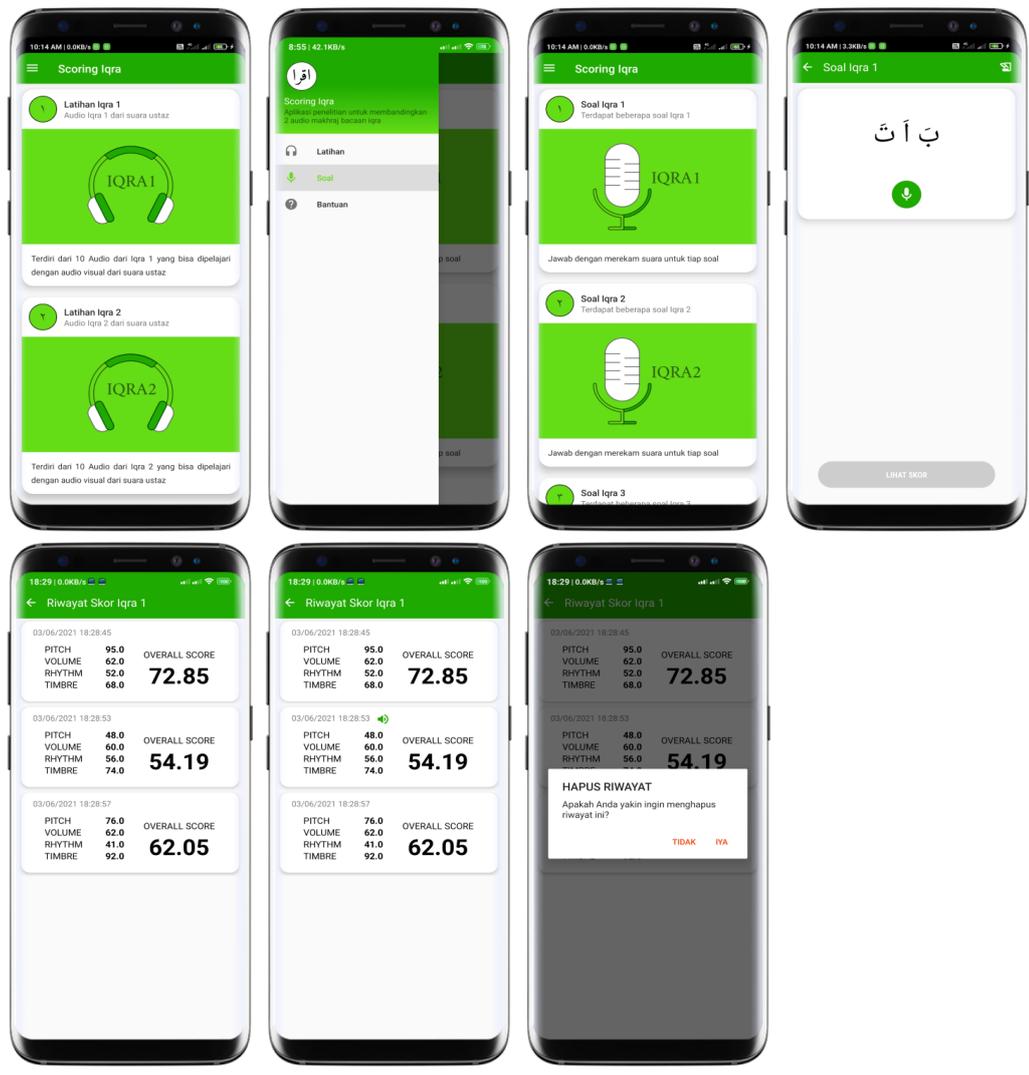




Gambar 5.23 Tampilan implementasi UI menu soal

5.2.4.4 Implementasi UI halaman riwayat skor

Pada *screen flow* ini, akan menampilkan alur untuk menuju ke halaman riwayat skor pengerjaan soal bacaan huruf hijaiyah. Hasil implementasi dapat dilihat pada gambar 5.24.



Gambar 5.24 Tampilan implementasi UI halaman riwayat skor

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab pengujian dan analisis ini menjelaskan 3 pengujian yaitu pengujian unit dari kode program untuk mendapatkan *overall score*, pengujian validasi yaitu pengujian *instrument* terhadap semua fungsional dengan teknik *black box*, dan pengujian korelasi kemiripan antara hasil *overall score* suara rekaman pengguna dengan suara rekaman ustaz.

6.1 Pengujian Unit

Pengujian unit ini bertujuan untuk menguji alur algoritma yang di implementasi sesuai dengan yang diharapkan. Pengujian ini fokus pada algoritma untuk mendapatkan *overall score*, dimana nilai tersebut berasal dari 4 parameter yang digunakan yaitu *pitch*, *volume*, *rhythm*, dan *timbre*. Selain itu, pada subbab ini akan menjelaskan hasil dari skenario tersebut.

6.1.1 Skenario Pengujian *Overall Score*

Terdapat 2 skenario pengujian unit terhadap nilai *overall score*. Pengujian pertama yaitu mendapatkan nilai *overall score* dengan cara memasukkan 2 berkas audio rekaman bacaan iqra 1 dari pengguna dan ustaz terlebih dahulu ke dalam fungsi `getDistance2Feature()` untuk mendapatkan nilai skor dari parameter *rhythm* dan *timbre*, kemudian berkas audio rekaman bacaan iqra 1 dari pengguna dan ustaz dimasukkan ke dalam fungsi `getDistance2File()` untuk mendapatkan nilai skor dari parameter *volume* dan *pitch*. Setelah nilai dari 4 parameter didapatkan, selanjutnya memanggil fungsi `getOverallScore()` yang ada pada *class Score* untuk mendapatkan nilai *overall score*. Tabel skenario pengujian unit pertama dapat dilihat pada tabel 6.1.

Tabel 6.1 Skenario pengujian fungsi `getOverallScoreTest`

Nama pengujian	<code>getOverallScoreTest()</code>
Tujuan pengujian	Mendapatkan nilai <i>overall score</i> dari <i>class Score</i> .
Prosedur pengujian	Memanggil beberapa fungsi yang ada di <i>class Score</i> . 1. <code>getDistance2Feature()</code> 2. <code>getDistance2File()</code> 3. <code>getOverallScore()</code>
Hasil yang diharapkan	Fungsi <code>getOverallScore()</code> pada <i>class Score</i> dapat mengembalikan nilai yang bukan bernilai 0 ataupun bernilai null.

Pengujian kedua yaitu cek kesamaan antara *overall score* yang diharapkan dengan *overall score* kenyataan yang telah diuji pada tabel 6.1. Adapun nilai harapan dari *pitch*, *volume*, *rhythm*, dan *timbre* didapatkan di daftar riwayat skor

aplikasi *Scoring*. Sedangkan *overall score* akan ditentukan secara manual dengan pembobotan masing-masing parameternya yaitu 55%, 5%, 38%, dan 2%. Tabel skenario pengujian unit kedua dapat dilihat pada tabel 6.2.

Tabel 6.2 Skenario pengujian fungsi isExpectEqualActual

Nama pengujian	isExpectEqualActual ()
Tujuan pengujian	Mengecek apakah nilai harapan dari <i>overall score</i> perhitungan manual sama dengan nilai kenyataan dari <i>overall score</i> pada <i>class Score</i> .
Prosedur pengujian	<ol style="list-style-type: none"> 1. Mengatur skor harapan dari 4 parameter. 2. Menghitung <i>overall score</i> sesuai pembobotan pada masing-masing skor parameter sebesar 55%, 5%, 38%, dan 2%. 3. Cek kesamaan dengan <code>assertEquals</code> dari <code>junit4</code>.
Hasil yang diharapkan	Nilai harapan dari <i>overall score</i> perhitungan manual sama dengan nilai kenyataan dari <i>overall score</i> pada <i>class Score</i> dan <code>assertEquals</code> tidak mengembalikan pesan error.

Adapun implementasi kode pengujian dapat dilihat pada tabel 6.3. Di mana dari kode program tersebut menggunakan *library junit4* sebagai *tools* dalam pengujian unit ini. `@Before` berfungsi sebagai instansiasi kelas awal dan `@Test` berfungsi sebagai anotasi fungsi yang ingin di uji.

Tabel 6.3 Implementasi kode pengujian ScoreTest

```

1 public class ScoreTest {
2     private final String fileUji =
3     "C:\\Users\\USER\\AndroidStudioProjects\\ScoringIqro\\app\\
4     src\\main\\res\\raw\\baata.wav";
5     private final String fileRef =
6     "C:\\Users\\USER\\AndroidStudioProjects\\ScoringIqro\\app\\
7     src\\main\\res\\raw\\iqro1b.wav";
8
9     private final double[] srt =
10    Score.getDistance2Feature(fileUji, fileRef);
11    private final double[] svp =
12    Score.getDistance2File(fileUji, fileRef);
13
14    private Score score;
15
16    @Before
17    public void setUp() {
18        score = new Score();
19    }
20
21    @Test
22    public void getOverallScoreTest() {

```

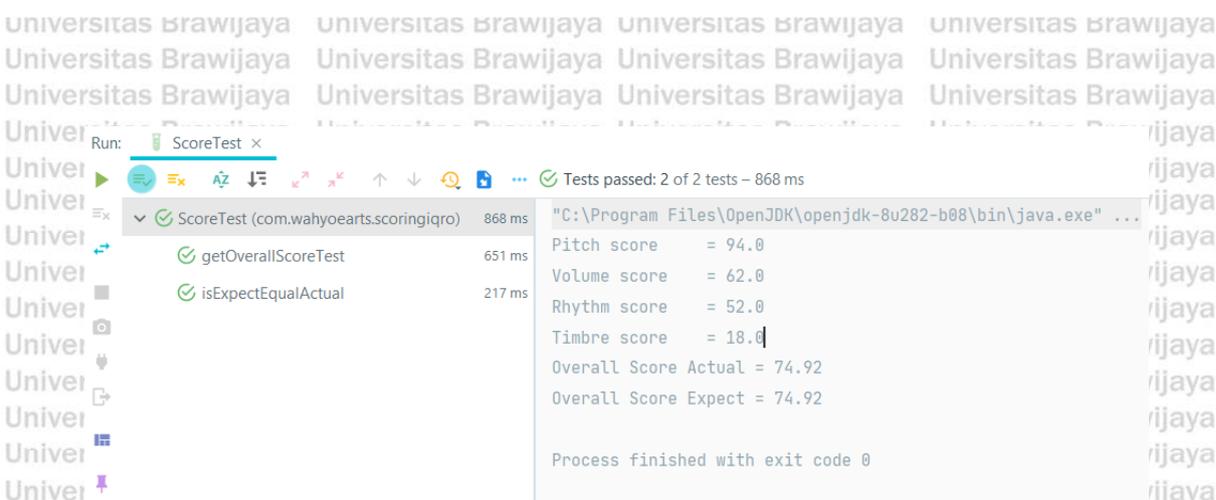
```

23 // scoring with apps - test method
24 getOverallScore()
25     System.out.println("Pitch score\t\t= " + svp[1]);
26     System.out.println("Volume score\t= " + svp[0]);
27     System.out.println("Rhythm score\t= " + srt[0]);
28     System.out.println("Timbre score\t= " + srt[1]);
29
30     double overallScoreActual =
31     score.getOverAllScore(svp[0], svp[1], srt[0], srt[1]);
32     System.out.println("Overall Score Actual = " +
33     overallScoreActual);
34     }
35
36     @Test
37     public void isExpectEqualActual() {
38         // expect score
39         double pitch = 94.0;
40         double volume = 62.0;
41         double rhythm = 52.0;
42         double timbre = 18.0;
43         double overallScoreExpect = volume * 0.05 + pitch *
44     0.55 + rhythm * 0.38 + timbre * 0.02;
45         System.out.println("Overall Score Expect = " +
46     overallScoreExpect);
47
48         // actual score
49         double overallScoreActual =
50     score.getOverAllScore(svp[0], svp[1], srt[0], srt[1]);
51
52
53     Assert.assertEquals(String.valueOf(overallScoreExpect),
54     String.valueOf(overallScoreActual));
55     }
56     }

```

6.1.2 Analisis Hasil Pengujian Unit

Setelah melakukan implementasi kode pada tabel 6.3, kemudian dilakukan proses pengujian. Dari hasil kedua pengujian menghasilkan keluaran sesuai harapan. Di mana pengujian pertama dapat menghasilkan nilai yang bukan 0 ataupun nilai null. Sedangkan pengujian kedua dapat menghasilkan kesamaan nilai antara nilai *overall score* harapan dan nilai *overall score* kenyataannya. *Screenshot* keluaran tersebut dapat dilihat pada gambar 6.1.



Gambar 6.1 Hasil pengujian unit overall score

6.2 Pengujian Validasi

Pengujian validasi ini menggunakan teknik *black box* yang bertujuan untuk menguji *instrument* dari fungsional yang disediakan oleh Aplikasi telah sesuai dengan kebutuhan fungsional pada bab sebelumnya. Pada pengujian ini juga terdapat analisis dari hasil skenario pengujian, sehingga dapat memastikan seberapa besar kebutuhan fungsional yang sudah diimplementasikan ke dalam Aplikasi *Scoring Iqra*.

6.2.1 Skenario Pengujian Validasi

Pada skenario pengujian dibawah terdapat 9 fungsi Aplikasi yang diuji sesuai kebutuhan fungsional pada bab sebelumnya. Skenario pengujian validasi dapat dilihat pada Tabel 6.4.

Tabel 6.4 Skenario pengujian validasi Aplikasi *Scoring Iqra*

No. Uji	Fungsi (Kode)	Prosedur Skenario Uji	Espektasi	Hasil
TES_01	Menampilkan 3 menu jilid iqra. (SI – 1 – 001)	Jika pengguna ingin memilih jilid iqra pada menu latihan, maka pengguna sudah bisa memilih jilid yang telah disediakan oleh sistem pada halaman utama Aplikasi. Selain cara tersebut, pengguna dapat memilih tombol <i>navigation drawer header</i> yang ada pada pojok kiri atas pada Aplikasi, kemudian pilih menu latihan.	Sistem dapat menampilkan 3 jilid iqra, baik di dalam menu latihan maupun menu soal.	VALID

		Jika pengguna ingin memilih jilid iqra pada menu soal, maka pengguna harus memilih tombol <i>navigation drawer header</i> yang ada pada pojok kiri atas pada Aplikasi dahulu. Kemudian pilih menu soal.		
TES_02	Membunyikan 10 suara bacaan huruf hijaiyah. (SI – 1 – 002)	<p>Pengguna memilih salah satu jilid dari ketiga jilid yang ada.</p> <p>Pengguna memilih salah satu huruf hijaiyah.</p> <p>Sistem akan menampilkan 10 daftar kata / huruf hijaiyah.</p> <p>Pengguna memilih salah satu daftar kata / huruf hijaiyah.</p> <p>Jika pengguna tidak berada pada menu latihan, maka harus pindah dahulu melalui tombol <i>navigation drawer header</i> pada pojok kiri atas Aplikasi.</p>	Setelah pengguna memilih salah satu daftar kata / huruf hijaiyah , maka Sistem dapat mengeluarkan bunyi dari huruf tersebut sesuai rekaman ustaz. Dimana <i>list</i> yang mengeluarkan suara dapat ditandai dengan munculnya <i>icon speaker</i> pada <i>list</i> tersebut.	VALID
TES_03	Menampilkan 1 soal bacaan huruf hijaiyah. (SI – 1 – 003)	<p>Pengguna menekan tombol <i>navigation drawer header</i> pada pojok kiri atas Aplikasi.</p> <p>Pengguna memilih menu soal, kemudian sistem akan menampilkan daftar ketiga jilid iqra.</p> <p>Pengguna memilih salah satu jilid iqra</p>	Sistem akan menampilkan 1 soal bacaan huruf hijaiyah untuk dikerjakan oleh pengguna.	VALID

TES_04	Merekam suara untuk menjawab soal. (SI – 1 – 004)	Pengguna menekan tombol <i>navigation drawer header</i> pada pojok kiri atas aplikasi.	Sistem akan menyimpan berkas rekaman suara pengguna ke dalam penyimpanan lokal dari perangkat pengguna berupa berkas audio yang berformat (.wav). Sistem juga akan memberikan <i>feedback</i> kepada pengguna berupa notifikasi bahwa rekaman telah berhasil disimpan.	VALID
		Pengguna memilih menu soal, kemudian sistem akan menampilkan daftar ketiga jilid iqra.		
		Pengguna memilih salah satu jilid iqra, dan sistem akan menampilkan soal untuk dikerjakan oleh pengguna.		
		Sebelum merekam suaranya, pengguna harus memilih tombol <i>icon microphone on</i> terlebih dahulu. Dan sistem akan mulai merekam suara yang ditandai muncul animasi <i>microphone</i> .		
		Kemudian pengguna harus mengakhiri rekaman suaranya, dengan cara menekan tombol <i>icon microphone off</i> .		
TES_05	Menghitung skor kemiripan suara rekaman. (SI – 1 – 005)	Sebelumnya pengguna harus berhasil menyimpan berkas audio rekamannya.	Sistem akan menampilkan skor pada tengah layar perangkat pengguna. Skor yang ditampilkan dalam rentang 0,00 sampai 100,00.	VALID
		Setelah itu, pengguna menekan tombol “lihat skor” yang ada pada halaman untuk merekam suara. Sistem akan memulai proses kalkulasi korelasi kemiripan antara		

		rekaman suara dari pengguna dengan suara referensi dari ustaz.		
TES_06	Menampilkan cara penggunaan aplikasi. (SI – 1 – 006)	Pengguna menekan tombol <i>navigation drawer header</i> pada pojok kiri atas aplikasi. Pengguna memilih menu bantuan.	Sistem akan menampilkan <i>Onboarding Screen</i> . Sehingga Pengguna dapat menekan tombol “lanjut” untuk mengetahui cara penggunaan Aplikasi selanjutnya atau bisa juga dengan cara <i>swipe/ geser</i> ke kanan dan geser ke kiri untuk kembali ke cara penggunaan Aplikasi sebelumnya. Serta pengguna dapat menekan tombol “lewati” untuk keluar dari menu bantuan.	VALID
TES_07	Menampilkan riwayat skor pengerjaan soal. (SI – 1 – 007)	Pengguna harus berada pada halaman mengerjakan soal sesuai TES_03. Pengguna menekan tombol <i>icon history</i> pada pojok kanan atas Aplikasi.	Sistem akan menampilkan halaman riwayat skor pengerjaan soal pengguna dalam bentuk <i>list</i> yang menampilkan 4 skor parameter suara (<i>pitch, volume, rhythm, dan timbre</i>), dan <i>overall score</i> , beserta tanggal dan waktu pengerjaan soal.	VALID

TES_08	Membunyikan riwayat suara rekaman. (SI - 1 - 008)	Pengguna harus berada pada halaman mengerjakan soal sesuai TES_03.	Sistem akan membunyikan rekaman suara pengguna, yang dapat ditandai dengan munculnya <i>icon speaker</i> pada daftar riwayat yang dipilih.	VALID
		Pengguna menekan tombol <i>icon history</i> pada pojok kanan atas Aplikasi.		
		Pengguna menekan salah satu daftar riwayat skor pengerjaan soal.		
TES_09	Menghapus riwayat skor pengerjaan soal. (SI - 1 - 009)	Pengguna harus berada pada halaman mengerjakan soal sesuai TES_03.	Sistem akan menghapus daftar riwayat skor pengerjaan dari lokal database.	VALID
		Pengguna menekan tombol <i>icon history</i> pada pojok kanan atas Aplikasi.		
		Pengguna melakukan menekan dan tahan pada salah satu daftar riwayat yang ingin dihapus.		
		Sistem akan menampilkan <i>pop up</i> konfirmasi ke pengguna, untuk memastikan bahwa pengguna benar-benar ingin menghapus daftar riwayat tersebut.		
		Pengguna memilih tombol "IYA" untuk melanjutkan hapus riwayat skor pengerjaannya, atau tombol "TIDAK" untuk membatalkan menghapus daftar		

		riwayat skor pengerjaannya.		
--	--	-----------------------------	--	--

6.2.2 Analisis Hasil Pengujian Validasi

Pada skenario pengujian validasi dari Tabel 6.4. Telah melakukan 9 pengujian fungsional sesuai analisis kebutuhan pada bab sebelumnya. 9 Fungsional tersebut yaitu menampilkan 3 menu jilid iqra, membunyikan 10 suara bacaan huruf hijaiyah, menampilkan 1 soal bacaan huruf hijaiyah, merekam suara untuk menjawab soal, menghitung skor kemiripan suara rekaman, menampilkan cara penggunaan aplikasi, menampilkan riwayat skor pengerjaan soal, dan membunyikan riwayat suara rekaman, serta menghapus riwayat skor pengerjaan soal. Adapun hasil yang didapatkan yaitu semua fungsional tersebut dapat diimplementasikan dan dijalankan dengan baik pada platform Android minimal pada SDK versi 21.

6.3 Pengujian Korelasi Kemiripan

Pengujian korelasi kemiripan pada Aplikasi *Scoring Iqra* dilakukan dengan cara membandingkan dari kedua berkas audio rekaman pengguna dengan rekaman ustaz. Pada pengujian ini menggunakan algoritma *Pearson Product Moment Coeficient Correlation*. Sebelum menguji korelasi kemiripan antara 2 *overall score* dari berkas audio rekaman pengguna dengan rekaman ustaz, perlu mencari jumlah responden. Adapun rumus untuk mencari jumlah responden yaitu menggunakan Rumus Slovin. Dimana pada penelitian ini jumlah populasi terdapat sebanyak 100 orang. Sedangkan *margin of error* ditetapkan sebanyak 25 % atau 0,25. Maka didapatkan jumlah sampel sebanyak 13,8 atau bisa dibulatkan menjadi 14 responden.

Demografi responden yang telah ditentukan pada penelitian ini yaitu pelajar dengan usia 8 – 15 tahun. Rata – rata semua responden tersebut sudah pernah belajar mengaji menggunakan metode Iqra pada Taman Pendidikan Al-Qur’an (TPA). Setidaknya dalam kegiatan belajar mengaji tersebut, responden telah lulus pada iqra 1, 2, dan 3. Dari 14 responden tersebut terdapat 6 laki-laki dan 8 perempuan. Sedangkan ustaz berperan besar yaitu untuk menilai bacaan dari responden secara manual, di mana ustaz memiliki pengalaman baik dalam mengajar Iqra ataupun Al-Qur’an baik di sekolah maupun TPA di masjid. Adapun foto ustaz dan 14 responden saat melakukan pengujian telah ditambahkan pada lampiran dokumen skripsi ini.

Adapun skenario pengujian korelasi kemiripan ini yaitu masing-masing responden membaca huruf hijaiyah sebanyak 3 soal. Soal tersebut meliputi soal pada iqra jilid 1, iqra jilid 2, dan iqra jilid 3. Setelah semua responden selesai membaca soal huruf hijaiyah, ustaz melakukan penilaian terhadap suara responden melalui daftar riwayat skor pada aplikasi *Scoring Iqra*. Maka akan menghasilkan 2 *overall score* antara penilaian dari aplikasi dan penilaian ustaz di setiap responden, sehingga dilakukan pencarian nilai korelasi kemiripan pada masing-masing jilid iqra dengan rumus *pearson correlation coefficient* dan dapat disimpulkan

kriterianya berdasarkan penelitian (Ratner, 2009). Sehingga dapat terlihat perubahan korelasi kemiripan dari penelitian sebelumnya seperti tabel 6.5.

Tabel 6.5 Perubahan korelasi kemiripan

IQRA	Perubahan Hasil Korelasi Kemiripan			
	Penelitian (Nugraha, et al., 2019)		Penelitian Sekarang	
	Nilai	Kriteria	Nilai	Kriteria
Iqra Jilid 1	0,034	Lemah	0,376	Sedang
Iqra Jilid 2	0,51	Sedang	0,795	Kuat
Iqra Jilid 3	-0,17	Lemah	0,255	Lemah

Dari tabel 6.5 tersebut dapat diketahui bahwa nilai korelasi kemiripan pada penelitian ini mengalami kenaikan dari penelitian sebelumnya. Di mana pada Iqra jilid 1 mengalami kenaikan sebesar 0.342, sedangkan pada Iqra jilid 2 mengalami kenaikan sebesar 0.285, dan pada Iqra jilid 3 mengalami kenaikan sebesar 0.425. Adapun perhitungan dalam mencari korelasi kemiripan pada iqra jilid 1 dapat dilihat pada tabel 6.6, Perhitungan dalam mencari korelasi kemiripan pada iqra jilid 2 dapat dilihat pada tabel 6.7, dan Perhitungan dalam mencari korelasi kemiripan pada iqra jilid 3 dapat dilihat pada tabel 6.8.



Tabel 6.6 Perhitungan korelasi kemiripan iqra jilid 1

						IMPLEMENTASI METODE PEARSON CORRELATION COEFFICIENT												
Penilaian Apps	PITCH	VOLUME	RHYTHM	TIMBRE	OverallScore	ERROR RATE	x	y	x.y	x ²	y ²	avg x	avg y	x - avg x	y - avg y	Dx . Dy	Dx ²	Dy ²
							Responden 1	90	62	51	37	72.72	2.7027027	72.72	74.74	5435.0928	5288.1984	5586.0676
Responden 2	92	61	55	25	75.05	5.41902962	75.05	79.35	5955.2175	5632.5025	6296.4225	72.03	74.92	3.0229	4.4321	13.39773	9.137665306	19.64389031
Responden 3	95	62	47	29	73.79	0.76654115	73.79	74.36	5487.0244	5444.9641	5529.4096	72.03	74.92	1.7629	-0.558	-0.983422	3.107665306	0.311204592
Responden 4	81	61	56	35	69.58	3.19977741	69.58	71.88	5001.4104	4841.3764	5166.7344	72.03	74.92	-2.4471	-3.038	7.43407	5.988508163	9.22857602
Responden 5	86	62	49	38	69.78	1.0212766	69.78	70.5	4919.49	4869.2484	4970.25	72.03	74.92	-2.2471	-4.418	9.927556	5.04965102	19.51746173
Responden 6	76	63	63	32	69.53	8.57330703	69.53	76.05	5287.7565	4834.4209	5783.6025	72.03	74.92	-2.4971	1.1321	-2.827122	6.235722449	1.281747449
Responden 7	78	62	59	58	69.58	7.75553493	69.58	75.43	5248.4194	4841.3764	5689.6849	72.03	74.92	-2.4471	0.5121	-1.253287	5.988508163	0.262290306
Responden 8	93	62	50	23	73.71	0.76555024	73.71	73.15	5391.8865	5433.1641	5350.9225	72.03	74.92	1.6829	-1.768	-2.975051	2.832008163	3.125318878
Responden 9	73	64	65	58	69.21	8.9581689	69.21	76.02	5261.3442	4790.0241	5779.0404	72.03	74.92	-2.8171	1.1021	-3.104894	7.936293878	1.214718878
Responden 10	86	64	57	65	73.46	3.72214941	73.46	76.3	5604.998	5396.3716	5821.69	72.03	74.92	1.4329	1.3821	1.980413	2.053079592	1.910318878
Responden 11	81	63	64	27	72.56	3.29201653	72.56	75.03	5444.1768	5264.9536	5629.5009	72.03	74.92	0.5329	0.1121	0.059756	0.283936735	0.01257602
Responden 12	82	61	58	22	70.63	5.52434457	70.63	74.76	5280.2988	4988.5969	5589.0576	72.03	74.92	-1.3971	-0.158	0.220549	1.952008163	0.024918878
Responden 13	93	64	53	39	75.27	0.95225322	75.27	74.56	5612.1312	5665.5729	5559.1936	72.03	74.92	3.2429	-0.358	-1.16048	10.51612245	0.128061735
Responden 14	87	60	57	50	73.51	4.18404588	73.51	76.72	5639.6872	5403.7201	5885.9584	72.03	74.92	1.4829	1.8021	2.67232	2.198865306	3.247718878
						MEAN ERROR	tot x	tot y	tot x.y	tot x ²	tot y ²				Tot Dx.Dy	Tot Dx ²	Tot Dy ²	
Penilaian Ustaz	PITCH	VOLUME	RHYTHM	TIMBRE	OverallScore	4.0597642	1008.38	1048.85	75568.9337	72694.49	78637.535				A	23.26491	63.76008571	59.94043571
Responden 1	85	75	60	72	74.74	similarity	(tot x) ²	(tot y) ²			sqrt tot x ²	sqrt tot y ²				sqrt (tot Dx ² + tot Dy ²)		
Responden 2	87	72	70	65	79.35	95.940236	1016830	1100086.32			269.61916	280.42385				61.8207677		
Responden 3	81	73	65	73	74.36		avg x	avg y										
Responden 4	80	70	61	60	71.88		72.02714	74.9178571										
						HASIL KORELASI KEMIRIPAN						KETERANGAN						
Responden 5	78	72	60	60	70.5	Tot Dx.Dy / sqrt (tot Dx ² + tot Dy ²) or A/B						x						
Responden 6	81	70	70	70	76.05	HASIL 0.376 KORELASI SEDANG						OverallScore aplikasi						
Responden 7	80	73	69	78	75.43							y						
Responden 8	80	65	65	60	73.15							tot						
Responden 9	82	68	69	65	76.02							Total						
Responden 10	83	69	68	68	76.3	ERROR RATE =												
Responden 11	79	72	70	69	75.03	ABS(y-x) / y * 100												
Responden 12	80	70	68	71	74.76													
Responden 13	80	74	67	70	74.56													
Responden 14	83	73	69	60	76.72													

Tabel 6.7 Perhitungan korelasi kemiripan iqra jilid 2

IMPLEMENTASI METODE PEARSON CORRELATION COEFFICIENT																		
Penilaian Apps	PITCH	VOLUME	RHYTHM	TIMBRE	OverallScore	ERROR RATE	x	y	x.y	x ²	y ²	avg x	avg y	x - avg x	y - avg y			
							Dx	Dy	Dx . Dy	Dx ²	Dy ²							
Responden 1	91	62	68	37	79.73	2.59940806	79.73	77.71	6195.8183	6356.8729	6038.8441	73.3	75.49	6.4293	2.2229	14.29138	41.3357148	4.941093878
Responden 2	93	64	57	41	76.83	3.1635997	76.83	79.34	6095.6922	5902.8489	6294.8356	73.3	75.49	3.5293	3.8529	13.59783	12.45585765	14.84450816
Responden 3	97	62	41	28	72.59	5.86175593	72.59	77.11	5597.4149	5269.3081	5945.9521	73.3	75.49	-0.7107	1.6229	-1.153388	0.505114796	2.633665306
Responden 4	84	62	59	61	72.94	1.61855948	72.94	74.14	5407.7716	5320.2436	5496.7396	73.3	75.49	-0.3607	-1.347	0.485934	0.130114796	1.814793878
Responden 5	98	62	51	25	76.88	0.0260078	76.88	76.9	5912.072	5910.5344	5913.61	73.3	75.49	3.5793	1.4129	5.057019	12.81128622	1.996165306
Responden 6	91	64	58	67	76.63	2.58366801	76.63	74.7	5724.261	5872.1569	5580.09	73.3	75.49	3.3293	-0.787	-2.620623	11.08414337	0.619593878
Responden 7	92	64	58	27	76.38	0.36524915	76.38	76.66	5855.2908	5833.9044	5876.7556	73.3	75.49	3.0793	1.1729	3.611562	9.48200051	1.375593878
Responden 8	98	61	49	49	76.55	3.87995982	76.55	79.64	6096.442	5859.9025	6342.5296	73.3	75.49	3.2493	4.1529	13.49382	10.55785765	17.24622245
Responden 9	84	63	63	33	73.95	4.48204598	73.95	77.42	5725.209	5468.6025	5993.8564	73.3	75.49	0.6493	1.9329	1.254977	0.421571939	3.735936735
Responden 10	93	65	48	19	73.02	0.09595613	73.02	72.95	5326.809	5331.9204	5321.7025	73.3	75.49	-0.2807	-2.537	0.712212	0.07880051	6.437093878
Responden 11	74	61	62	27	67.85	4.55760304	67.85	71.09	4823.4565	4603.6225	5053.7881	73.3	75.49	-5.4507	-4.397	23.96757	29.71028622	19.33486531
Responden 12	81	63	45	23	65.26	10.4801097	65.26	72.9	4757.454	4258.8676	5314.41	73.3	75.49	-8.0407	-2.587	20.80248	64.65308622	6.693308163
Responden 13	91	61	53	29	73.82	1.66511256	73.82	75.07	5541.6674	5449.3924	5635.5049	73.3	75.49	0.5193	-0.417	-0.216616	0.269657653	0.174008163
Responden 14	72	62	54	28	63.78	10.4087653	63.78	71.19	4540.4982	4067.8884	5068.0161	73.3	75.49	-9.5207	-4.297	40.91187	90.64400051	18.46543673
MEAN ERROR							tot x	tot y	tot x.y	tot x ²	tot y ²					Tot Dx.Dy	Tot Dx ²	Tot Dy ²
Penilaian Ustaz	PITCH	VOLUME	RHYTHM	TIMBRE	OverallScore	3.6991286	1026.21	1056.82	77599.8569	75506.066	79876.635			A		134.196	284.1394929	100.3122857
Responden 1	85	60	70	68	77.71	similarity	(tot x) ²	(tot y) ²	sqrt tot x ²		sqrt tot y ²			B		sqrt (tot Dx ² * tot Dy ²)		
Responden 2	86	68	72	64	79.34	96.300871	avg x	avg y	274.78367		282.62455							
Responden 3	84	69	69	62	77.11		73.30071	75.4871429										
Responden 4	80	60	68	65	74.14	HASIL KORELASI KEMIRIPAN						KETERANGAN						
Responden 5	86	70	65	70	76.9	Tot Dx.Dy / sqrt (tot Dx ² * tot Dy ²) or A/B						x Overallscore aplikasi						
Responden 6	84	70	62	72	74.7	HASIL	0.795	KORELASI KUAT	y Overallscore ustaz									
Responden 7	82	72	70	68	76.66							tot Total						
Responden 8	84	70	75	72	79.64													
Responden 9	84	64	70	71	77.42													
Responden 10	79	68	65	70	72.95	ERROR RATE =												
Responden 11	78	65	62	69	71.09	ABS(y-x) / y * 100												
Responden 12	80	60	65	60	72.9													
Responden 13	85	70	62	63	75.07													
Responden 14	80	63	60	62	71.19													

Tabel 6.8 Perhitungan korelasi kemiripan iqra jilid 3

						IMPLEMENTASI METODE PEARSON CORRELATION COEFFICIENT																	
Penilaian Apps	PITCH	VOLUME	RHYTHM	TIMBRE	OverallScore	ERROR RATE	x	y	x.y	x^2	y^2	avg x	avg y	x - avg x	y - avg y	Dx . Dy	Dx^2	Dy^2					
							Responden 1	87	62	54	19	71.85	4.77137177	71.85	75.45	5421.0825	5162.4225	5692.7025	75.15	74.93	-3.2971	0.525	-1.731
Responden 2	94	62	62	31	78.98	3.75722543	78.98	76.12	6011.9576	6237.8404	5794.2544	75.15	74.93	3.8329	1.195	4.580264	14.69079388	1.428025					
Responden 3	93	63	50	31	73.92	5.59386973	73.92	78.3	5787.936	5464.1664	6130.89	75.15	74.93	-1.2271	3.375	-4.141607	1.505879592	11.390625					
Responden 4	91	62	54	62	74.91	0.04003203	74.91	74.94	5613.7554	5611.5081	5616.0036	75.15	74.93	-0.2371	0.015	-0.003557	0.056236735	0.000225					
Responden 5	86	63	58	53	73.55	0.20435967	73.55	73.4	5398.57	5409.6025	5387.56	75.15	74.93	-1.5971	-1.525	2.435643	2.550865306	2.325625					
Responden 6	89	61	62	30	76.16	4.93248829	76.16	72.58	5527.6928	5800.3456	5267.8564	75.15	74.93	1.0129	-2.345	-2.37515	1.025879592	5.499025					
Responden 7	86	62	49	60	70.22	5.15937331	70.22	74.04	5199.0888	4930.8484	5481.9216	75.15	74.93	-4.9271	-0.885	4.360521	24.27673673	0.783225					
Responden 8	98	63	57	19	79.09	8.31279102	79.09	73.02	5775.1518	6255.2281	5331.9204	75.15	74.93	3.9429	-1.905	-7.511143	15.54612245	3.629025					
Responden 9	91	61	59	26	76.04	4.36453472	76.04	72.86	5540.2744	5782.0816	5308.5796	75.15	74.93	0.8929	-2.065	-1.84375	0.797193878	4.264225					
Responden 10	93	61	61	59	78.56	4.74666667	78.56	75	5892	6171.6736	5625	75.15	74.93	3.4129	0.075	0.255964	11.64759388	0.005625					
Responden 11	93	63	58	37	77.08	0.69569699	77.08	77.62	5982.9496	5941.3264	6024.8644	75.15	74.93	1.9329	2.695	5.20905	3.735936735	7.263025					
Responden 12	90	62	41	46	69.1	2.89488477	69.1	71.16	4917.156	4774.81	5063.7456	75.15	74.93	-6.0471	-3.765	22.76749	36.56793673	14.175225					
Responden 13	88	66	57	48	74.32	4.96163683	74.32	78.2	5811.824	5523.4624	6115.24	75.15	74.93	-0.8271	3.275	-2.708893	0.684165306	10.725625					
Responden 14	93	63	61	40	78.28	2.64883294	78.28	76.26	5969.6328	6127.7584	5815.5876	75.15	74.93	3.1329	1.335	4.182364	9.814793878	1.782225					
						MEAN ERROR	tot x	tot y	tot x.y	tot x^2	tot y^2							Tot Dx.Dy	Tot Dx^2	Tot Dy^2			
Penilaian Ustaz	PITCH	VOLUME	RHYTHM	TIMBRE	OverallScore	3.7916974	1052.06	1048.95	78849.0717	79193.074	78656.126							A	23.4762	133.7712857	63.54735		
Responden 1	85	70	63	63	75.45	similarity	(tot x)^2	(tot y)^2	sqrt tot x^2	sqrt tot y^2							B	sqrt (tot Dx^2 * tot Dy^2)					
Responden 2	86	72	63	64	76.12	96.208303	1106830	1100296.1	281.41264	280.457							92.19984118						
Responden 3	89	69	65	60	78.3	avg x	avg y																
Responden 4	85	65	62	69	74.94	75.14714	74.925																
						HASIL KORELASI KEMIRIPAN						KETERANGAN											
						Tot Dx.Dy / sqrt (tot Dx^2 * tot Dy^2) or A/B						x						OverallScore aplikasi					
						HASIL 0.255 KORELASI LEMAH						y						OverallScore ustaz					
						ERROR RATE =						tot						Total					
						ABS(y-x) / y * 100																	

BAB 7 PENUTUP

Pada bagian penutup ini terdapat kesimpulan dan saran terhadap proses dan hasil dari skripsi. Di mana kesimpulan dan saran di tuliskan secara terpisah. Adapun kesimpulan yang didapatkan dan saran yang diberikan akan dijelaskan sebagai berikut:

7.1 Kesimpulan

Pada proses dan hasil penelitian ini dapat ditarik suatu kesimpulan, dimulai dari pendahuluan sampai tahap pengujian adalah sebagai berikut:

1. Hasil perubahan kebutuhan sistem dari penelitian sebelumnya yaitu adanya perubahan *User Interface* (UI) seperti tema / warna aplikasi, UI pada halaman latihan dalam mendengar cara baca huruf hijaiyah, dan juga mekanisme merekam jawaban dari soal bacaan huruf hijaiyah. Serta adanya penambahan fungsional baru yaitu menampilkan daftar riwayat skor pengerjaan soal telah diimplementasikan sesuai kebutuhan sistem baru.
2. Hasil setelah implementasi algoritma GMM untuk menentukan skor *timbre* yaitu dapat menghasilkan tingkat kemiripan antara kedua suara berdasarkan parameter *timbre* dan adanya kesalahan dalam penilaian *overall score* dapat diminimalisir, meskipun masih belum cukup sempurna.
3. Hasil pengujian korelasi kemiripan *overall score* setelah ditambahkan parameter *timbre* terjadi kenaikan dari korelasi kemiripan pada penelitian sebelumnya. Adapun hasil korelasi kemiripan penelitian ini, pada Iqra 1 sebesar 0,376 yang termasuk dalam kriteria sedang, pada Iqra 2 sebesar 0,795 yang termasuk dalam kriteria kuat, dan pada Iqra 3 sebesar 0,255 yang termasuk dalam kriteria lemah.

7.2 Saran

Saran-saran yang dapat diberikan, apabila terdapat pengembangan kembali terhadap Aplikasi *Scoring Iqra* ini adalah sebagai berikut:

1. Menambahkan Iqra jilid 4 sampai jilid 6 dengan memasukkan semua huruf hijaiyah ke dalam Aplikasi *Scoring Iqra*.
2. Menambah pilihan soal huruf hijaiyah di setiap jilid Iqra.
3. Meminimalisir kembali kesalahan penilaian *overall score* bacaan pengguna yang kurang sesuai dengan bacaan ustaz menggunakan algoritma lain.

DAFTAR REFERENSI

- Alexandra, J., 2017. *Agile Development Methods*. [Online] Available at: <https://sis.binus.ac.id/2017/05/08/agile-development-methods/#:~:text=Agile%20Development%20Methods%20adalah%20sekelompok,terhadap%20perubahan%20dalam%20bentuk%20apapun.> [Diakses 15 September 2020].
- Ambler, S. W., 2002. *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*. New York: Robert Ipsen.
- Budi, S., 2012. Upaya Peningkatan Kepekaan Nada dalam Tangga Nada Pentatonik dan Diatonik melalui Listening Songs pada Siswa SMP Negeri 2 Turi Sleman Yogyakarta. *eprints.uny.ac.id/8180/3/BAB 2-08208244022.pdf*, pp. 8-23.
- Chok, N. S., 2010. Pearson's Versus Spearman's And Kendall's Correlation Coefficients For Continuous Data. *University of Pittsburgh*, p. 4.
- Fajry, A. F., 2019. *Lima Metode Belajar Al-Quran yang Terkenal Di Indonesia*. [Online] Available at: <https://bincangsyariah.com/ubudiyah/lima-metode-belajar-al-quran-yang-terkenal-di-indonesia/> [Diakses 13 September 2020].
- Fauzan, A., Arwani, I. & Fanani, L., 2018. Pembangunan Aplikasi Iqro' Berbasis Android Menggunakan Google Speech. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 2, pp. 29-35.
- Fujihara, H. & Goto, M., 2007. A Music Information Retrieval System Based On Singing Voice Timbre. *ISMIR*, pp. 467-470.
- Hamidah, N., 2011. Upaya Peningkatan Kemampuan Membaca Al-Qur'an Dengan Metode Iqro' Pada Siswa Kelas III SD Negeri Kebumen 01 Kecamatan Banyubiru Tahun 2011.
- Hartono, S. O. E., 2017. Penggunaan Metode Al-Husna Untuk Peningkatan Kualitas Membaca Al-Qur'An Dalam Program Pengenalan Al-Qur'An di SMP Muhammadiyah 8 Surakarta Tahun Ajaran 2016/2017. pp. 1-15.
- Hermawan, 2020. *Perkembangan Android dari Masa ke Masa*. [Online] Available at: <https://www.tagar.id/perkembangan-android-dari-masa-ke-masa> [Diakses 15 September 2020].
- Hidayat, A., 2017. *Cara Hitung Rumus Slovin Besar Sampel*. [Online] Available at: <https://www.statistikian.com/2017/12/hitung-rumus-slovin-sampel.html> [Accessed 14 Juni 2021].
- Munir, A. & Sudarsono, 1994. *Ilmu Tajwid dan Seni Baca Al Qur'an*. Jakarta: Rineka Cipta.
- Nasser, G. A., Sunaryo, D. & Sarwosri, 2013. Rancang Bangun Aplikasi : Skoring Bacaan Al-Qur'an berdasarkan Kemiripan Teks Al-Qur'an menggunakan

Speech Recognition Engine dan Faktor Akustik pada Kasus Murattal Saad Al Ghamidi berbasis Android. pp. 2-4.

Nielsen, J., 2012. *How Many Test Users*. [Online] Available at: <https://www.nngroup.com/articles/how-many-test-users/> [Diakses 15 September 2020].

Nuari, R., F. Y. C. & Kusriani, 2019. IMPLEMENTASI METODE TOPSIS SISTEM PENDUKUNG KEPUTUSAN MENENTUKAN LOKASI TOKO PADA PERUSAHAAN RETAIL. *Jurnal INFORMA Politeknik Indonusa Surakarta*, Volume 5, pp. 36-44.

Nugraha, M. H., Brata, K. C. & Brata, A. H., 2019. Pengembangan Aplikasi Scoring Iqro' Berbasis Android Menggunakan Parameter Suara Pitch, Volume dan Rhythm. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 3, pp. 2646-2652.

Nur'aini, F. S., 2014. Konstruksi Makna Mengaji Dalam Program Acara Magrib Mengaji Di Radio Mq Fm Bandung.

Prayogi, Y. R., 2019. Modifikasi Metode MFCC untuk Identifikasi Pembicara di Lingkungan Ber-Noise. *JOINTECS*, Volume 4, pp. 13-16.

Pressman, R. S., 2009. *Software Engineering A Practitioner's Approach 7th Ed* - Roger S. Pressman.

Putra, R. B. D., Budi, E. S. & Kadafi, A. R., 2020. Perbandingan Antara SQLite, Room, dan RDBLiTe dalam Pembuatan Basis Data pada Aplikasi Android. *Jurnal Riset Komputer STMIK Nusa Mandiri*, Volume 7, pp. 376-381.

Putri, A. S. P. I., 2017. Kolaborasi Metode Iqro' Dan Metode Tatsmur Dalam Pembelajaran Membaca Al-Qur'an Bagi LANSIA (Studi TPQ Ummi Sholikhah di Kampung Bratan, Kelurahan Pajang, Kecamatan Laweyan). p. 22.

Ratner, B., 2009. The correlation coefficient: Its values range between + 1 / - 1, or do they ?. *Journal of Targeting, Measurement and Analysis for Marketing* - doi: 10.1057/jt.2009.5, Volume 17, p. 139 - 142.

Religia, Y., 2019. FEATURE EXTRACTION UNTUK KLASIFIKASI PENGENALAN WAJAH MENGGUNAKAN SUPPORT VECTOR MACHINE DAN K-NEAREST NEIGHBOR. *Pelita Teknologi: Jurnal Ilmiah Informatika*, pp. 85-92.

Rostianingsih, S., Adipranata, R. & Wibisono, F. S., 2008. ADAPTIVE BACKGROUND DENGAN METODE GAUSSIAN MIXTURE MODELS UNTUK REAL-TIME TRACKING. *Media Neliti*, pp. 68-77.

Saefullah, S., 2017. *ISLAMPOS*. [Online] Available at: <https://www.islampos.com/52697-52697/> [Accessed 3 Juni 2021].

Sasongko, A., 2019. *Keutamaan Mempelajari dan Mengajarkan Alquran*. [Online] Available at: <https://republika.co.id/berita/dunia-islam/islam->

nusantara/19/08/14/pw6nfnq313-keutamaan-mempelajari-dan-mengajarkan-alquran [Diakses 13 September 2020].

Srijatun, 2017. Implementasi Pembelajaran Baca Tulis Al-Qur'an Dengan Metode Iqro Pada Anak Usia Dini Di RA Perwanida Slawi Kabupaten Tegal. *Nadwa / Jurnal Pendidikan Islam*, Volume 1, pp. 25-42.

Statcounter, 2020. *Statcounter GlobalStats*. [Online] Available at: <https://gs.statcounter.com/os-market-share/mobile/indonesia> [Diakses 15 September 2020].

Sulistya, M., 2016. Metode Iqro' Terhadap Kemampuan Membaca Huruf Hijaiyyah Anak Autis. *JURNAL PENDIDIKAN KHUSUS*, pp. 1-10.

Takari, M., 2016. Karya Musik Dalam Konteks Seni Pertunjukan. www.researchgate.net - <https://doi.org/10.13140/RG.2.1.1095.4486>, p. 5.

Takayadi, H., 2012. Perancangan Dan Pembangunan Perangkat Lunak Skoring Kualitas Suara Pada Karaoke Player Berdasarkan Pitch Volume Dan Ritme Untuk Platform Android. *ITS-Undergraduate-51001120001655* - <http://digilib.its.ac.id/ITS-Undergraduate-51001120001655/22836>.

Tsai, W.-H., Member, IEEE & Lee, H.-C., 2012. Automatic Evaluation of Karaoke Singing Based on Pitch, Volume, and Rhythm Features. *IEEE Transactions on Audio, Speech and Language Processing*, Volume 20, pp. 1233-1243.

Wells, D., 1999. *Extreme Programming: A gentle introduction*. [Online] Available at: <http://www.extremeprogramming.org/> [Diakses 26 November 2020].

Wikipedia, 2020. *Wikipedia*. [Online] Available at: https://id.wikipedia.org/wiki/Alih_aksara_Arab-Latin [Diakses 23 Desember 2020].

LAMPIRAN A DOKUMENTASI FOTO

A.1 Ustaz sedang melakukan penilaian



A.2 Responden sedang menjawab soal bacaan huruf hijaiyah



Pandhu San Marendha
11 tahun



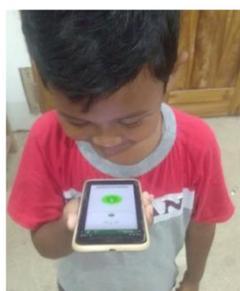
Alfilil Wiguna
10 tahun



Andrean Hehanusa Saputra
11 tahun



Angger Dimas Prasetyo
11 tahun



Krisna Yudha Prasetyo
9 tahun



Naufal Pratama
10 tahun



Nur Azizah Rahmah
15 tahun



Julaidah Najatul Rohmah
13 tahun



Wahyuningsih
14 tahun



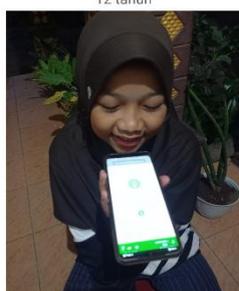
Wandha Marsha Az Zahra
12 tahun



Shafa Febriani Putri
8 tahun



Athaya Rahima Tiffany
10 tahun



Ayatul Khusna
11 tahun



Defi Novita Andriyani
8 tahun

A.3 Form penilaian *pitch*, *volume*, *rhythm*, dan *timbre*

PENILAIAN BACAAN RESPONDEN MENGGUNAKAN APLIKASI

Penilaian bacaan iqra 1 konten آت oleh Aplikasi

RESPONDEN	PITCH	VOLUME	RHYTHM	TIMBRE	OVERALL SCORE
Responden 1	90	62	51	37	72,72
Responden 2	92	61	55	25	75,05
Responden 3	95	62	47	29	73,79
Responden 4	81	61	56	35	69,58
Responden 5	86	62	79	38	69,78
Responden 6	76	63	63	32	69,53
Responden 7	78	62	59	58	69,58
Responden 8	93	62	50	23	73,71
Responden 9	73	64	65	58	69,21
Responden 10	86	64	57	65	73,46
Responden 11	81	63	63	27	72,56
Responden 12	82	61	58	22	70,63
Responden 13	93	64	53	39	75,27
Responden 14	87	60	57	50	73,51

Penilaian bacaan iqra 2 konten ثت oleh Aplikasi

RESPONDEN	PITCH	VOLUME	RHYTHM	TIMBRE	OVERALL SCORE
Responden 1	91	62	68	37	79,73
Responden 2	93	64	57	41	76,83
Responden 3	97	62	41	28	72,59
Responden 4	84	62	59	61	72,94
Responden 5	98	62	51	25	76,88
Responden 6	81	64	58	67	76,63
Responden 7	92	64	58	27	76,38
Responden 8	98	61	49	49	76,55
Responden 9	84	63	63	33	73,95
Responden 10	93	65	48	19	73,02
Responden 11	74	61	62	27	67,85
Responden 12	81	63	45	23	65,26
Responden 13	91	61	53	29	73,82
Responden 14	72	62	54	28	63,78

Penilaian bacaan iqra 3 konten جتن oleh Aplikasi

RESPONDEN	PITCH	VOLUME	RHYTHM	TIMBRE	OVERALL SCORE
Responden 1	87	62	59	19	71,85
Responden 2	94	62	62	31	78,98
Responden 3	93	63	50	31	73,92
Responden 4	91	62	59	62	74,91
Responden 5	86	63	58	53	73,55
Responden 6	89	61	62	30	76,16
Responden 7	86	62	49	60	70,12
Responden 8	98	63	57	19	79,09
Responden 9	91	61	59	26	76,04
Responden 10	93	61	61	59	78,56
Responden 11	93	63	58	37	77,08
Responden 12	90	62	41	46	69,1
Responden 13	88	66	57	48	79,32
Responden 14	93	63	61	40	78,28

Keterangan:

- *Pitch* adalah Tinggi / rendahnya nada pada suara.
- *Volume* adalah Keras / lembutnya pada suara.
- *Rhythm* adalah Irama / tempo pada suara.
- *Timbre* adalah Kualitas warna nada pada suara.
- Penilaian pengucapan bacaan iqra ditulis dalam skala 0 – 100.
- *Overall Score* pada penilaian aplikasi akan dihitung secara otomatis sesuai rumus.
- *Overall Score* pada penilaian ustaz akan dihitung secara manual sesuai rumus.

PENILAIAN BACAAN RESPONDEN OLEH USTAZ

Penilaian bacaan iqra 1 konten آت oleh Ustaz

RESPONDEN	PITCH	VOLUME	RHYTHM	TIMBRE	OVERALL SCORE
Responden 1	85	75	70	72	79,79
Responden 2	87	72	70	65	79,35
Responden 3	81	73	65	73	79,36
Responden 4	80	70	61	60	71,88
Responden 5	78	72	60	60	70,5
Responden 6	81	70	70	70	76,05
Responden 7	80	73	69	78	75,43
Responden 8	80	65	65	60	73,15
Responden 9	82	68	69	65	76,02
Responden 10	83	69	68	68	76,3
Responden 11	79	72	70	69	75,03
Responden 12	80	70	68	71	74,76
Responden 13	80	74	67	70	74,56
Responden 14	83	73	69	60	76,72

Penilaian bacaan iqra 2 konten ثت oleh Ustaz

RESPONDEN	PITCH	VOLUME	RHYTHM	TIMBRE	OVERALL SCORE
Responden 1	85	60	70	68	77,71
Responden 2	86	68	72	64	79,39
Responden 3	84	69	69	62	77,11
Responden 4	80	60	68	65	79,19
Responden 5	86	70	65	70	76,9
Responden 6	84	70	62	72	79,7
Responden 7	82	72	70	68	76,66
Responden 8	84	70	75	72	79,69
Responden 9	84	64	70	71	77,92
Responden 10	79	68	65	70	72,95
Responden 11	78	65	62	69	71,09
Responden 12	80	60	65	60	72,9
Responden 13	85	70	62	63	75,07
Responden 14	80	63	60	62	71,19

Penilaian bacaan iqra 3 konten جتن oleh Ustaz

RESPONDEN	PITCH	VOLUME	RHYTHM	TIMBRE	OVERALL SCORE
Responden 1	85	70	63	63	75,45
Responden 2	86	72	63	64	76,12
Responden 3	89	69	65	60	78,3
Responden 4	85	65	62	69	74,94
Responden 5	80	66	65	70	73,4
Responden 6	82	66	60	69	72,58
Responden 7	78	70	69	71	74,04
Responden 8	80	60	65	66	73,02
Responden 9	79	69	65	63	72,86
Responden 10	86	70	60	70	75
Responden 11	84	70	70	66	77,62
Responden 12	80	60	60	68	71,16
Responden 13	85	69	70	70	78,2
Responden 14	85	65	65	78	76,26

Keterangan:

- Pitch adalah Tinggi / rendahnya nada pada suara.
- Volume adalah Keras / lembutnya pada suara.
- Rhythm adalah Irama / tempo pada suara.
- Timbre adalah Kualitas warna nada pada suara.
- Penilaian pengucapan bacaan iqra ditulis dalam skala 0 – 100.
- Overall Score pada penilaian aplikasi akan dihitung secara otomatis sesuai rumus.
- Overall Score pada penilaian ustaz akan dihitung secara manual sesuai rumus.

A.4 Error rate

Penilaian Apps	ERROR RATE PADA HASIL OVERALL SCORE IQRA JILID 1											Penilaian Ustaz
	PITCH	VOLUME	RHYTHM	TIMBRE	X	ERROR RATE	Y	PITCH	VOLUME	RHYTHM	TIMBRE	
Responden 1	90	62	51	37	72.72	2.702702703	74.74	85	75	60	72	Responden 1
Responden 2	92	61	55	25	75.05	5.419029616	79.35	87	72	70	65	Responden 2
Responden 3	95	62	47	29	73.79	0.766541151	74.36	81	73	65	73	Responden 3
Responden 4	81	61	56	35	69.58	3.199777407	71.88	80	70	61	60	Responden 4
Responden 5	86	62	49	38	69.78	1.021276596	70.5	78	72	60	60	Responden 5
Responden 6	76	63	63	32	69.53	8.573307035	76.05	81	70	70	70	Responden 6
Responden 7	78	62	59	58	69.58	7.75534933	75.43	80	73	69	78	Responden 7
Responden 8	93	62	50	23	73.71	0.765550239	73.15	80	65	65	60	Responden 8
Responden 9	73	64	65	58	69.21	8.958168903	76.02	82	68	69	65	Responden 9
Responden 10	86	64	57	65	73.46	3.72214941	76.3	83	69	68	68	Responden 10
Responden 11	81	63	64	27	72.56	3.292016527	75.03	79	72	70	69	Responden 11
Responden 12	82	61	58	22	70.63	5.524344569	74.76	80	70	68	71	Responden 12
Responden 13	93	64	53	39	75.27	0.952253219	74.56	80	74	67	70	Responden 13
Responden 14	87	60	57	50	73.51	4.184045881	76.72	83	73	69	60	Responden 14
						MEAN ERROR						
						ERROR RATE = ABS(Y-X) / Y * 100						ERROR RATE = ABS(Y-X) / Y * 100
						4.059764156						
						SIMILARITY						
						95.94023584						

Penilaian Apps	ERROR RATE PADA HASIL OVERALL SCORE IQRA JILID 2											Penilaian Ustaz
	PITCH	VOLUME	RHYTHM	TIMBRE	X	ERROR RATE	Y	PITCH	VOLUME	RHYTHM	TIMBRE	
Responden 1	91	62	68	37	79.73	2.599408056	77.71	85	60	70	68	Responden 1
Responden 2	93	64	57	41	76.83	3.163599698	79.34	86	68	72	64	Responden 2
Responden 3	97	62	41	28	72.59	5.861755933	77.11	84	69	69	62	Responden 3
Responden 4	84	62	59	61	72.94	1.618559482	74.14	80	60	68	65	Responden 4
Responden 5	98	62	51	25	76.88	0.026007802	76.9	86	70	65	70	Responden 5
Responden 6	91	64	58	67	76.63	2.583668005	74.7	84	70	62	72	Responden 6
Responden 7	92	64	58	27	76.38	0.365249152	76.66	82	72	70	68	Responden 7
Responden 8	98	61	49	49	76.55	3.879959819	79.64	84	70	75	72	Responden 8
Responden 9	84	63	63	33	73.95	4.482045983	77.42	84	64	70	71	Responden 9
Responden 10	93	65	48	19	73.02	0.095956134	72.95	79	68	65	70	Responden 10
Responden 11	74	61	62	27	67.85	4.557603038	71.09	78	65	62	69	Responden 11
Responden 12	81	63	45	23	65.26	10.48010974	72.9	80	60	65	60	Responden 12
Responden 13	91	61	53	29	73.82	1.665112562	75.07	85	70	62	63	Responden 13
Responden 14	72	62	54	28	63.78	10.40876528	71.19	80	63	60	62	Responden 14
						MEAN ERROR						
						ERROR RATE = ABS(Y-X) / Y * 100						ERROR RATE = ABS(Y-X) / Y * 100
						3.69912862						
						SIMILARITY						
						96.30087138						

Penilaian Apps	ERROR RATE PADA HASIL OVERALL SCORE IQRA JILID 3											Penilaian Ustaz
	PITCH	VOLUME	RHYTHM	TIMBRE	X	ERROR RATE	Y	PITCH	VOLUME	RHYTHM	TIMBRE	
Responden 1	87	62	54	19	71.85	4.771371769	75.45	85	70	63	63	Responden 1
Responden 2	94	62	62	31	78.98	3.757225434	76.12	86	72	63	64	Responden 2
Responden 3	93	63	50	31	73.92	5.593869732	78.3	89	69	65	60	Responden 3
Responden 4	91	62	54	62	74.91	0.040032026	74.94	85	65	62	69	Responden 4
Responden 5	86	63	58	53	73.55	0.204359673	73.4	80	66	65	70	Responden 5
Responden 6	89	61	62	30	76.16	4.932488289	72.58	82	66	60	69	Responden 6
Responden 7	86	62	49	60	70.22	5.159373312	74.04	78	70	69	71	Responden 7
Responden 8	98	63	57	19	79.09	8.312791016	73.02	80	60	65	66	Responden 8
Responden 9	91	61	59	26	76.04	4.364534724	72.86	79	69	65	63	Responden 9
Responden 10	93	61	61	59	78.56	4.746666667	75	86	70	60	70	Responden 10
Responden 11	93	63	58	37	77.08	0.695696985	77.62	84	70	70	66	Responden 11
Responden 12	90	62	41	46	69.1	2.894884767	71.16	80	60	60	68	Responden 12
Responden 13	88	66	57	48	74.32	4.961636829	78.2	85	69	70	70	Responden 13
Responden 14	93	63	61	40	78.28	2.64883294	76.26	85	65	65	78	Responden 14
						MEAN ERROR						
						ERROR RATE = ABS(Y-X) / Y * 100						ERROR RATE = ABS(Y-X) / Y * 100
						3.79169744						
						SIMILARITY						
						96.20830256						