



### **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.

## BAB 3

### METODE PENELITIAN

#### 3.1. Pengumpulan Data

Pengumpulan data dilakukan pada laboratorium *big data* di kampus UMN. Data yang dikumpulkan adalah data suara, masing-masing adalah suara manusia yang kencang dan suara manusia yang pelan sebanyak 152 buah *file* suara. Alat yang digunakan untuk merekam data suara tersebut adalah sebuah laptop, dengan bantuan sebuah *smartphone* yang berperan sebagai cadangan apabila terdapat masalah pada hasil rekaman di laptop. Waktu pengumpulan suara dilakukan pada sekitar bulan September hingga Oktober 2018.

Proses pengumpulan data dilakukan pada lingkungan ruang *big data* di kampus UMN yang tertutup. Responden diposisikan duduk di depan sebuah komputer laptop yang digunakan untuk merekam suara berjarak sekitar 50 cm dari posisi duduk responden. Sebuah *smartphone* diletakan di dekat komputer laptop juga dipergunakan untuk merekam, rekaman dari *smartphone* ini berperan sebagai cadangan apabila terjadi kerusakan atau cacat pada hasil rekaman komputer laptop. Responden diminta untuk menyediakan 30 suara mereka berbicara dengan volume normal, dan 30 suara mereka berbicara dengan volume kencang yang responden anggap mengganggu. Salah satu responden dari 3 responden hanya dapat menyediakan 17 suara kencang dan 15 suara pelan, menghasilkan total suara yang terkumpul menjadi 152 suara.

Pengumpulan data ini dilakukan untuk menentukan *threshold* dari program Python di yang telah disebutkan sebelumnya sebagai bagian dari proses pendeteksian kebisingan.

### 3.2. Penyambungan *software* Python dengan Arduino

Langkah berikutnya yakni membuat Python dapat menangkap seluruh lemparan data yang dilakukan oleh Arduino. Pada pengerjaan skripsi ini, akan digunakan *library* `py.serial` untuk menangkap data dari Arduino.

Setelah data ditangkap, data yang dikirimkan oleh Arduino adalah data yang belum ter-*decode*, maka perlu dilakukan sebuah konversi terlebih dahulu agar data dapat terbaca oleh program Python.

Pada program di Arduino, nilai *baud rate* ditentukan 230400 karena kecepatan ini dibutuhkan untuk mencapai kecepatan pengiriman data dari Arduino ke Python yang tinggi. Kecepatan ini pada nantinya akan mempengaruhi jumlah panjang *array* penampung data pada program Python.

### 3.3. *Software*

Pada *software* ini, akan memiliki 2 parameter, yang pertama adalah untuk memfilter hanya suara yang memiliki frekuensi tinggi yang akan diolah untuk nantinya akan melewati parameter yang kedua yakni parameter untuk mengukur intensitas dari data suara yang masuk.

*Software* Python akan menerima data suara dari Arduino hingga terkumpul sejumlah data yang cukup banyak, data suara tersebut akan melewati proses kalkulasi FFT, dan akan melewati parameter yang pertama

yakni untuk melihat apakah data tersebut memiliki frekuensi tinggi yang melebihi parameter yang pertama.

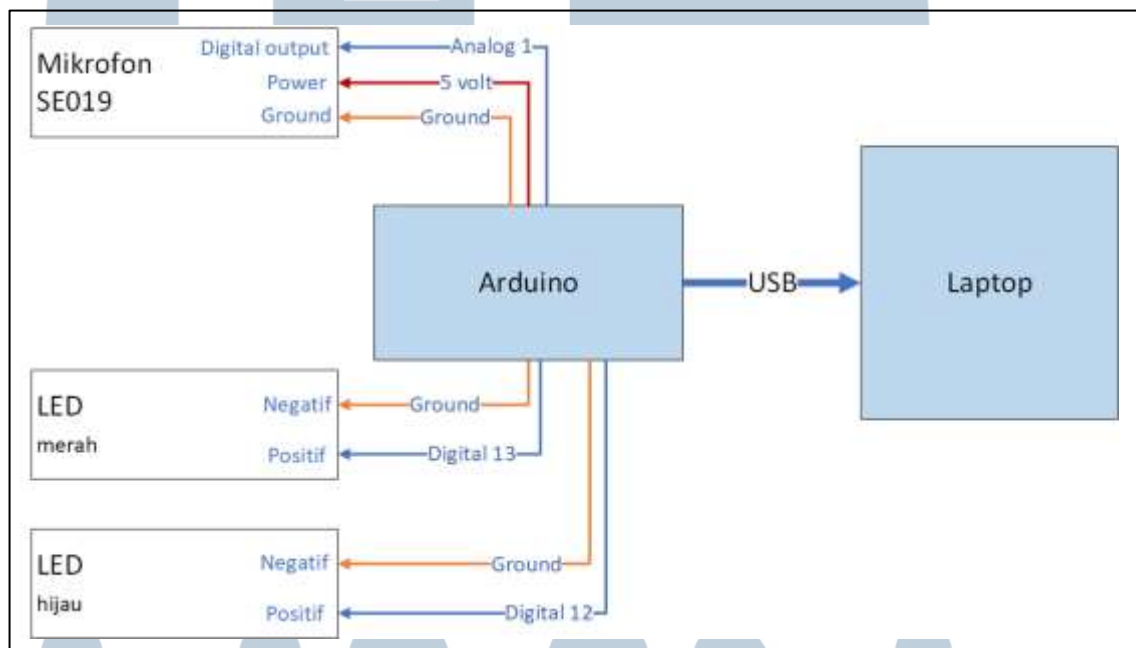
Parameter yang pertama ini adalah nilai rentang frekuensi pada *human speech* yakni 85Hz hingga. Dipilih kedua nilai ini agar suara yang akan diproses hanyalah suara-suara yang memiliki frekuensi yang memiliki frekuensi dominan di antara nilai 85Hz dan 255Hz. Apabila suara yang masuk memenuhi kriteria parameter ini maka dapat dikatakan bahwa suara ini merupakan suara manusia.

Apabila ada data suara yang melewati parameter pertama tersebut, data-data suara yang melewati parameter itu akan dibandingkan dengan nilai *threshold* volume dari hasil pengolahan data. Parameter kedua ini akan mengecek apakah data tersebut memiliki intensitas yang tinggi. Apabila data suara tersebut melewati kedua parameter ini, maka dapat dikatakan ada sebuah kebisingan dalam suara itu.

### 3.4. Rangkaian Arduino

Rangkaian Arduino terdiri dari 1 papan Arduino Uno, 1 mikrofon, 2 lampu LED, sebuah resistor, dan 5 kabel *jumper*, 3 berjenis *male-female* untuk menyambungkan mikrofon, dan 4 berjenis *male-male* untuk menyambungkan kedua buah lampu LED ke papan Arduino. Khusus untuk sumber tenaga LED akan dialirkan melalui sebuah resistor terlebih dahulu. Satu buah LED akan berfungsi sebagai indikator bahwa Arduino sedang aktif.

Setelah itu papan Arduino di program untuk menangkap sinyal audio dari mikrofon dari *port* analog 0 (A0) lalu meng-echo hasil tangkapan tersebut ke *port* serial (COM9) dalam bentuk data ASCII desimal. Diagram rangkaian Arduino dapat dilihat sebagai berikut.



**Gambar 3.1. Diagram Arduino**

Rangkaian Arduino dirakit dengan rincian sebagai berikut. Untuk LED berwarna merah disambungkan pada *port digital* 13, dan ke *grounding*.

Serupa untuk LED hijau, disambungkan pada *port* 12 dan juga disambung ke *grounding*. Modul mikrofon Arduino disambungkan ke *port* analog-in A0 sebagai masukkan papan Arduino, *port power* 5 volt untuk sumber daya mikrofon, dan *grounding*. Sumber tenaga Arduino akan berasal dari koneksi USB yang terhubung pada laptop.

### 3.5. Skalabilitas

Penerapan perangkat keras pada skripsi ini memiliki skalabilitas yang bervariasi. Penerapan yang dilakukan pada skripsi ini merupakan sebuah langkah awal untuk menunjukkan ada banyak bentuk penerapan lain yang dapat dilakukan.

Skalabilitas yang dapat dilakukan banyak terdapat pada perangkat keras Arduino. Papan Arduino ini mampu menerima banyak modul lainnya selain modul LED sebagai modul keluaran. Misalkan, lampu LED yang digunakan pada skripsi ini dapat diganti menjadi sebuah modul *relay*. Modul *relay* ini bekerja seperti layaknya sebuah saklar. Saklar ini diaktifkan oleh sebuah arus listrik yang relatif kecil, yang pada dasarnya dapat disediakan arus listrik tersebut oleh Arduino. Arus listrik tersebut yang akan berinteraksi dengan *relay* apakah modul ini diatur agar menyala atau mati. Dengan modul *relay* ini, Arduino menjadi mampu untuk mengontrol nyala atau mati sebuah alat atau benda elektronik lain yang membutuhkan tenaga listrik lebih besar. Sehingga tidak lagi menggunakan nyala LED sebagai bentuk peringatan bahwa terdeteksi adanya kebisingan, namun bisa menjadi sebuah nyala rangkaian lampu neon yang tentu apabila cukup besar ukurannya akan memiliki visibilitas yang lebih tinggi dibandingkan sebuah nyala lampu LED yang kecil.

Sumber suara yang dapat dideteksi oleh modul mikrofon ini masih sebatas kurang dari 1 meter. Hal ini diperkirakan disebabkan karena batasan

perangkat keras yaitu modul mikrofon SE019 masih memiliki beberapa kekurangan. Salah satu kekurangan pada mikrofon ini terdapat pada potensiometer pada mikrofon ini. Pengatur pada potensiometer pada mikrofon ini sangat sensitif, sehingga sulit untuk menentukan posisi yang tepat untuk mencari keseimbangan antara jarak cakupan tangkapan suara dan akurasi mikrofon ini menangkap suara. Apabila terlalu sensitif hasil olahan suara yang dihasilkan mikrofon ini menjadi tidak akurat, dengan artian walau sumber suara mengeluarkan suara dengan frekuensi sekitar 200Hz setelah diolah suaranya di Python hasilnya akan menjadi 50Hz.

