

# Penentuan Arah Sumber Suara dengan Metode *Interaural Time Difference* Menggunakan Mikrokontroler STM32F4

Mohamad Asfari dan Muhammad Rivai, Tasripan

Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember

*e-mail*: muhammad\_rivai@ee.its.ac.id

**Abstrak**—Sistem pendengaran manusia dapat mengestimasi lokasi sumber suara dari pengamatan sinyal suara binaural dengan akurasi yang cukup tinggi. Manusia dapat mengenali lokasi suara secara 3 dimensi dan bahkan dapat mengenali keberadaan sumber suara pada posisi belakang dimana tidak terdapat informasi visual. Sudah banyak dilakukan penelitian mengenai pengenalan lokasi sumber suara menggunakan beberapa deret mikrofon yang memakai lebih dari 3 buah mikrofon, akan tetapi penelitian sistem pengenalan lokasi sumber suara yang menggunakan 2 buah mikrofon seperti pendengaran manusia masih jarang dilakukan. Pada penelitian ini merancang suatu sistem pendeteksi sumber suara dengan menggunakan 2 buah mikrofon dan mikrokontroler STM32F4 sebagai prosesor utama. Metode *Interaural Time Difference* digunakan untuk menentukan arah dengan cara membandingkan perbedaan waktu sampai sebuah suara yang dipancarkan oleh sumber suara. Berdasarkan hasil pengujian, sistem ini dapat menentukan arah sumber suara sebesar  $180^\circ$ . Semakin besar jarak antar mikropon dapat menghasilkan kesalahan sudut yang semakin kecil. Diharapkan alat ini dapat diaplikasikan dalam sistem pengusir hama burung untuk menentukan arah suara pengusir.

**Kata Kunci**—Arah Sumber Suara, *Interaural Time Difference*, STM32F4.

## I. PENDAHULUAN

SISTEM pendengaran manusia dapat mengestimasi lokasi sumber suara dari pengamatan sinyal suara binaural dengan akurasi yang cukup. Manusia dapat mengenali lokasi suara secara 3 dimensi dan bahkan dapat mengenali keberadaan sumber suara pada posisi belakang dimana tidak terdapat informasi visual. Sudah banyak dilakukan penelitian mengenai pengenalan lokasi sumber suara menggunakan beberapa *array* mikrofon yang memakai lebih dari 3 buah mikrofon, akan tetapi penelitian sistem pengenalan lokasi sumber suara yang menggunakan 2 buah mikrofon seperti pendengaran manusia masih jarang dilakukan [1].

Penelitian pada sistem lokalisasi suara pada manusia sudah dilakukan selama beberapa dekade dan mekanisme dari lokalisasi sumber suara pada manusia sudah diketahui. Dari penelitian itu terlihat bahwa manusia menggunakan beberapa petunjuk untuk mendapatkan lokasi sumber suara yaitu membedakan level atau intensitas suara dan waktu sampai suara dari sumber ke telinga [2].

Pada penelitian ini dirancang suatu sistem untuk menghitung arah sudut dari sumber suara menggunakan

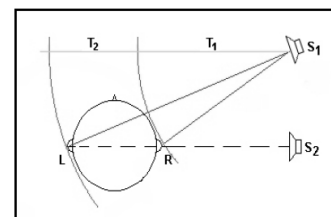
metode *Interaural Time Difference* (ITD) dengan cara menangkap suara dari sumber dengan 2 buah mikrofon lalu membedakan waktu sampai suara dari sumber menuju kedua mikrofon dengan posisi yang berbeda.

## II. INTERAURAL TIME DIFFERENCE

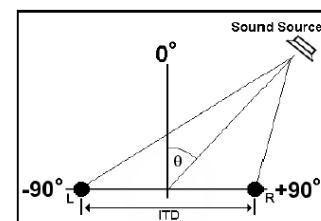
### A. Sound Localization

*Sound localization* adalah proses menentukan lokasi sumber suara berdasarkan beberapa observasi yang diterima pada sinyal suara [2]. Terdapat beberapa metode dalam proses penentuan lokasi pada *sound localization*, salah satunya adalah metode *Time Difference of Arrival* (TDOA). Metode ini terinspirasi dari cara kerja kedua telinga manusia seperti yang diilustrasikan pada Gambar 1. Jika 2 buah mikrofon dipasang pada jarak tertentu, dan saat sumber suara berjarak lebih dekat pada 1 mikrofon daripada mikrofon lainnya, akan terdapat perbedaan waktu sampai antara mikrofon yang lebih dekat dengan yang lebih jauh, perbedaan ini yang selanjutnya digunakan untuk perhitungan sudut arah datangnya sumber suara. Metode ini dapat juga disebut dengan metode *Interaural Time Difference* (ITD)[3].

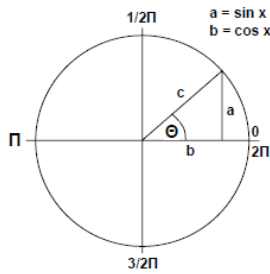
Untuk menentukan lokasi sumber suara pada lapangan dibutuhkan perhitungan azimuth antara sumber suara dengan sensor suara. Azimuth yang dihasilkan disini mempresentasikan perbandingan lokasi antara sumber suara dengan sensor suara dengan sudut  $0^\circ$  sebagai posisi sumber suara tepat di depan sensor suara. Sudut ini didapat dari perhitungan TDOA dari sinyal yang diterima oleh kedua sensor suara.



Gambar 1. Ilustrasi perbedaan waktu sampai (TDOA) antara sumber suara dengan telinga manusia sebagai sensor suara



Gambar 2. Perhitungan sudut untuk menentukan azimuth. Kiri dari  $0^\circ$  negatif dan kanan dari  $0^\circ$  positif



Gambar 3. Perhitungan sudut menggunakan prinsip trigonometri

Gambar 2 menunjukkan orientasi sudut posisi depan sumber suara terhadap sensor suara. Pencarian azimuth pada metode ITD ini pada dasarnya menggunakan prinsip trigonometri dengan nilai  $c$  pada gambar 3 sebagai jarak antara kedua mikrofon dan nilai  $a$  sebagai jarak yang didapat dari kecepatan suara yang dikalikan dengan waktu jeda pada kedua mikrofon.

Nilai  $a$  disini merupakan hasil kali dari kecepatan suara dengan waktu jeda dari kedua mikrofon sesuai dengan persamaan (1).

$$length = t \times V_{sound} = (\Delta \times \sigma) \times V_{sound} \quad (1)$$

dengan nilai  $t$  adalah waktu jeda yang digunakan untuk mencari nilai sudut,  $\Delta$  adalah frekuensi sampling,  $\sigma$  adalah hasil dari *delay* yang dihitung oleh metode *cross-correlation*, nilai  $V_{sound}$  disini adalah kecepatan rambat suara pada udara sebesar 384 m/s pada suhu sebesar 24° C. Setelah itu didapat nilai  $a$  dan  $c$  yang hasilnya dapat digunakan untuk menghitung sudut arah sumber suara dengan hukum sinus sesuai pada persamaan (2) dan persamaan (3).

$$\sin \theta = \frac{a}{c} \therefore \theta = \sin^{-1} \frac{a}{c} \quad (2)$$

$$\theta = \sin^{-1} \frac{(\Delta \times \sigma) \times V_{sound}}{c} \quad (3)$$

**B. Cross-correlation**

Salah satu variabel yang digunakan untuk mencari nilai  $\sigma$  pada persamaan (1) didapat menggunakan metode *cross-correlation*. Metode ini adalah cara untuk mencari perbedaan dari kedua sinyal, pada kasus ini yang dicari adalah perbedaan fasanya sehingga didapat jeda atau perbedaan waktu sampai sumber suara menuju kedua mikrofon.

$$Corr(g, h)_j(t) \equiv \sum_{k=0}^{N-1} g_{j+k} h_k \quad (4)$$

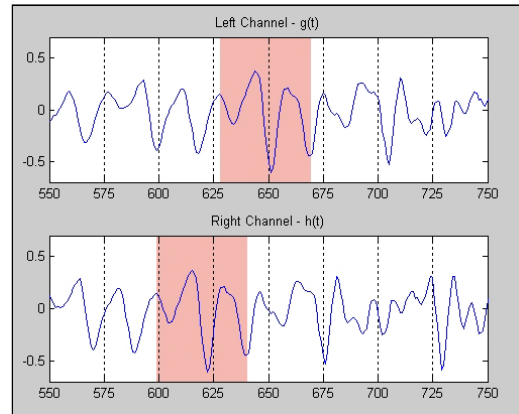
Fungsi *cross-correlation* disini didefinisikan pada persamaan (4), fungsi korelasi  $Corr(g, h)(t)$  akan memiliki nilai maksimum pada  $(t_n)$  saat fungsi  $g(t)$  digeser seiring waktu melalui persamaan  $h(t)$ .

*Cross-correlation* disini digunakan untuk membandingkan kesamaan 2 buah vektor A dan B (dimana keduanya memiliki nilai untuk sinyal  $g(t)$  dan  $h(t)$ ). Vektor A dan B disini merupakan input dari *cross-correlation* yang mempresentasikan sinyal suara yang diterima pada mikrofon. Sinyal tersebut lalu akan digeser melalui satu sama lain pada

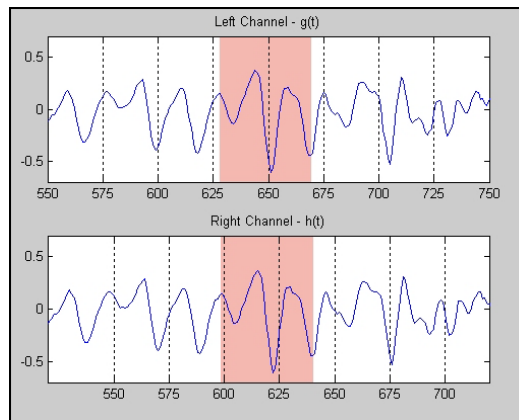
semua poin untuk menghasilkan vektor C dengan panjang sesuai dengan persamaan (5).

$$length(C) = (length(A) + length(B)) - 1 \quad (5)$$

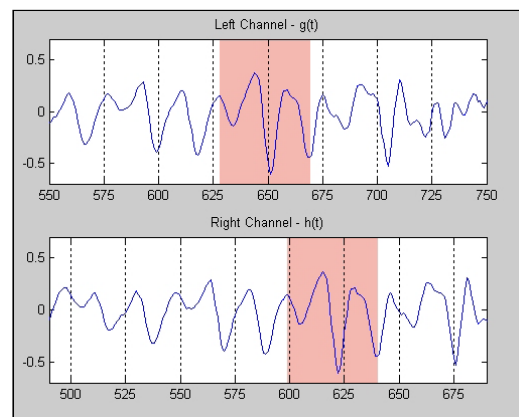
Nilai maksimum dari hasil vektor C menunjukkan korelasi maksimum diantara kedua sinyal  $g(t)$  dan  $h(t)$  dengan waktu *delay*  $\sigma$ . Pada plot dari vektor C akan didapat 2 axis x dan y dimana axis y mempresentasikan hasil perkalian dari nilai vektor A dan B dengan waktu *delay*  $\sigma$ , sedangkan axis x menunjukkan waktu atau *step* pada algoritma *cross-correlation* atau disebut juga waktu *delay*  $\sigma$ .



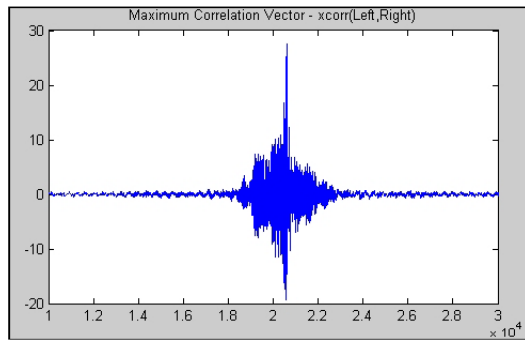
Gambar 4. Contoh plot kedua sinyal  $g(t)$  dan  $h(t)$  saat sebelum berpapasan



Gambar 5. Contoh plot kedua sinyal  $g(t)$  dan  $h(t)$  saat berpapasan



Gambar 6. Contoh plot kedua sinyal  $g(t)$  dan  $h(t)$  setelah berpapasan



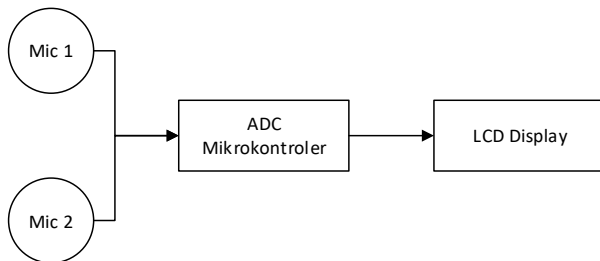
Gambar 7. Plot hasil korelasi kedua sinyal

Sebagai contoh pada gambar 4 terdapat 2 sinyal dengan bentuk sama dan fasa yang berbeda, lalu salah satu sinyal digeser sehingga fasa kedua sinyal berpapasan seperti gambar 5, setelah itu menjauh lagi seperti gambar 6. Dari gambar tersebut didapat hasil *cross-correlation* yang ditunjukkan oleh gambar 7 dengan waktu sesuai dengan pergeseran kedua sinyal, pada saat kedua sinyal belum berpapasan terlihat hasil dari *cross-correlation* bernilai rendah atau tidak terdapat kesamaan, setelah itu salah satu sinyal digeser sehingga kedua sinyal berpapasan dan terjadi kenaikan nilai pada hasil *cross-correlation* lalu mengecil lagi setelah sinyal terus bergeser dan perbedaannya membesar kembali [4].

### III. PERANCANGAN SISTEM

#### A. Diagram Blok Sistem Keseluruhan

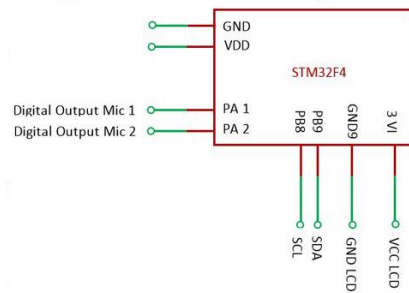
Diagram blok pada Gambar 8 menunjukkan alur kerja keseluruhan dari sistem pendeteksi arah suara. Sesuai dengan alur diagram blok pada gambar 5, cara kerja dari sistem ini menggunakan 2 buah mikrofon sebagai sensor suara dan LCD sebagai output dari sistem.



Gambar 8. Diagram blok sistem keseluruhan

#### B. Perancangan Hardware

Perancangan *hardware* pada pendeteksi suara ini menggunakan 2 buah mikrofon, mikrokontroler STM32F4, serta LCD sebagai *output display*, perancangan *hardware* yang dilakukan meliputi perancangan hubungan antara mikrokontroler dengan *input* dan *output*.



Gambar 9. Rancangan *hardware* alat pendeteksi suara

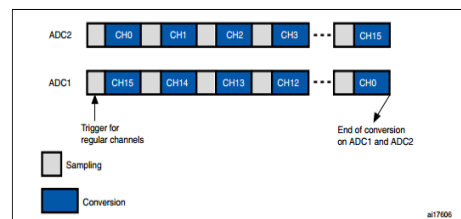
Seperti yang diperlihatkan pada gambar 9, kedua mikrofon sebagai sensor suara terhubung dengan 2 channel ADC pada mikrokontroler, ADC yang digunakan adalah ADC1 dan ADC2 serta VCC dan GND sebagai *power supply* pada mikrofon. Setelah itu *output* ditampilkan pada LCD berupa angka sudut yang sudah dihitung, LCD ini terhubung dengan VCC dan GND sebagai *power supply*, serta SDA dan SCL sebagai komunikasi data.

#### C. Perancangan Software

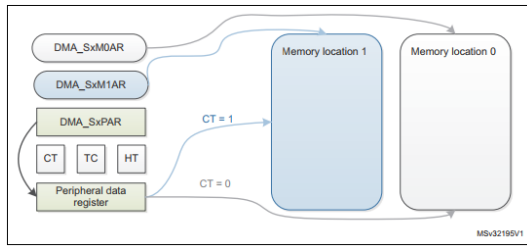
Perancangan *software* yang dilakukan pada mikrokontroler STM32F4 digunakan beberapa *mode* pada sistem ADC yaitu ADC *Dual Mode – Simultaneous* dan alokasi memori secara langsung dengan *Direct Memory Access (DMA) – Double Buffer*. Setelah itu pencarian perbedaan fasa dilakukan dengan metode *cross-correlation* dan perhitungan sudut menggunakan rumus ITD.

#### D. ADC Dual Mode – Simultaneous

Pada STM32 yang memiliki ADC lebih dari 2 ini terdapat mode dimana kedua ADC dapat bekerja secara bersamaan, proses ini melakukan 2 konversi ADC secara bersamaan sesuai dengan sinkronisasi ADC 1 dan 2 [5]. Masing-masing ADC mengonversi susunan channel atau satu channel. Proses konversi dapat dimulai dengan *trigger* eksternal atau melalui *software*. Pada mode ini, hasil konversi ADC1 dan ADC2 akan disimpan pada register data ADC1 (32-bit format). Gambar 10 menunjukkan bagaimana ADC1 dan ADC2 mengonversi 2 bagian secara bersamaan. ADC1 mengonversi channel 15 sampai 0 sedangkan ADC2 mengonversi channel 0 sampai 15.



Gambar 10. Proses konversi ADC1 dan ADC2



Gambar 11. Proses alokasi memori pada DMA mode *double buffer*

E. *Direct Memory Access (DMA) – Double Buffer*

*Direct Memory Accessi* (DMA) adalah proses yang memungkinkan proses transfer data berjalan pada *background* atau tanpa membebani kinerja prosesor, ditunjukkan pada Gambar 11. Pada proses ini, prosesor utama dapat melakukan pekerjaan lain dan hanya dapat di-*interrupt* ketika blok data sudah siap untuk diproses. Data dengan jumlah besar dapat diproses tanpa menyebabkan efek yang besar pada performa sistem.

Mode *double buffer* disini adalah operasi dimana pengisian data pada memori dilakukan menggunakan 2 buah *memory pointer*, jadi pengisian data awalnya dilakukan pada memori 1, setelah penuh data akan mengisi memori 2, dengan data pada memori 1 siap untuk diproses. Setelah data memori 2 penuh, data pada memori 2 akan diproses dan pengisian data berpindah kembali pada memori 1.

IV. PENGUJIAN SISTEM

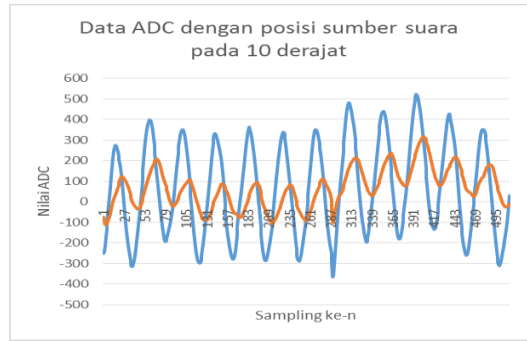
Pengujian meliputi pengujian ADC dari mikrokontroler dalam membaca nilai digital dari pembacaan suara yang dilakukan mikrofon serta perhitungan sudut menggunakan suara yang dihasilkan dari *frequency generator*. Jarak antar mikrofon yang diujikan sebesar 10, 20, dan 30 cm dan suara yang diujikan berupa sinyal sinusoidal dengan frekuensi 900 Hz dan intensitas suara sebesar 75 dB.

A. *Pengujian ADC Mikrokontroler STM32F4*

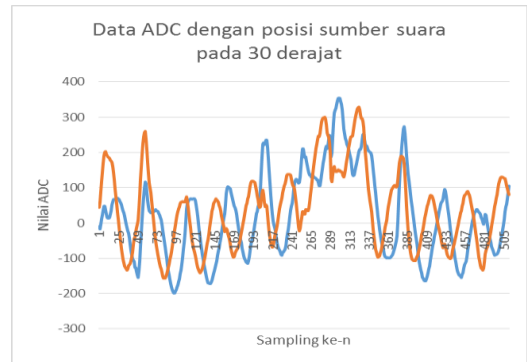
Gambar 12 menunjukkan pengujian dengan menggunakan *smartphone* sebagai sumber suara. Pada pengujian ini mikrofon membaca suara analog yang dibaca dari *smartphone* dan menghasilkan keluaran digital yang lalu dibaca oleh ADC pada mikrokontroler. Setelah 512 data masuk ke register memori, nilai tersebut disusun menjadi grafik yang membentuk sinyal serupa dengan bentuk sinyal suaranya seperti dapat dilihat pada gambar 13 dan 14.



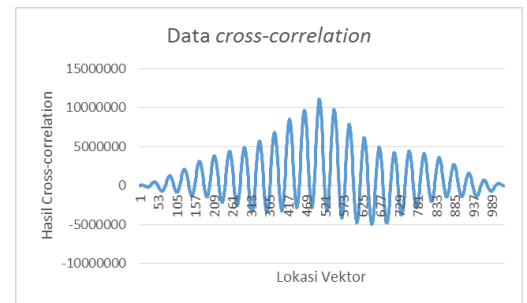
Gambar 12. *Layout* percobaan sudut dengan jarak antar mikrofon 10 cm



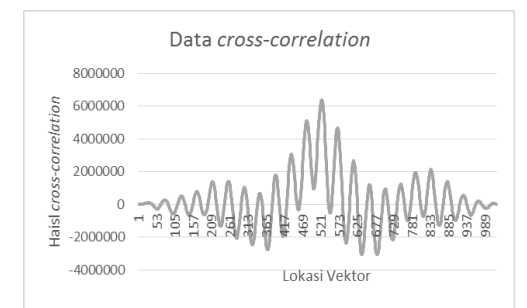
Gambar 13. Data ADC dengan posisi sumber suara pada sudut 10°



Gambar 14. Data ADC dengan posisi sumber suara pada sudut 30°



Gambar 15. Hasil *cross-correlation* kedua sinyal dengan posisi sumber suara pada 10°



Gambar 16. Hasil *cross-correlation* kedua isnyal dengan posisi sumber suara pada 30°

Selanjutnya input dari ADC akan diproses pada perhitungan *cross-correlation* sehingga menghasilkan grafik baru yang nilai maksimum dari kumpulan data tersebut adalah beda fasa dari kedua sinyal input yang kemudian dipakai untuk menghitung sudut arah sumber suara, ditunjukkan pada Gambar 15 dan 16.

Tabel 1.

Pengujian sudut dengan jarak antar mikrofon 10 cm dan jarak sumber suara 10 cm

Sudut tujuan (°)	Sudut terukur (°)	yang	Error Sudut (%)
-90	-90,85		0,94
-70	-70,33		0,47
-50	-45,79		8,42
-30	-23,65		21,17
-10	-9,25		7,5
0	0		0
10	-13,95		39,5
30	28,73		4,23
50	52,54		5,08
70	70,33		0,47
90	90,85		0,94

Tabel 2.

Pengujian sudut dengan jarak antar mikrofon 20 cm dan jarak sumber suara 10 cm

Sudut tujuan (°)	Sudut terukur (°)	yang	Error Sudut (%)
-90	-90,85		0,94
-70	-70,33		0,47
-50	-52,54		5,08
-30	-34,06		13,53
-10	-9,25		7,5
0	0		0
10	9,25		7,5
30	28,73		4,23
50	39,76		20,48
70	60,37		13,76
90	90,85		0,94

## B. Pengujian Arah Sumber Suara

Setelah nilai maksimum dari perhitungan *cross-correlation* didapat, selanjutnya dilakukan perhitungan sudut dari arah sumber suara menggunakan rumus trigonometri dengan variabel jarak antar mikrofon, frekuensi *sampling*, kecepatan suara dan hasil perhitungan korelasi. Tabel 1 menunjukkan pengujian sudut dengan jarak antar mikrofon 10 cm dan jarak sumber suara 10 cm. Percobaan dilanjutkan dengan menambah jarak antar mikrofon, ditunjukkan pada Tabel 2. Semakin besar jarak antar mikrofon dapat menghasilkan kesalahan sudut yang semakin kecil.

## V. KESIMPULAN

Pada penelitian telah dirancang dan dibuat suatu alat pendeteksi arah sumber suara berdasarkan metode *Interaural Time Difference* menggunakan mikrokontroler STM32F4. Metode *Interaural Time Difference* digunakan untuk menentukan arah dengan cara membandingkan perbedaan waktu sampai sebuah suara yang dipancarkan oleh sumber suara. Berdasarkan hasil pengujian menunjukkan bahwa pada saat menggunakan 2 buah mikrofon kondensor sebagai sensor suara, jarak antar mikrofon sebesar 10 cm dapat membaca sinyal dengan nilai *error* di bawah 10 derajat. Resolusi pembacaan sudut dapat ditingkatkan dengan cara menjauhkan jarak antar mikrofon.

## DAFTAR PUSTAKA

- [1] M. M Agung Nursyeha, "Pengenalan Suara Burung menggunakan Mel Frequency Cepstrum Coefficient dan Jaringan Syaraf Tiruan pada Sistem Pengusir Hama Burung," *Inst. Teknol. Sepuluh Nop.*, 2016.
- [2] I. Rusli, Meifal, John Malta, "Prediksi Arah Sumber Suara untuk Perawatan Prediktif," in *Prosiding Seminar Nasional Tahunan Teknik Mesin (SNTTM)*, 2010.
- [3] C. D. L. Kim, Yong-Eun, Dong-Hyun Su, Jin-Gyung Chung, Xinming Huang, "Efficient Sound Source Localization Method Using Region Selection," *IEEE*, 2009.
- [4] P. Dahlan, Bassilio, Mansoor, Wathiq, Abbasi, Milad, Honarbaksh, "Sound Source Localization for Automatic Camera Steering," *IEEE*, 2009.
- [5] N. A. Firdaus, "Alat Pelacak Arah Suara pada Sistem Pengusir Hama Burung Menggunakan ARM STM32F4," *Inst. Teknol. Sepuluh Nop.*, 2015.