Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB III

ANALISIS DAN PERANCANGAN

3.1 Metode Penelitian

Perancangan aplikasi *Audio Detection* atau *AudiTion* ini dilakukan dengan perancangan aplikasi, dimulai dari desain tampilan, pengelompokan *database* suara, pencocokan suara yang terdeteksi dengan suara pada *database* dengan menggunakan algoritma AdaBoost, dan pengujian aplikasi. Untuk mencapai tujuan penelitian, dilakukan langkah-langkah sebagai berikut,

3.1.1 Studi Literatur

Penelitian diawali dengan pengumpulan materi yang mendukung seperti paper atau jurnal mengenai tuna rungu, *Artificial Intelligence*, *Machine learning*, dan pemrograman Android. Beberapa sumber informasi yang diperoleh dari internet juga membantu penulis dalam proses pemrograman maupun pemecahan masalah.

3.1.2 Perancangan dan Pengembangan Aplikasi *AudiTion*

Aplikasi *AudiTion* dikembangkan dengan menggunakan bahasa pemrograman Java sesuai dengan bahasa pemrograman untuk *developing* aplikasi pada sistem operasi Android. Pada tahap awal perancangan,

Audition Aplikasi..., Ayu Indah Shekar Melati, FTI UMN, 2013

dirancanglah tampilan untuk aplikasi AudiTion, *AudiTion* akan menampilkan *chart* dari sinyal suara yang terdeteksi dan nama dari suara tersebut. Sebagai contoh, ketika pengguna mendeteksi suara bel pintu maka aplikasi akan memroses suara tersebut, menampilkan grafik dari suara bel pintu dan nama suara, yaitu "door_bell".

Untuk *database* suara yang terdapat dalam aplikasi *AudiTion*, dipilihlah 23 suara (John Tracy, 2000) yang merupakan suara yang terdengar dalam kehidupan sehari-hari, dalam memudahkan pengelompokan *database* suara dibagi dalam dua kelompok suara, yaitu *indoor-sounds* (suara yang umum terdengar di dalam ruangan) dan *outdoor-sounds* (suara yang umum terdengar di luar ruangan). Setelah dikelompokan masing-masing suara diekstraksi dengan menggunakan aplikasi jAudio, untuk mendapatkan fitur suara sebagai perbandingan dengan suara yang terdeteksi.

Selanjutnya perancangan dilakukan dengan melakukan *train*ing pada *training set* (*database* suara) dan membangun *test set* (suara yang terdeteksi) untuk memprediksi suara. Pengklasifikasian akan menggunakan algoritma AdaBoost untuk memprediksi suara yang terdeteksi.

3.1.3 Pengujian Aplikasi *AudiTion*

Aplikasi AudiTion diuji dengan membandingkan hasil prediksi dari proses pendeteksian suara-suara yang terdapat dalam *database*. Suara dideteksi dengan beberapa kondisi yaitu,

- Suasana lingkungan saat pendeteksian suara benar-benar hening
- Suasana lingkungan saat pendeteksian suara bising/ada keramaian
- Jarak *handphone* dekat dengan sumber suara
- Jarak *handphone* jauh dengan sumber suara

3.1.4 Dokumentasi

Dokumentasi dilakukan dengan penulisan keterangan pada program aplikasi *AudiTion*, berupa penjelasan fungsi dan variabel yang digunakan. Dokumentasi diperlukan agar aplikasi yang telah dirancang dapat dengan mudah untuk dimengerti oleh peneliti lain yang ingin melanjutkan penelitian.

3.2 Analisis

Pada penelitian ini, algoritma yang digunakan adalah algoritma AdaBoostM1, Bagging tidak digunakan karena proses *building training set* untuk kedua database sangat lama, sehingga yang dibandingkan hanya *weak learner,* yaitu *Decision Stump* dan REPTree. Berikut adalah hail dari

masing-masing pengujian *weak learner.* Hasil klasifikasi AdaBoostM1 dengan menggunakan *weak learner* Decision Stump terdapat dalam Tabel 3.1 untuk *database indoor sounds* dan Tabel 3.2 untuk *database outdoor sounds*.

Tabel 3.1 Hasil klasifikasi training set database indoor-sounds menggunakan algoritma AdaBoostM1 dengan weak learner Decision Stump

a	b	c	d	e	f	g	h	i	j	k	l			classified as
2089	0	0	0	0	0	0	485	0	0	0	0	a	=	baby_crying
0	0	0	0	0	0	0	34	0	0	0	0	b	=	beep
816	0	0	0	0	0	0	111	0	0	0	0	c	=	broom_sweeps
843	0	0	0	0	0	0	40	0	0	0	0	d	=	door_creaking
122	0	0	0	0	0	0	19	0	0	0	0	e	=	door_slam
705	0	0	0	0	0	0	43	0	0	0	0	f	=	door_bell
854	0	0	0	0	0	0	43	0	0	0	0	g	=	footstep
152	0	0	0	0	0	0	2174	0	0	0	0	h	=	hairdryer
173	0	0	0	0	0	0	41	0	0	0	0	i	=	knock
1292	0	0	0	0	0	0	215	0	0	0	0	j	=	ringing
256	0	0	0	0	0	0	748	0	0	0	0	k	=	water_runs
191	0	0	0	0	0	0	116	0	0	0	0	l	=	whistle

Tabel 3.2 Hasil klasifikasi training set database outdoor-sounds menggunakan algoritma AdaBoostM1 dengan weak learner Decision Stump

a	b	c	d	e	f	g	h	i	j	k			classified as
0	0	0	0	380	0	0	0	0	1440	0	a	=	airplanes
0	0	0	0	207	0	0	0	0	1057	0	b	=	applause
0	0	0	0	123	0	0	0	0	464	0	c	=	birds_chirp
0	0	0	0	308	0	0	0	0	120	0	d	=	car_honk
0	0	0	0	1867	0	0	0	0	480	0	e	=	crowded
0	0	0	0	116	0	0	0	0	237	0	f	=	dog_bark
0	0	0	0	201	0	0	0	0	1901	0	g	=	engine_start
0	0	0	0	138	0	0	0	0	195	0	h	=	screaming
0	0	0	0	240	0	0	0	0	574	0	i	=	thunder
0	0	0	0	240	0	0	0	0	2101	0	j	=	train
0	0	0	0	794	0	0	0	0	46	0	k	=	wind_blowing

Klasifikasi AdaBoostM1 dengan menggunakan *weak learner* Decision Stump mengklasifikasi suara yang terdapat pada *training set* menjadi dua jenis suara. Untuk *indoor*-sounds diklasifikasi menjadi suara tangisan bayi (*baby crying)* dan pengering rambut *(hairdryer)*, sedangkan

untuk *outdoor*-sounds diklasifikasi menjadi suara keramaian *(crowded)* dan suara kereta api (*train).*

Algoritma AdaBoostM1 dengan *weak learner* REPTree mengklasifikasi *training set* dengan membangun sebuah *decision tree*, contoh *decision tree* REPTree untuk *database indoor sounds* dengan jumlah instance 865 dapat dilihat pada Lampiran 2, sehingga setiap instances dari *training set*, *indoor*-sounds maupun *outdoor*-sounds *database*, ditampilkan dan dapat dijadikan dasar untuk klasifikasi. Tabel 3.3 menyajikan data hasil klasifikasi untuk *database indoor sounds* dan Tabel 3.4 untuk *database outdoor sounds.*

Tabel 3.3 Hasil klasifikasi training set database indoor-sounds dengan menggunakan algoritma AdaBoostM1 dengan weak learner REPTree

a	b	c	d	e	f	g	h	i	j	k	l			classified as
2326	0	16	25	0	27	48	44	1	20	40	27	**a**	=	baby_crying
2	32	0	0	0	0	0	0	0	0	0	0	**b**	=	beep
29	0	814	7	3	7	36	2	2	22	4	1	**c**	=	broom_sweeps
64	0	7	749	3	11	28	0	8	5	4	4	**d**	=	door_creaking
4	0	6	9	83	3	15	1	17	3	0	0	**e**	=	door_slam
14	0	6	2	1	704	8	2	5	0	5	1	**f**	=	door_bell
112	0	31	22	4	9	666	8	10	14	17	4	**g**	=	footstep
95	0	2	5	1	1	10	2188	0	3	20	1	**h**	=	hairdryer
2	0	14	4	17	1	17	1	148	10	0	0	**i**	=	knock
17	0	25	2	5	7	15	1	6	1425	4	0	**j**	=	ringing
91	1	1	17	1	0	3	13	0	7	867	3	**k**	=	water_runs
42	0	2	9	1	2	4	1	0	0	0	246	**l**	=	whistle

Tabel 3.4 Hasil klasifikasi training set database outdoor-sounds dengan menggunakan algoritma AdaBoostM1 dengan weak learner REPTree

a	b	c	d	e	f	g	h	i	j	k			classified as
1394	66	9	1	27	2	48	0	52	206	15	**a**	=	airplanes
27	1113	23	2	14	2	20	5	6	52	0	**b**	=	applause
5	76	424	5	14	3	31	9	1	18	1	**c**	=	birds_chirp
10	0	12	326	20	1	7	6	3	30	13	**d**	=	car_honk
33	48	14	24	2031	10	36	5	9	92	45	**e**	=	crowded
14	8	9	11	85	121	15	1	3	68	18	**f**	=	dog_bark
95	40	13	3	72	6	1687	3	16	152	15	**g**	=	engine_start
3	38	18	6	21	1	8	205	1	32	0	**h**	=	screaming
81	1	4	6	24	4	31	0	501	139	23	**i**	=	thunder
179	39	26	9	106	7	85	10	38	1827	15	**j**	=	train
48	7	0	3	116	1	6	8	10	7	634	**k**	=	wind_blowing

Sehingga pada penelitian ini, algoritma yang digunakan adalah algoritma AdaBoostM1 dengan menggunakan REPTree sebagai *weak learner*. Presentase tingkat ketepatan klasifikasi algoritma AdaBoostM1 untuk masing-masing *weak learner* yaitu Decision Stump dan REPTree, ditampilkan dalam tabel 3.5,

Tabel 3.5 Presentase ketepatan klasifikasi training set

	Indoor sounds		*Outdoor sounds*	
	Correctly	Incorrectly	Correctly	Incorrectly
Decision Stump	36.8708% (4263 instances)	63.1292% (7299 instances)	29.9947% (3968 instances)	70.0053 % (9261 instances)
REPTree	88.6352% (10248 instances)	11.3648 % (1314 instances)	77.5796 % (10263 instances)	22.4204 % (2966 instances)

Sebagai modifikasi dari database *indoor sounds* dipilih lima suara. Kelima suara tersebut adalah suara tangisan bayi, ketukan pintu, bel pintu, dering telepon, dan peluit.

Presentase tingkat akurasi untuk *database indoor sounds* dengan hanya lima suara menggunakan *weak learner* Decision dan REPTree ditampilkan dalam tabel 3.6,

Tabel 3.6 Presentase ketepatan klasifikasi training set baru untuk database indoor sounds

	Correctly		Incorrectly	
	5 Sounds (5350)	5 Sounds (865)	5 Sounds (5350)	5 Sounds (865)
Decision Stump	63.6262 % (3404 instances)	45.0867 % (390 instances)	36.3738 % (1946 instances)	54.9133 % (475 instances)
REPTree	97.4766 % (5215 instances)	98.0347 % (848 instances)	2.5234 % (135 instances)	1.9653 % (17 instances)

Pengurangan jumlah instances untuk *database indoor sounds* dengan lima suara bertujuan untuk melihat apakah jumlah instances yang lebih sedikit berpengaruh pada tingkat akurasi pendeteksian suara pada saat proses pengujian.

Aplikasi Android dibuat dengan menggunakan Android Development Tools Bundle (Eclipse Integrated Development Environment dan Android SDK), menggunakan bahasa pemrograman Java. Versi Java Development Kit (JDK) yang digunakan adalah versi 6 untuk komputer 32-bit. Ekstraksi fitur suara untuk suara-suara yang akan dimasukan dalam *database* digunakan aplikasi jAudio versi 1.0.4. Algoritma AdaBoost serta REPTree sebagai *weak learner* terdapat pada WEKA *Library*.Untuk

pengujian awal pada desktop komputer digunakan WEKA GUI versi 3-8, sedangkan versi yang digunakan untuk Android adalah WEKAStripped.

3.3 Perancangan

Rancang bangun aplikasi AudiTion secara umum dibagi dalam tiga tahap besar yaitu, Penyusunan *Database* Suara, System Flow AudiTion, dan Sketsa Antarmuka Aplikasi.

3.3.1 Penyusunan *Database* Suara

Suara-suara untuk *database* diambil dari *library sound effect* yang berada di internet, untuk setiap jenis suara diambil lima contoh suara. Keseluruhan jenis suara ada 23 suara yang berada di sekitar manusia dan dapat didengar sehari-hari (John Tracy, 2000). Dalam memudahkan penyusunan *database* ke-23 suara tersebut dibagi menjadi dua kelompok suara, yaitu *indoor sounds* dengan 12 suara dan *outdoor sounds* dengan 11 suara. Berikut adalah daftar jenis suara dari masing-masing kelompok,

a. *Indoor*-sounds

- Suara tangisan bayi (*baby crying*)
- Suara *beep*
- Suara orang menyapu (*broom sweeps*)
- Suara pintu terbuka (*door creaking*)
- Suara pintu dibanting (*door slam*)
- Suara bel pintu (*door bell*)

- Suara langkah kaki (*foot step*)
- Suara pengering rambut (*hairdryer*)
- Suara pintu diketuk (*knock*ing door)
- Suara telepon bordering (*ringing*)
- Suara hujan/air mengalir (*water runs*)
- Suara peluit (*whistle*)

b. *Outdoor*-sounds

- Suara pesawat terbang (*airplanes*)
- Suara tepuk tangan (*applause*)
- Suara burung berkicau (*birds chirp*)
- Suara klakson kendaraan (*car honk*)
- Suara keramaian orang (*crowded*)
- Suara anjing menggonggong (*dog bark*)
- Suara mesin kendaraan saat dinyalakan (*engine start*)
- Suara teriakan (*screaming*)
- Suara petir (*thunder*)
- Suara kereta api (*train*)
- Suara angin berhembus (*wind blowing*)

Masing-masing suara dari setiap jenis suara diekstraksi dengan menggunakan sebuah aplikasi untuk ekstraksi suara yaitu jAudio. *Screenshot* tampilan aplikasi jAudio dapat dilihat pada Lampiran 2. Untuk suara pada database, fitur ekstraksi suara yang

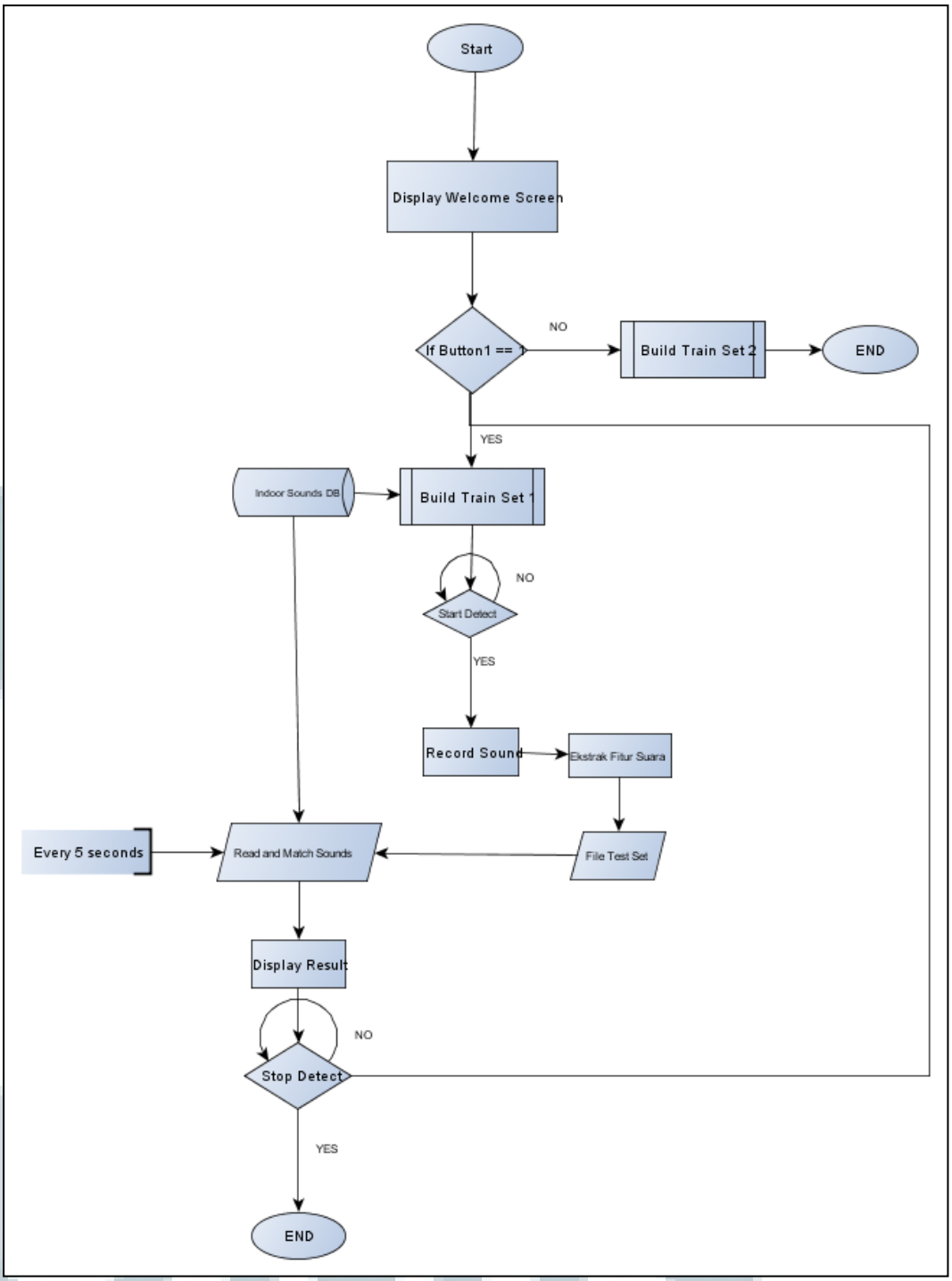
digunakan adalah *Strongest Frequency via FFT Maksimum, Root Mean Square,* dan *Zero Crossing*. Pemilihan ketiga fitur tersebut disesuaikan dengan hasil *FFT Based* yang digunakan saat mengekstrak fitur suara yang terdeteksi. Dari hasil pehitungan dengan menggunakan *FFT Based* hasil yang didapat hanya tiga yaitu, frekuensi, *pitch* dan *level* suara.

3.3.2 System Flow AudiTion

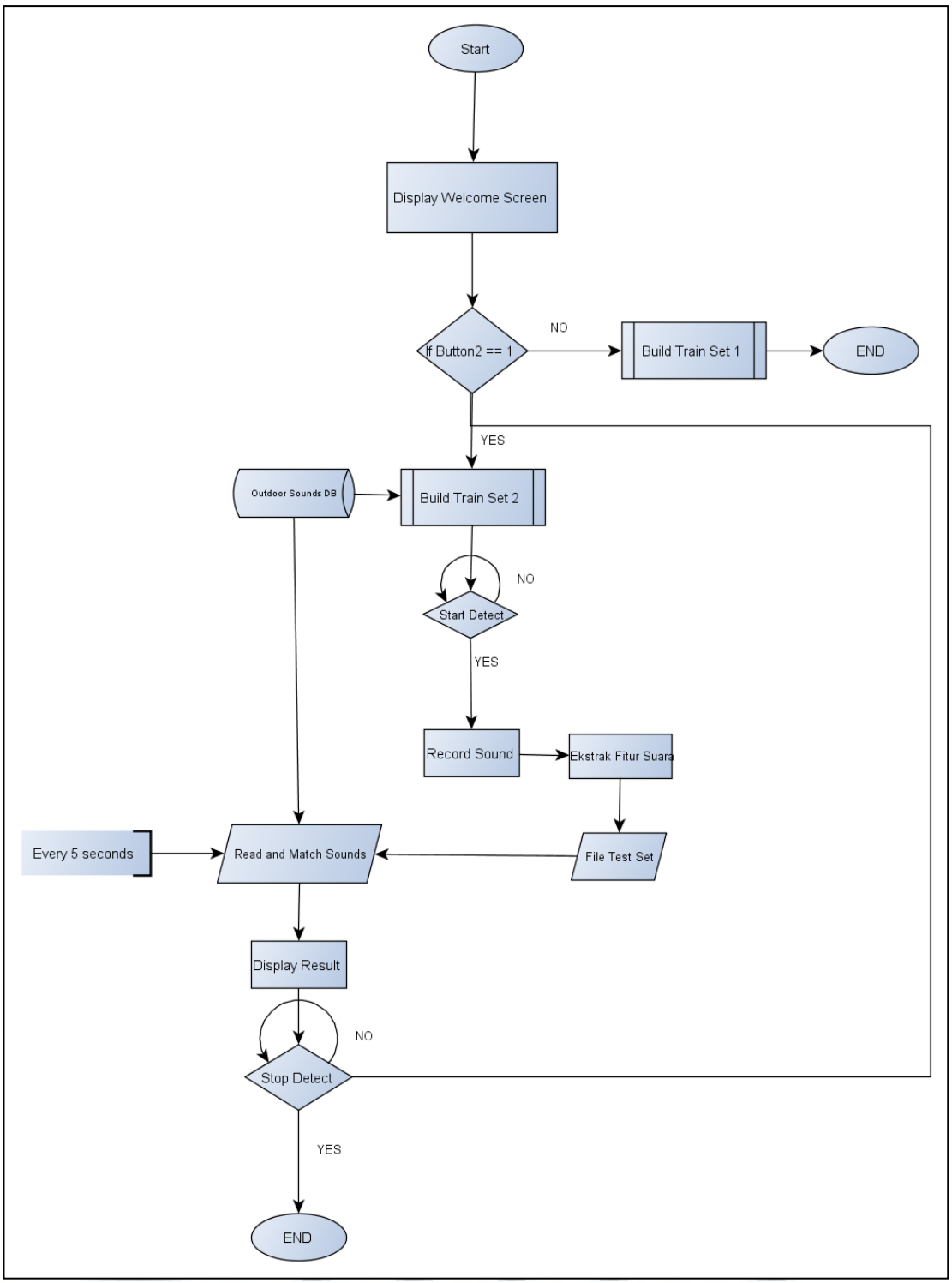
Pembangunan aplikasi AudiTion dimulai dengan pembuatan alur sistem aplikasi AudiTion. Sistem aplikasi AudiTion pertama-tama menampilkan halaman *welcome screen* aplikasi. Pada halaman *welcome screen* terdapat dua pilihan untuk memilih suara dengan kondisi di dalam ruangan (*indoor*) atau di luar ruangan (*outdoor*) yang ingin di deteksi. Kemudian sistem akan membangun *training set* berdasarkan *database* yang sesuai dengan kondisi yang telah dipilih sebelum proses pendeteksian dimulai. Setelah *training set* dibangun, aplikasi akan mulai mendeteksi suara setelah tombol *start* dipilih, dan akan berhenti setelah menekan tombol *stop*.

Saat tombol *start* ditekan, aplikasi akan merekam suara yang berada disekitar, mengekstraksi suara tersebut untuk mendapatkan nilai frekuensi, nilai *peak* tertinggi frekuensi dan nilai *magnitude* suara. Ketiga fitur tersebut akan langsung disimpan dalam sebuah file sebagai *test set* yang kemudian

langsung dicocokan dengan *training set* yang sudah dibangun sebelumnya. Saat aplikasi mendeteksi suara *chart* frekuensi dan level suara akan ditampilkan. Setiap lima detik suara akan dicocokan dan hasil prediksi akan ditampilkan. Saat tombol *stop* ditekan maka semua proses akan terhenti. Secara keseluruhan sistem alir aplikasi AudiTion digambarkan pada Gambar 3.1 dan 3.2.

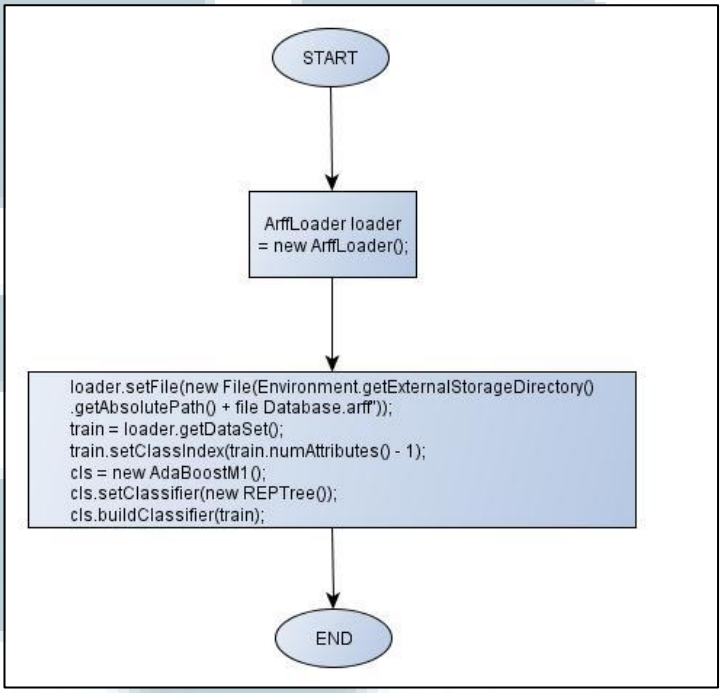


Gambar 3.1 System Flow aplikasi AudiTion database Indoor sounds



Gambar 3.2 System Flow sistem aplikasi AudiTion database Outdoor sounds

Saat proses *build training set* berlangsung aplikasi akan memanggil *classifier* dalam WEKA *library* yaitu *classifier* AdaBoostM1. Diagram alir saat proses *building training set* berlangsung ditampilkan pada Gambar 3.3.



Gambar 3.3 Flowchart proses building training set aplikasi AudiTion

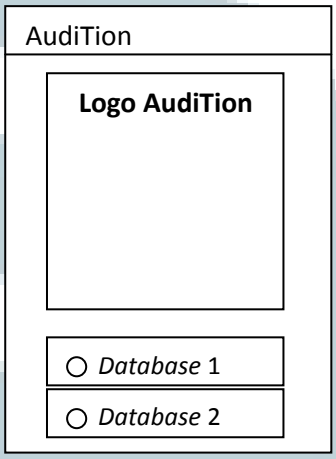
3.3.3 Sketsa Antarmuka Aplikasi

Pada subbab berikut ditampilkan sketsa antarmuka dari aplikasi Android yang dibuat,

1. *Welcome screen*

   Halaman awal dari aplikasi AudiTion merupakan sebuah *welcome screen* yang berisi logo dari AudiTion dan radio *button* untuk memilih *database* mana yang akan digunakan (*indoor*-sounds *database* atau *outdoor*-sounds *database*).

Sketsa tampilan untuk halaman *welcome screen* ditampilkan pada Gambar 3.4.



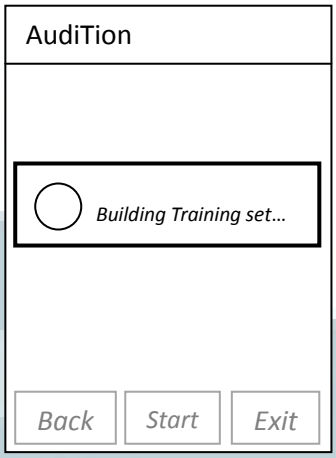
Gambar 3.4 Sketsa tampilan welcome screen AudiTion

2. Halaman Hasil

Halaman hasil dari aplikasi AudiTion terdiri dari tiga *button*, yaitu *button Start Detect* (yang otomatis akan berubah menjadi *Stop Detect* setelah diklik), *button Back* untuk kembali ke halaman *welcome screen*, dan *button Exit* untuk keluar dari aplikasi.

a. *Building Training set*

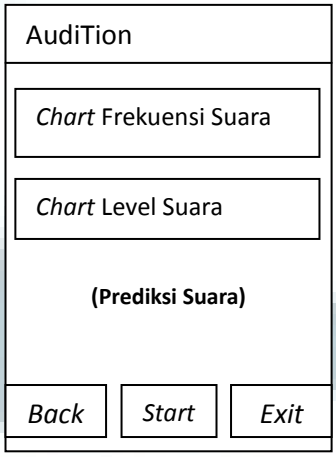
Halaman ini merupakan bagian dari halaman hasil setelah *welcome screen*, dengan sebuah *progress status* untuk mengindikasikan bahwa *training set* sedang dibangun. Sketsa tampilan saat proses building set berlangsung ditampilkan pada Gambar 3.5.



Gambar 3.5 Sketsa tampilan halaman hasil saat proses building training set

b. Deteksi dan Prediksi Suara

Proses pendeteksian suara dapat dilakukan setelah *training set* selesai dibangun dengan menekan tombol *Start Detect*. Selama mendeteksi, *chart* frekuensi dan level suara akan ditampilkan, selain itu setiap lima detik hasil prediksi juga akan ditampilkan. Sketsa proses pendeteksian suara ditapilkan pada Gambar 3.6.



Gambar 3.6 Sketsa tampilan halaman hasil saat proses pendeteksian suara

Semua proses aplikasi, pendeteksian suara dan pencocokan suara untuk mendapatkan hasil prediksi, akan terhenti setelah *button stop* ditekan.

UMN