

Tempat Sampah Otomatis *Speech Recognition* Menggunakan *Pocketsphinx*

Rizki Aditya¹⁾, Abdul Muid²⁾, Uray Ristian³⁾

^{1,3)} Jurusan Rekayasa Sistem Komputer, Fakultas MIPA Universitas Tanjungpura

²⁾ Jurusan Fisika, Fakultas MIPA Universitas Tanjungpura

Jl. Prof. Dr. H. Hadari Nawawi, Pontianak Telp./Fax.: (0561) 577963

E-Mail : h1051131006@student.untan.ac.id¹⁾; muid@physics.untan.ac.id²⁾; eristian@siskom.untan.ac.id³⁾;

ABSTRAK

Penelitian ini dirancang sebuah tempat sampah otomatis menggunakan *speech recognition* sebagai pembuka tutup tempat sampah yang bekerja secara otomatis dan dilengkapi umpan balik berupa lampu LED serta LCD yang akan memberi tahu bahwa tempat sampah tersebut penuh sehingga perlu dibuang sampahnya. Mikrofon digunakan sebagai penerima suara, motor servo sebagai pembuka tutup tempat sampah, sensor ultrasonik sebagai deteksi sampah, LED dan LCD sebagai indikator, serta Raspberry Pi 3 sebagai pengolah datanya. Agar alat ini dapat digunakan secara *offline* maka digunakan library PocketSphinx karena menggunakan Hidden Markov Model sebagai metodenya. Dari penelitian, diperoleh hasil pada noise rendah, waktu respon alat rata-rata 5,13 detik, pada noise sedang, rata-rata memerlukan waktu 9,17 detik dan dengan menetapkan batas toleransi 11 detik sebagai respon tidaknya alat pada noise tinggi, perintah suara dianggap tidak terdeteksi.

Kata Kunci – Raspberry Pi 3, PocketSphinx, Hidden Markov Model, Tempat Sampah, *Speech Recognition*

1. PENDAHULUAN

Membuang sampah pada tempatnya merupakan kewajiban kita sebagai warga yang baik dalam menjaga lingkungan. Terkadang membuang sampah pada tempatnya masih terkendala dengan keterbatasan tangan manusia. Terkadang sering juga didapati tempat sampah sudah penuh tapi belum dibuang muatannya karena tidak ada yang tahu. Maka dari itu kebutuhan teknologi yang dapat mengontrol dan memonitoring tempat sampah sangat diperlukan. Salah satu pemanfaatan teknologi yang sederhana yakni tempat pembuangan sampah menggunakan *Speech Recognition*. *Speech Recognition* memungkinkan mengontrol berbagai alat dan perlengkapan lainnya tanpa menggunakan tangan melainkan hanya dengan perintah suara karena *Speech Recognition* memproses suara masukan secara otomatis sehingga siap melakukan perintah yang diberikan (Zwass, 2016).

Penelitian yang terkait dengan *Speech Recognition* telah dilakukan oleh Indriyani (2018) yang berjudul “Perancangan *Smart Door Lock* menggunakan *Voice Recognition* berbasis Raspberry Pi 3”. Penelitian tersebut menghasilkan aplikasi yang dapat mengidentifikasi suara pengguna tertentu sehingga dapat membuka kunci pintu ruang server (Indriyani, Aryani, Iskandar, 2018). Penelitian selanjutnya yang terkait dengan *Speech Recognition* telah dilakukan oleh Arbaus (2016) yang berjudul “Perancangan *Smart Door Lock* menggunakan *Voice Recognition* berbasis Raspberry Pi 3”. Penelitian ini menghasilkan aplikasi yang mampu mengenali suara untuk membuka pintu dengan jarak efektif dari sumber suara sejauh 10-20 cm (Arbaus, Prasetya, Sari, 2016). Penelitian lainnya yang terkait dengan tempat sampah otomatis yakni penelitian yang dilakukan oleh Widiyanto (2014) yang berjudul “Tempat Sampah Buka Tutup Otomatis”. Penelitian tersebut menghasilkan sistem yang membuka dan menutup menggunakan sensor inframerah

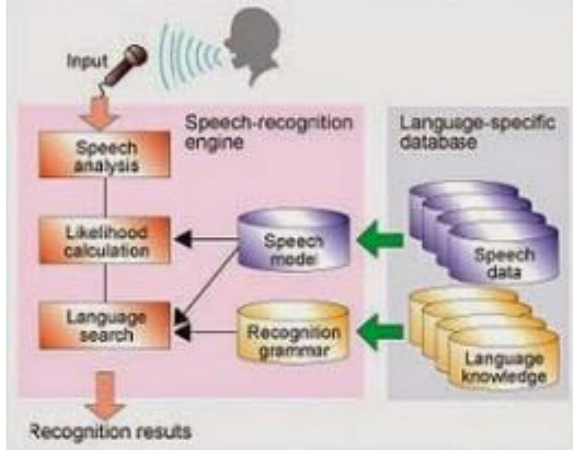
(Widiyanto & Murti, 2014). Dari penelitian yang telah dilakukan sebelumnya, pengembangan yang dapat dilakukan adalah membangun perangkat *embedded Speech Recognition* menggunakan *PocketSphinx*. *Pocketsphinx* bekerja menggunakan metode *Hidden Markov Model* (HMM) sehingga perangkat dapat bekerja secara *offline*. Selain itu penerapan HMM pada pengenalan suara digunakan untuk memprediksi suara yang diucapkan berdasarkan *dictionary file* dan *language model*. HMM akan memberikan probabilitas berdasarkan kosakata yang akan diucapkan. Probabilitas kemudian dihitung berdasarkan pembobotannya sehingga mempunyai nilai terbaik yang memungkinkan kata tersebut dikeluarkan berdasarkan persamaan suara (Allauddin, Kurniawan, Setiawan, 2016). Tempat sampah otomatis bekerja dengan mendapatkan masukan berupa perintah suara (sinyal analog) melalui mikrofon. Kemudian diolah oleh modul *Speech Recognition* dengan cara mencocokkannya dengan data *lm (language model)* dan data *dic (dictionary)* dimana menggunakan *Hidden Markov Model* sehingga perintah suara dapat dikenali menjadi bentuk tulisan. Setelah perintah didapat dalam bentuk tulisan maka dapat diprogram untuk keluaran berupa tutup yang terbuka. Untuk menciptakan sebuah perangkat *embedded* tersebut diperlukan sebuah mikrofon, *usb sound card*, Raspberry Pi 3, LCD, LED hijau, LED merah, sensor ultrasonik, motor servo. Perangkat tersebut dikontrol agar dapat bekerja sama sebagai sebuah sistem yang baik dan terstruktur.

2. TINJAUAN PUSAKA

A. *Speech Recognition*

Speech Recognition bekerja dengan cara merubah sinyal analog menjadi sinyal digital oleh *analog to digital converter* (ADC). Pada Gambar 1 Ilustrasi *Speech Recognition* merupakan skema dari yang menjelaskan proses untuk mengenali dan

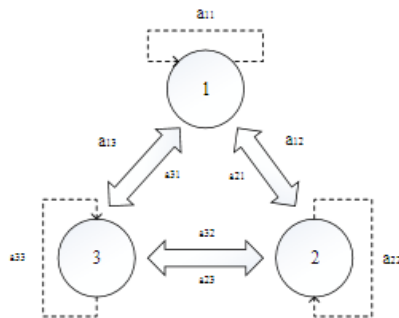
memahami kata-kata yang diucapkan oleh satu orang atau banyak orang yang diinputkan melalui mikrofon (ASR). Kemudian memprosesnya sehingga dapat mengenali apa yang diucapkan oleh penggunanya. ASR adalah singkatan dari *Automatic Speaker Recognition* yang digunakan untuk mengenali seseorang dari frasa yang diucapkan.



Gambar 1. Ilustrasi *Speech Recognition* (Mutohar, 2007)

B. Hidden Markov Model

Hidden Markov Model adalah perluasan dari rantai *Markov* dimana *statenya* tidak dapat diamati secara langsung (tersembunyi), tetapi hanya dapat diobservasi melalui suatu himpunan pengamatan lain.



Gambar 2. Rantai Markov (Firdaniza, Gusriani, & Akmal, 2006)

Gambar 2 Rantai Markov merupakan proses stokastik dengan ruang *state* (keadaan) diskrit, serta mempunyai sifat dimana *state* “saat ini” hanya bergantung pada *state* “sebelumnya” (Firdaniza, Gusriani, Akmal, 2006). Pada rantai *Markov* setiap *state* dapat diamati secara langsung sedangkan pada *Hidden Markov Model* memerlukan penyelesaian terhadap tiga hal mendasarnya agar dapat melihat *statenya* yakni:

1. *Evaluation problem* dapat diselesaikan menggunakan Algoritma *Forward* $\alpha_t(i)$:

$$\alpha_t(i) = P(O_1 = v_{m_1}, \dots, O_t = v_{m_t}, Q_t = s_i | \lambda)$$

..... (1)

$\alpha_t(i)$: Variabel *forward*

O_1, O_2, O_t : Peluang barisan observasi

$s_i | \lambda$: *State* s_i pada waktu t jika diberikan λ .

2. *Decoding problem* dapat diselesaikan dengan mencari barisan *state* yang optimal $Q = \{Q_1, Q_2, \dots, Q_T\}$.

3. *Learning problem* dapat menggunakan algoritma *Baum-Welch*. Dalam algoritma *Baum-Welch*, didefinisikan empat variabel yakni : variabel *forward*, variabel *backward*, variabel $\xi_t(i, j)$, dan variabel $\gamma_t(i)$. Variabel

forward dan variabel *backward* akan digunakan dalam perhitungan variabel $\xi_t(i, j)$ dan variabel $\gamma_t(i)$.

Variabel *backward*:

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \dots, O_T | Q_t = s_i | \lambda)$$

..... (2)

$\beta_t(i)$: Variabel *backward*

$O_{t+1}, O_{t+2}, \dots, O_T | Q_t$: Peluang dari barisan observasi parsial $s_i | \lambda$: *State* s_i pada waktu t dan model λ .

Variabel $\xi_t(i, j)$:

$$\xi_t(i, j) = P(Q_t = s_i, O_{t+1} = s_j | O, \lambda) \dots (3)$$

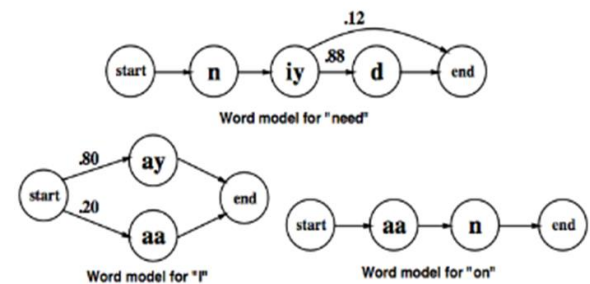
Persamaan diatas merupakan peluang proses pada *state* s_i dan pada saat $t+1$ berada pada *state* s_j , jika diberikan barisan O dan model λ .

Variabel $\gamma_t(i)$:

$$\gamma_t(i) = P(Q_t = s_i | O, \lambda) \dots (4)$$

Persamaan diatas merupakan peluang proses berada pada *state* s_i saat waktu t , jika diberikan barisan observasi O dan model λ .

Dengan rumus variabel di atas memungkinkan *Hidden Markov Model* dapat digunakan dalam berbagai masalah di kehidupan sehari-hari seperti pengenalan pola, pengenalan suara, pengenalan DNA dan masalah lainnya yang *state-nya* tidak diketahui.



Gambar 3. Alur Kerja *Hidden Markov Model* (Alauddin, Kurniawan, & Setiawan, 2016)

Gambar 3 Alur Kerja *Hidden Markov Model* menjelaskan penerapan *Hidden Markov Model* yang

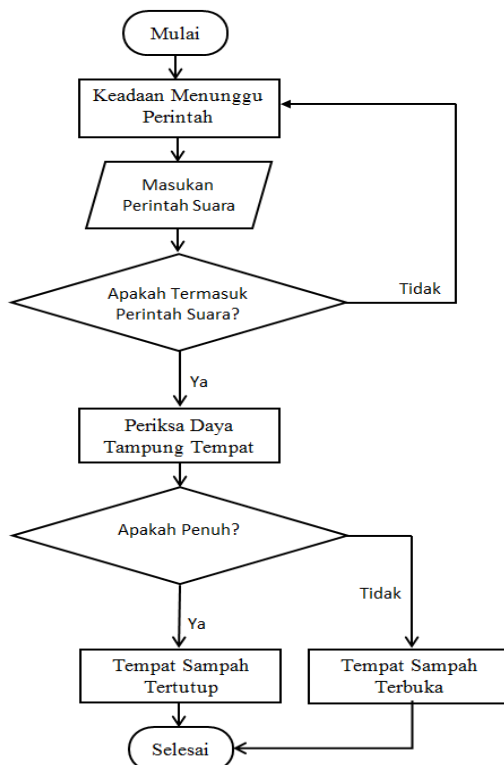
memprediksi suara berdasarkan *dictionary file (dic)*, *language model (lm)* dan *acoustic model*. *Hidden Markov Model* akan memberikan probabilitas berdasarkan pembobotannya sehingga mempunyai nilai terbaik yang memungkinkan kata tersebut dikeluarkan berdasarkan persamaan suara (Allauddin, Kurniawan, Setiawan, 2016).

C. PocketSphinx

PocketSphinx ialah sistem pengenalan suara tercepat yang dikembangkan oleh Carnegie Mellon University (CMU). *PocketSphinx* menggunakan bahasa pemrograman *C* murni dan sangat optimal karena bersifat *real-time* sehingga mesin dapat mengenali suara dengan akurat (Allauddin, Kurniawan, Setiawan, 2016). *PocketSphinx* mengandalkan respon yang sangat cepat dan konsumsi sumber daya yang minim sehingga memungkinkan pengembangan pengenalan suara cocok digunakan untuk aplikasi *desktop*, komando dan kontrol serta pendiktean. *Library* ini menyediakan fasilitas untuk para penggunaanya untuk memodifikasi bahasa yang akan digunakan untuk mengenali suara seseorang (Lenzo, 2009). Aplikasi *PocketSphinx* merupakan sistem pengenalan suara yang biasa digunakan untuk aplikasi desktop pada saat ini. *PocketSphinx* biasa digunakan untuk kebutuhan *embedded system* dan robot karena *library* didukung memiliki kelebihan komputasi yang cepat dan ringan dalam mengenali suara seseorang.

3. METODE PENELITIAN

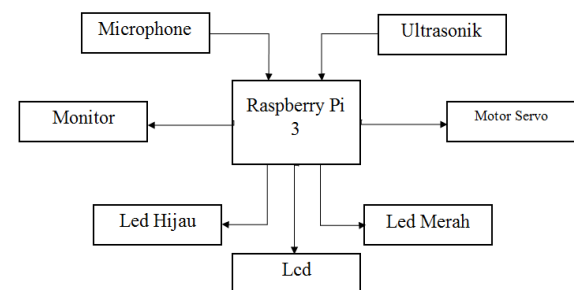
Berikut ini merupakan flowchart sistem dari tempat sampah otomatis:



Gambar 4. Flowchart Tempat Sampah Otomatis

Gambar 4 Flowchart Tempat Sampah Otomatis menjelaskan alur kerja dari *Speech Recognition* terhadap tempat sampah. Dimulai dengan keadaan menunggu (*stand by*), kemudian melakukan perulangan sampai mendeteksi suara. Ketika suara terdeteksi maka akan dilakukan proses klarifikasi yang menentukan apakah suara tersebut merupakan salah satu dari perintah suara atau bukan. Apabila bukan salah satu dari perintah suara maka akan kembali ke proses menunggu (*stand by*). Apabila merupakan salah satu dari perintah suara maka akan berlanjut ke proses selanjutnya yaitu memeriksa daya tampung tempat sampah. Proses pengukuran daya tampung tempat sampah menggunakan sensor ultrasonik. Apabila hasil pengukuran kurang dari atau sama dengan 10 cm yang menandakan tempat sampah penuh, maka tutup tempat sampah tidak akan terbuka dan proses selesai. Namun apabila hasil dari pengukuran lebih dari 10 cm maka daya tampung tempat sampah dinyatakan belum penuh sehingga tutup tempat sampah terbuka dan proses selesai.

Berikut ini merupakan gambar diagram blok sistem:



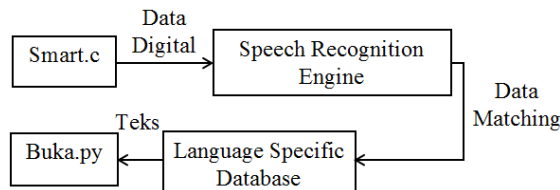
Gambar 5. Diagram Blok Sistem

Pada Gambar 5 Diagram Blok Sistem menjelaskan sistem akan bekerja ketika mendapatkan *input* berupa perintah suara seperti “open” atau “buka”. Perintah suara diberikan melalui mikrofon dan diteruskan oleh *CMUSphinx* melalui *library PocketSphinx* agar dapat dicocokkan dengan data *lm (language model)* dan *dic (dictionary)* yang sudah di atur agar dapat mengenali ucapan “open” atau “buka”. Kemudian pada program *C* yang telah dibuat akan mengatur ucapan yang dikenali agar dapat memanggil *file Python*. *File Python* yang dipanggil berisi aturan kerja sistematis alat agar dapat bekerja sesuai porsi kondisi daya tampung tempat sampah. Apabila daya tampung tempat sampah kosong setelah di ukur dengan sensor ultrasonik maka tutup tempat sampah akan terbuka menggunakan motor servo, LED hijau akan menyala dan LCD akan menampilkan tulisan “Silahkan buang sampah yang anda bawa”. Namun apabila daya tampung tempat sampah penuh setelah diukur dengan sensor ultrasonik maka tutup tempat sampah akan tetap tertutup, LED merah akan menyala dan LCD akan menampilkan tulisan “Tempat sampah penuh silahkan buang muatan”. Pada monitor akan ditampilkan proses tahapan

pengenalan suara dan proses pemanggilan *file Python* yang siap di panggil apabila perintah suara sesuai.

A. Proses Pengolahan Perintah Suara.

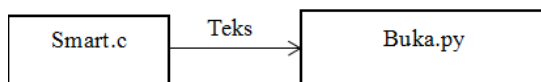
Perintah suara yang sudah diucapkan pengguna akan diproses *Speech Recognizer* menggunakan *sound card* dan menyimpannya dalam bentuk digital. Hasil digitalisasi kemudian dikonversi menjadi spektrum suara yang akan dianalisa dengan cara membandingkannya dengan *template* suara pada database sistem. HMM (*Hidden Markov Model*) bekerja dengan menggunakan model statistikal. Model statistikal bekerja dengan menggunakan nilai-nilai statistik dari sebuah sinyal. Oleh karena itu variabel masukan HMM sebagai parameter-parameter adalah data diskrit gelombang suara yang disebut *n*. Agar dapat memproses data suara yang diterima dengan *language specific database*, maka digunakan *Library Pocketsphinx*. *Library Pocketsphinx* di *include* ke dalam kode program *C* yang diberi nama *smart.c*.



Gambar 6. Diagram Alir program smart.c

Pada Gambar 6 Diagram Alir program *smart.c* dapat dilihat *library PocketSphinx* di *include* ke dalam kode program *smart.c*. *Library* ini memproses perintah suara yang terdeteksi *Speech Recognizer* dan mencocokkannya dengan data *language specific database* yang berisi data dic (*speech model*) dan *lm (recognition grammar)* yang disebut juga sebagai *language model*.

Library Pocketsphinx bekerja sebagai *Speech Recognition Engine* yang memproses perintah suara mulai dari inialisasi dekoder, menampilkan hasil dari dekoder yang berhubungan dengan perintah suara (*language model*) dan menampilkan teks yang sesuai dengan pengucapan perintah pada *language specific database*.

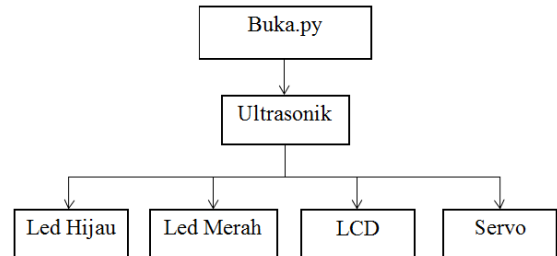


Gambar 7. Diagram Alir pemanggilan Program Buka.py

Pada Gambar 7 Diagram Alir pemanggilan Program *Buka.py* menjelaskan pengaturan perintah selanjutnya ketika perintah suara sudah terdeteksi oleh *Library Pocketsphinx* menjadi sebuah teks, maka program akan mengeksekusi kode program aksi berupa kode program *Python* yang diberi nama *buka.py*. Kode program *buka.py* akan tereksekusi apabila perintah suara yang terdeteksi berupa kata “*open*” atau “*buka*”.

B. Proses Penerimaan Data, Informasi dan Notifikasi

Proses penerimaan data, informasi dan notifikasi dilakukan pada kode program *buka.py* yang menggunakan bahasa *Python*. kode program *buka.py* mengatur penerimaan data hasil pengukuran dari sensor ultrasonik yang mengukur kedalaman tabung tempat sampah.



Gambar 8. Diagram Blok Program Buka.py

Pada Gambar 8 Diagram Blok Program *Buka.py* menjelaskan program bekerja dimulai dengan melakukan pengukuran jarak objek yang ada di dalam tempat sampah menggunakan sensor ultrasonik. Hasil pengukuran dibagi menjadi dua kondisi yakni ketika hasil pengukuran sensor ultrasonik menunjukkan angka lebih dari 10 cm dan kurang dari atau sama dengan 10 cm.

Apabila jarak objek dengan sensor ultrasonik di dalam tempat sampah kurang dari atau sama dengan 10 cm, maka tempat sampah dinyatakan penuh. Maka oleh *Raspberry Pi* memberikan tanggapan berdasarkan kode program *buka.py* berupa mengirimkan notifikasi berupa LED merah dan informasi melalui LCD bahwa “*Tempat Sampah Penuh Harap Segera Buang Muatan Terima Kasih*”.

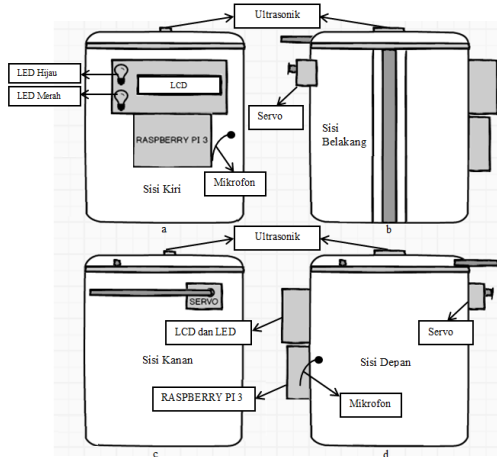
Apabila jarak objek dengan sensor ultrasonik di dalam tempat sampah lebih dari 10 cm, maka tempat sampah dinyatakan tidak penuh atau dengan kata lain masih mampu menampung sampah. Maka oleh *Raspberry Pi* memberikan tanggapan berdasarkan kode program *buka.py* berupa mengirimkan notifikasi berupa LED hijau dan informasi melalui LCD bahwa “*Tutup Terbuka Silahkan Buang Sampah Yang Anda Bawa*”.

C. Proses Pengontrolan Buka Tutup Sampah Otomatis

Pada Gambar 8 Flowchart Program *Buka.py* dapat dilihat tutup tempat sampah dapat terbuka dan tertutup karena menggunakan motor servo sebagai penggerakannya. Motor servo bekerja melalui perintah *Raspberry Pi* apabila memenuhi kondisi tertentu. Apabila hasil pengukuran menyatakan jarak objek dengan sensor ultrasonik lebih dari 10 cm maka *Raspberry Pi* akan memberikan tanggapan berdasarkan kode program *buka.py* berupa menggerakkan lengan motor servo menggunakan pin yang terhubung antara servo dan *Raspberry Pi* sehingga tutup tempat sampah terbuka. Kemudian untuk menutup tutup tempat sampah, pada kode program *buka.py* terdapat durasi lama tutup tempat

sampah terbuka. Ketika durasi tersebut telah habis maka Raspberry Pi mendapatkan perintah baru dari kode program buka.py yakni menggerakkan lengan servo ke arah sebaliknya sehingga tutup tempat sampah tertutup.

D. Perancangan Perangkat Keras



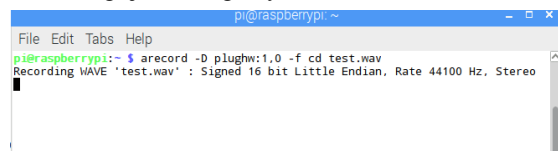
Gambar 9. Tempat Sampah Otomatis, tampak sisi kiri (a), tampak sisi belakang (b), tampak sisi kanan (c), tampak sisi depan (d)

Pada Gambar 9 Tempat Sampah Otomatis, tampak sisi kiri (a), tampak sisi belakang (b), tampak sisi kanan (c), tampak sisi depan (d), posisi alat sensor telah disusun sedemikian rupa dimana mikrofon terhubung langsung dengan Raspberry Pi 3 melalui *sound card* sedangkan sensor ultrasonik terpasang di atas tutup tempat sampah namun menghadap ke bawah, tujuannya agar dapat mengukur daya tampung tempat sampah. LCD, LED merah dan LED hijau terletak di dalam *case* kotak di sisi kiri tempat sampah. Sedangkan motor servo terletak di sisi kanan tempat sampah. Dengan penempatan alat sensor yang sesuai, sistem dapat bekerja dengan maksimal dan terstruktur. Sistem ini memberikan *input* melalui mikrofon dan sensor ultrasonik sedangkan *output* melalui monitor, motor servo, LCD, LED hijau dan LED merah. Untuk pengumpulan dan pengolahan data sistem ini menggunakan Raspberry Pi 3 sehingga semua alat sensor terhubung langsung pada pin Raspberry Pi 3.

4. HASIL DAN PEMBAHASAN

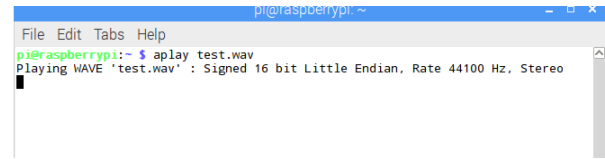
A. Implementasi Mikrofon

Berikut ini tahapan pengujian dengan cara merekam suara dan memutar rekamannya kembali untuk menguji berfungsinya mikrofon.



Gambar 10. Perintah Uji Rekam Suara

Pada Gambar 10 Perintah Uji Rekam Suara terdapat kalimat perintah *arecord -D plughw:1,0 -f cd test.wav*. perintah *arecord* sendiri bertujuan untuk merekam suara, *plughw* untuk mengatur suara masukan dan keluaran tetap melalui *sound card* yang terhubung melalui slot *usb*. Sedangkan *test.wav* adalah nama *file* suara yang direkam.



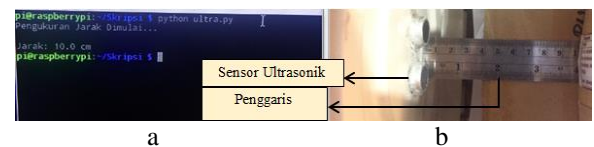
Gambar 11. Perintah Putar Suara

Pada Gambar 11 Perintah Putar Suara terdapat kalimat perintah *aplay test.wav* yang bertujuan untuk memutar *file* suara yang sudah di rekam. Bila suara rekaman terdengar dengan jelas maka dapat dilihat bahwa mikrofon dapat bekerja dengan baik tanpa kendala.

Hasil pengujian mikrofon yakni berupa perekaman suara membuktikan Raspberry Pi dapat menerima suara masukan menggunakan mikrofon melalui perantara *usb sound card*. Suara yang terekam akan semakin jelas apabila lingkungan sekitar perekaman tidak berisik. Sehingga tingkat kebisingan di lingkungan sekitar perekaman akan mempengaruhi kualitas suara yang di dapat. Suara yang tidak diinginkan sehingga mengganggu proses perekaman suara disebut juga sebagai *noise*. *Noise* akan mempengaruhi kinerja sistem *Speech Recognition* dalam mengidentifikasi suara sehingga sistem memiliki kecenderungan tidak dapat mengenali suara masukan yang diberikan.

B. Implementasi Sensor Ultrasonik

Berikut ini pengecekan sensor ultrasonik yang bertujuan untuk memastikan tidak ada kendala saat program utama yakni buka.py berjalan.



Gambar 12. Pengujian Sensor Ultrasonik, hasil pada terminal (a), pengukuran pada penggaris (b)

Pada Gambar 12 Pengujian Sensor Ultrasonik, hasil pada terminal (a), pengukuran pada penggaris (b), dapat dilihat bahwa sensor ultrasonik dapat bekerja dengan baik sehingga dipastikan tidak ada kendala saat program utama berjalan. Pada gambar proses pengukuran hanya dilakukan satu kali oleh sensor karena pada kode program tidak menggunakan perintah *while true*. Perintah *while true* akan selalu bekerja ketika kondisi sensor ultrasonik memenuhi syarat pengukuran sehingga pengukuran jarak antar sensor ultrasonik dengan objek akan terus di lakukan berulang-ulang atau dikenal dengan sebutan *looping*.

C. Implementasi Motor Servo

Pada Gambar 13 Pengaturan Sudut Lengan Servo Membuka dan Menutup, tutup terbuka (a), tutup tertutup (b), pengecekan motor servo dilakukan sambil mengatur sudut perpindahan lengan motor servo dan mengatur titik angkat tutup tempat sampah agar lengan motor servo bekerja dengan maksimal.



Gambar 13. Pengaturan Sudut Lengan Servo Membuka dan Menutup, tutup terbuka (a), tutup tertutup (b)

Pengaturan motor servo pada tahap pengujian menghasilkan gerakan yang sesuai yakni kurang dari 90 derajat. Tujuannya agar ketika lengan servo dalam posisi membuka tidak mendorong tutup jauh ke belakang namun tetap menempel atau menyandar pada lengan servo. Sehingga ketika akan menutup, tutup tempat sampah yang menyandar pada lengan servo akan mengikuti gerakan lengan servo yang menutup.

D. Implementasi Speech Recognition

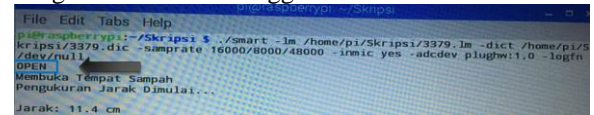
Program smart.c dibuat sebagai wadah yang mengatur proses masukan suara oleh mikrofon dan mengolah data suara menjadi teks menggunakan *library*. Untuk pengujian program smart.c menggunakan perintah yang di tuliskan di dalam terminal Raspberry Pi.



Gambar 14. Pemanggilan Program *Speech Recognition*

Pada Gambar 14 Pemanggilan Program *Speech Recognition* merupakan pemanggilan program *Speech Recognition* untuk menjalankan keseluruhan program. Perintah yang digunakan adalah `./smart -lm /home/pi/Skripsi/3379.lm -dict /home/pi/Skripsi/3379.dic -samprate 16000/8000/48000 -inmic yes -adcdev plughw:1.0`. Pada perintah di terminal tersebut dapat dilihat bahwa program smart.c di *compile* bersamaan dengan *file lm* dan *dic* sebagai *template* beserta alamat filenya dan didalamnya terdapat *inmic yes* menandakan perintah tersebut berjalan bersamaan dengan penggunaan mikrofon. Dapat dilihat pada

gambar bahwa program smart.c sudah berjalan dengan baik dan menunggu masukan suara.



Gambar 15. Pengujian Perintah Suara

Pada Gambar 15 Pengujian Perintah Suara dapat dilihat program berhasil mendeteksi perintah suara berupa kata “open” melalui mikrofon. Setelah mendeteksi kata perintah program langsung menjalankan program selanjutnya yaitu mengukur daya tampung tempat sampah dan memberikan respon yang sesuai berdasarkan daya tampung tempat sampah. tutup tempat sampah akan terbuka ketika hasil pengukuran sensor ultrasonik berada lebih dari 10 cm dan LED hijau akan menyala beserta pemberitahuan khusus pada LCD. Setelah itu LED merah akan menyala menandakan tutup tempat sampah tertutup. Namun apabila hasil pengukuran kurang dari 10 cm maka LED merah menyala beserta pemberitahuan khusus pada LCD. Hal ini dikarenakan daya tampung pada tempat sampah penuh sesuai dengan hasil pengukuran sensor ultrasonik.

Tabel 1. Waktu Respon Saat di Kamar

No	Penguji	Waktu Respon	Banyak Respon	Keadaan Sekitar	Kendala
1	Rizki	4,92 s	1x	Situasi tidak ramai dan tidak gaduh	Tidak ada
2	Satria	5,59 s	1x	Situasi tidak ramai dan tidak gaduh	Tidak ada
3	Prawira	5,22 s	3x	Situasi tidak ramai dan tidak gaduh	Alat gagal mendeteksi perintah suara dari pengguna

Menurut Tabel 1 Waktu Respon Saat di Kamar, pada data pengujian satu dan dua dinyatakan alat dapat menerima perintah suara dari penggunaanya tanpa kendala dengan syarat situasinya tidak ramai dan tidak gaduh sehingga memudahkan alat dalam mengidentifikasi perintah suara yang diberikan dengan data template seperti *language model* dan *dictionary*. Data pengujian ketiga didapat setelah tiga kali percobaan dan melewati waktu batas toleransi berdasarkan banyaknya percobaan yakni 11 detik sehingga dapat dipastikan program tidak dapat merespon perintah suara. Hal ini dikarenakan *tone* suara pengguna sulit untuk diidentifikasi untuk dicocokkan dengan data template. Namun dikarenakan menggunakan metode HMM program akan berusaha mengidentifikasi suara dengan perintah suara yang paling mendekati sehingga ketika percobaan satu dan dua program seperti dilatih agar dapat mendeteksi perintah suara yang mendekati. Akhirnya pada percobaan ketiga program dapat mengidentifikasi perintah suara yang diberikan.

Tabel 2. Waktu Respon Saat di Hall SISKOM

No	Penguji	Waktu Respon	Banyak Respon	Keadaan Sekitar	Kendala
1	Irham	5,12 s	1x	Situasi ramai dan tidak gaduh	Tidak ada
2	Andre	5,00 s	4x	Situasi ramai dan gaduh	Alat gagal mendeteksi perintah suara dari pengguna
3	Titania	4,55 s	1x	Situasi tidak ramai dan tidak gaduh	Tidak ada
4	Nabila	4,80 s	3x	Situasi tidak ramai dan tidak gaduh	Pengucapan perintah suara kurang tegas

Pada Tabel 2 Waktu Respon Saat di Hall SISKOM, data pengujian satu dan tiga dinyatakan alat dapat menerima perintah suara dari penggunaannya tanpa kendala dengan syarat situasinya tidak gaduh sehingga memudahkan alat dalam mengidentifikasi perintah suara yang diberikan dengan data template seperti *language model* dan *dictionary*. Data pengujian kedua didapat setelah empat kali percobaan dan melewati batas waktu maksimal berdasarkan banyaknya percobaan yakni 11 detik sehingga dapat dipastikan program tidak dapat merespon perintah suara. Penyebab suara penguji (Andre) tidak dapat dideteksi oleh alat dikarenakan saat penguji memberikan perintah suara, disekitar penguji terdapat beberapa orang yang sedang mengobrol sehingga alat sulit mendeteksi perintah suara yang diberikan oleh penguji hingga percobaan ketiga. Pada saat percobaan keempat, perintah suara dapat diidentifikasi karena beberapa orang yang mengobrol tadi sudah menjauh dari penguji. Untuk penguji keempat, perintah suara penguji (Nabila) tidak dapat dideteksi walaupun situasinya tidak gaduh. Hal ini dikarenakan artikulasi ataupun pengucapan yang dilakukan penguji kurang tegas sehingga alat kesulitan memproses perintah suara yang diberikan sehingga dengan metode HMM alat akan melatih perintah suara yang diterima ke template yang paling mendekati, sehingga pada percobaan ketiga perintah suara yang diberikan oleh penguji dapat diidentifikasi oleh alat dan diproses.

Tabel 3. Pengujian yang telah dilakukan

No	Pengujian	Proses	Hasil Yang Diharapkan	Hasil Pengujian
1	Mikrofon	Raspberry Pi menerima <i>input</i> suara melalui <i>usb</i> eksternal sebagai default	Raspberry Pi merekam suara yang di masukkan melalui mikrofon	Berhasil
2	Sensor Ultrasonik	Mengukur selisih pengukuran ultrasonik dengan alat ukur standar (penggaris)	Ultrasonik dapat mengukur dengan jarak yang tepat atau selisih kecil dengan alat ukur standar	Berhasil
3	<i>Speech Recognition</i>	Mengubah suara menjadi tulisan atau text	<i>Speech Recognition</i> dapat mengubah perintah suara secara akurat	Berhasil

Tabel 3 Pengujian yang telah dilakukan merupakan tabel hasil pengujian yang telah dilakukan dan hasil yang telah didapat dari pengujian. Pengujian mikrofon dan sensor ultrasonik dilakukan terpisah agar dapat diketahui program dan alat yang belum rampung. Sedangkan pengujian *Speech Recognition* dilakukan pada tahap akhir karena dapat di pastikan semua kode program dan alat tidak ada yang bermasalah sehingga bisa didapat waktu respon *Speech Recognition* dengan perintah suara yang diberikan.

Tabel 4. Klasifikasi Keadaan Sekitar Alat

Keadaan Sekitar	Total Waktu (s)	Banyak Percobaan	Rata-Rata (s)
Tidak Gaduh	25,65	5	5,13
Sedikit Gaduh	45,86	5	9,17
Gaduh	62,88	5	12,57

Pada Tabel 4 Klasifikasi Keadaan Sekitar Alat Klasifikasi berdasarkan waktu respon alat terhadap *noise* disekitar alat saat pengujian, dapat dilihat bahwa rata-rata waktu respon saat tidak gaduh didapat waktu 5,13 detik sedangkan waktu respon saat sedikit gaduh didapat rata-ratanya 9,17 detik. Untuk pengujian saat gaduh dapat dipastikan alat tidak dapat merespon perintah suara yang diucapkan penguji. Berdasarkan data dari tabel dan pengujian berulang ditentukan batas toleransi sebagai respon tidaknya alat setelah 11 detik berlalu.

5. KESIMPULAN

A. Kesimpulan

Berdasarkan dari pengujian yang telah dilakukan maka diperoleh beberapa kesimpulan yaitu:

- Speech Recognition* dapat diterapkan pada tempat sampah otomatis dengan menggunakan *library PocketSphinx* yang menerapkan metode *Hidden Markov Model*.
- Perancangan alat yang secara terstruktur menghasilkan hasil pengukuran dan gerakan membuka serta menutup yang optimal dan akurat. Hal ini dikarenakan posisi sensor ultrasonik untuk mengukur mendapatkan tempat yang strategis untuk mengukur daya tampung tempat sampah. Selain itu lengan motor servo dapat bekerja dengan optimal karena penentuan letak dan sudut angkat tutup tempat sampah yang ringan berada di ujung dari titik sudut tumpuan, sehingga lengan servo dapat bekerja dengan baik.
- Dari klasifikasi pengujian didapat waktu respon rata-rata saat tidak gaduh (*noise* rendah) adalah 5,13 detik, sedikit gaduh (*noise* sedang) 9,17 detik dan toleransi untuk memastikan alat tidak merespon perintah suara adalah setelah 11 detik.
- Penggunaan *Speech Recognition* menggunakan *Pocketsphinx* tidak dapat digunakan pada tempat yang ramai dan gaduh.

6. DAFTAR PUSTAKA

- Alauddin, M. W., Kurniawan, W., & Setiawan, B. D. (2016). Rancang Bangun Alat Pendeteksi Suara Panggilan Manusia Berbahasa Indonesia untuk Tunarungu menggunakan Library PocketSphinx Berbasis Embedded System. 1-9.
- Arbaus, D., Prasetya, D. A., & Sari, A. P. (2016). Kecerdasan Buatan pada Sistem Pintu Otomatis menggunakan Voice Recognition berbasis Raspberry Pi. *Sistem Jurnal Ilmu Teknik*.
- Cempaka, F., Muid, A., & Ruslianto, I. (2016). Rancang Bangun Lengan Robot Sebagai Alat Pemindah Barang Berdasarkan Warna menggunakan Sensor Fotodiode. *Jurnal Coding, Sistem Komputer Untan*, 04, 57-67.
- Firdaniza, Gusriani, N., & Akmal. (2006). Hidden Markov Model. 14.
- Indriyani, F., Aryani, D., & Iskandar, D. (2018). Perancangan Smart Door Lock menggunakan Voice Recognition berbasis Raspberry Pi 3. *CERITA Journal*, 180-189.
- Lenzo, K. (2009, November 7). CMU Sphinx. Retrieved 10 10, 2018, from <http://www.speech.cs.cmu.edu/sphinx/doc/Sphinx.html>
- Multahada, Muid, A., & Ilhamsyah. (2016). Rancang Bangun Sistem Kunci Otomatis Kendaraan Bermotor Berbasis Mikrokontroler Menggunakan RFID. *Jurnal Coding, Sistem Komputer Untan*, 04, 129-137.
- Mutohar, A. (2007). Voice Recognition. *ITB Journal*, 13.
- Ramadhan, A. I., Triyanto, D., & Ruslianto, I. (2016). Pengembangan Sistem Parkir Otomatis menggunakan ARDUINO MEGA 2560 berbasis Website. *Jurnal Coding, Sistem Komputer Untan*, 184-194.
- Siregar, K. W., Triyanto, D., & Nirmala, I. (2018). Sistem Monitoring Dan Kontrol Pemakaian Air Pada Kamar Kos Menggunakan Teknologi Wireless Sensor Network Berbasis Website. *Jurnal Coding, Sistem Komputer Untan*, 184-194.
- Tomasua, S., Triyanto, D., & Nirmala, I. (2016). Sistem Kendali dan Monitoring Penggunaan Peralatan Listrik di Rumah menggunakan Raspberry Pi dan Web Service. *Jurnal Coding, Sistem Komputer Untan*, 85-96.
- Widiyanto, D. S., & Murti, B. B. (2014). Tempat Sampah Buka Tutup Otomatis. *Electronic Theses & Dissertation (ETD) | Gadjah Mada University*, 86.
- Zwass, V. (2016, February 10). Encyclopedia Britannica, inc. Retrieved 10 10, 2018, from Encyclopedia Britannica: <https://www.britannica.com/technology/speech-recognition>