



# PETIR

## JURNAL PENGKAJIAN DAN PENERAPAN TEKNIK INFORMATIKA

VOLUME 8 - NOMOR 1

MARET 2015

ISSN 1978-9262

PENERAPAN METODE *CERTAINTY FACTOR* DALAM MENENTUKAN MAKANAN YANG DIKONSUMSI BERDASARKAN KONDISI DAN KEBUTUHAN STANDAR TUBUH MANUSIA

*Abdul Haris; Azizah Ekarini*

SISTEM PENUNJANG KEPUTUSAN PUSAT KESEHATAN MASYARAKAT KUMUH PADAT KUMUH MISKIN BERDASARKAN DATA MINING MENGGUNAKAN DATA WAREHOUSE

*Hendra; Astriana Mulyani*

RANCANG BANGUN MODEL SISTEM *CONTROLLING* DAN OTOMATISASI ROBOT PENGANGKUT SAMPAH DALAM RUANGAN

*Riki Ruli A. Siregar; Suyanto*

IMPLEMENTASI DAN ANALISA JARINGAN SARAF TIRUAN DENGAN *FEATURE NORMALIZATION* DAN *PRINCIPAL COMPONENT ANALYSIS* UNTUK *DIGIT CLASSIFIER*

*Nur Abdul Wahid; Sarwo; Adi Wahyu Setiawan*

DETEKSI MATA *REAL TIME* MENGGUNAKAN *OPENCV* UNTUK *ANDROID*

*Yustika Erliani; Bagus Priambodo*

APLIKASI PENJUALAN DAN PERSEDIAAN BARANG

*Riyan Maulana; Novrini Hasti*

RANCANG BANGUN APLIKASI SISTEM PENDUKUNG KEPUTUSAN PENYELEKSIAN SISWA KELAS AKSELERASI DENGAN METODE *NAIVE-BAYESIAN*

*Yasni Djamain; Dwitya Khresna Evamandasari*

ANALISIS DAN IMPLEMENTASI TEKNOLOGI *ADAPTIVE BITRATE STREAMING* TERHADAP KONDISI *BANDWIDTH* (STUDI KASUS : VALU TV (PT DISTRIBUSI MEDIA TEKNOLOGI))

*Indra Iriyanti; Arini; Defiana Arnaldy*

IMPLEMENTASI ALGORITMA SIDIK JARI AUDIO UNTUK MENDETEKSI DUPLIKAT LAGU

*Raka Yusuf; Harni Kusniyati; Erick Estrada*

APLIKASI SISTEM PENDUKUNG KEPUTUSAN PEREKRUTAN CALON PEGAWAI NEGERI SIPIL (CPNS) DI KEMENTERIAN PERDAGANGAN RI PADA TES KOMPETENSI BIDANG (TKB) DENGAN METODE *ANALYTIC NETWORK PROCESS (ANP)*

*Rahma Farah Ningrum; Romadhona Akbar Hady*

SISTEM INFORMASI PELAYANAN KESEHATAN DI LABORATORIUM KLINIK PARANIDA BANDUNG

*Deasy Permatasari; Fanji Wijaya*

IMPLEMENTASI DAN PENGUJIAN KINERJA ORACLE 10g *REAL APLICATION CLUSTER (RAC)* PADA SISTEM OPERASI SUN SOLARIS 10

*Gatot Budi Santoso; Yanuar Indra Wirawan*

ISSN 1978-9262



771978 926272

SEKOLAH TINGGI TEKNIK - PLN (STT-PLN)

PETIR

VOL. 8

NO. 1

HAL. 1 - 132

JAKARTA, MARET 2015

ISSN 1978-9262

# RANCANG BANGUN MODEL SISTEM CONTROLLING DAN OTOMATISASI ROBOT PENGANGKUT SAMPAH DALAM RUANGAN

Riki Ruli A. Siregar<sup>1</sup>, Suyanto<sup>2</sup>  
Jurusan Teknik Informatika  
Sekolah Tinggi Teknik PLN Jakarta  
[ruliriki@gmail.com](mailto:ruliriki@gmail.com)<sup>1</sup>, [suyantoblog@gmail.com](mailto:suyantoblog@gmail.com)<sup>2</sup>

## Abstrak

*Sampah menjadi persoalan yang cukup serius. Selama ini masyarakat membuang begitu saja sampah ke tempat-tempat sampah dan menyerahkan urusan selanjutnya kepada petugas kebersihan dan urusan selesai. Tetapi sesungguhnya permasalahan tidak selesai sampai di situ, Timbunan sampah di suatu tempat menyebabkan problem tersendiri, problem kesehatan, pencemaran dan keindahan lingkungan. Sampah juga akan mempengaruhi terhadap kebersihan yang ada disuatu ruangan dimana semakin banyaknya konsumsi rumah tangga maka akan semakin banyak pula sampah yang akan dihasilkan. Tujuan dari penulisan ini adalah merancang sebuah model sistem kontrolling dan otomatisasi robot pengangkut sampah yang digunakan untuk mengangkut sampah dari ruangan hingga luar ruangan. Metode yang digunakan pada penelitian ini berupa pendekatan sistematis dari model sistem otomatisasi. Hasil dari perancangan ini diharapkan kerja manusia dalam pengangkutan sampah dari dalam ruangan hingga luar ruangan dapat terbantu oleh robot, sehingga akan mempermudah dalam pekerjaan manusia.*

**Kata kunci:** *Android, Kontroling, Robotika, Sampah.*

## Abstract

*Trash has been a serious problem. All this time, people is used to just throw away their garbage to garbage cans and leave the res of the process to instances that are responsible for taking care the daily waste of human's lives. In real life, garbage problems extend to the point of health and environment issues. Garbage also affects the cleanliness of a room, in which the more household consumption the more garbage is being created. The purpose of this writing is to design a controlling system model and a garbage robot automation. The result of this design is expected to help humans to carry their garbage out of their rooms.*

**Keyword :** *Android, Controlling, Robotic, Trash.*

## 1. Pendahuluan

### 1.1 Latar Belakang

Sampah merupakan materi atau zat, baik yang bersifat organik maupun anorganik yang dihasilkan dari setiap aktivitas manusia. Aktivitas bisa dalam rumah tangga, industri, maupun kegiatan komersial.

Sampah menjadi persoalan yang cukup serius. Selama ini masyarakat membuang begitu saja sampah ke tempat-tempat sampah dan menyerahkan urusan selanjutnya kepada petugas kebersihan dan urusan selesai. Tetapi sesungguhnya permasalahan tidak selesai sampai di situ, Timbunan sampah di suatu tempat menyebabkan problem tersendiri,

problem kesehatan, pencemaran dan keindahan lingkungan.

Sampah juga akan berpengaruh besar jika pembuangan sampah tidak dilakukan secara rutinitas apalagi didalam sebuah ruangan yang nantinya akan menyebabkan bau menyegat dan menimbulkan penyakit, hal ini disebabkan adanya sifat kemalasan dari manusia yang berhubungan dengan sampah dan identik kotor. Akibat sampah yang kotor manusia akan enggan untuk memungut bahkan membuang sampah yang berserakan sehingga akan menyebabkan timbunan sampah yang menumpuk.

Untuk menanggulangi permasalahan ini, sangat dibutuhkan sebuah teknologi otomati-

sasi untuk mengangkut sampah yang didukung oleh kepedulian manusia itu sendiri. Pengelolaan pengangkutan sampah dengan menggunakan teknologi ini diharapkan dapat membantu kinerja manusia dalam pengelolaan maupun pembuangan sampah, sehingga sistem pengangkutan lebih efektif dan efisien.

Selain itu teknologi yang bisa kita terapkan adalah dengan mengontrol sistem otomatisasi tersebut menggunakan handphone yang memiliki OS android dimana semakin banyaknya teknologi android yang beredar dipasaran akan memudahkan dalam pengoperasian gerak robot.

### 1.2 Rumusan Masalah

Berdasarkan uraian latar belakang diatas, perancangan model sistem otomatisasi ini akan mencari jawaban dari pertanyaan-pertanyaan sebagai berikut:

- a. Bagaimana penggunaan proses sistem dan metode dalam penerapan model sistem otomatisasi ?
- b. Bagaimana sensor, *voice recorder* dan remote itu terintegrasi ketika sistem dikontrol melalui handphone android ?

### 1.3 Batasan Masalah

Agar pembahasan dari penelitian ini tidak menyimpang dari apa yang telah dirumuskan diatas, maka diperlukan batasan-batasan. Batasan-batasan masalah dalam penelitian ini adalah:

- a. Perancangan *software android* menggunakan *software/framework android appinventor*.
- b. Perancangan gerak robot meliputi tiga mode yaitu otomatis, remote dan *voice*.
- c. Robot hanya menggunakan satu sensor (*ultrasonik*) untuk mode otomatis.
- d. Mode *voice* membutuhkan koneksi internet untuk mengambil data dari *google voice*.
- e. Koneksi yang digunakan hanya menggunakan *bluetooth*.
- f. Rute untuk mode otomatis sudah ditetapkan sebelumnya.
- g. Koneksi yang masuk ke robot melalui *bluetooth* hanya bisa satu akun yang terkoneksi.

### 1.4 Tujuan

Adapun tujuan penulis adalah untuk membuat model sistem otomatisasi yang nantinya dapat membantu manusia dalam proses pengangkutan sampah keluar ruangan.

### 1.5 Metode Penelitian

Metode yang digunakan pada penelitian ini berupa pendekatan sistematis dari model sistem otomatisasi .

## 2. Landasan Teori

### 2.1. Robot

Robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dahulu. Robot biasanya digunakan untuk tugas berat, berbahaya, pekerjaan berulang dan kotor. Kebanyakan robot digunakan dalam bidang industri, sebagai contoh untuk pembersihan limbah beracun, penjelajahan dawah air dan ruang angkasa. Tapi seiring dengan perkembangan, robot sudah mulai masuk dalam bidang hiburan dan bidang pendidikan, alat pembantu rumah tangga, seperti penyedot debu dan pemotong rumput.

Robotika merupakan bidang dinamis yang perkembangannya maju pesat. Perkembangan ini selain melibatkan komputasi, permesinan dan elektronika juga menyangkut perkembangan teknologi terapan. Penelitian dibidang terakhir ini biasanya berakar dari industri, untuk memecahkan masalah industri dengan teknologi yang ada. Misalnya adalah pengembangan perangkat lunak untuk mendapatkan algoritma baru bagi pengendalian robot, pengembangan sistem penglihatan dengan sistem resolusi yang lebih tinggi, perbaikan kemampuan sensor dan pengembangan protokol komunikasi untuk komunikasi dengan komputer dan peralatan pabrik sehingga robot diasumsikan sebagai gabungan antara perangkat mekanik dan perangkat elektronik yang berfungsi untuk menggantikan pekerjaan manusia yang beresiko tinggi, seperti pekerjaan pada temperatur yang tinggi, zat kimia, lingkungan kotor, dan pada kondisi yang tidak mungkin dikerjakan oleh manusia. Ada juga robot sebagai alat hiburan dan ada pula robot yang bertugas untuk menggantikan pekerjaan yang menuntut keahlian (*accuracy*), kecepatan dan lain-lain.

### 2.2. Ultrasonik

Sensor Ultrasonik adalah alat elektronika yang kemampuannya bisa mengubah dari energi listrik menjadi energi mekanik dalam bentuk gelombang suara *ultrasonic*. Sensor ini terdiri dari dua bagian yaitu *transmitter* dan *receiver* dengan komponen yang sama yaitu

*piezoelectric transducer*. Komponen ultrasonik ini bekerja berdasarkan prinsip speaker konvensional. Hanya frekuensi fungsi responnya dibatasi secara tajam, yaitu pada 40 KHz bila difungsikan sebagai speaker, ia akan mengeluarkan suara ultra dengan frekuensi 40 KHz, sedangkan bila difungsikan sebagai microphone, ia akan hanya memproses sinyal suara yang mempunyai frekuensi 40 KHz. Seperti yang kita ketahui suara ultra tidak akan dapat didengar oleh telinga manusia. Frekuensi ini juga tidak efisien dipancarkan ke "space" dengan cara resonansi pancaran gelombang radio karena akan memerlukan antena yang sangat besar dan panjang.

Pada dasarnya sensor ini digunakan juga untuk uji kondisi dua keadaan, yaitu "ada atau tidak adanya ultrasonik". Dalam aplikasi, metode transmisi dan "switching" informasi terimanya bermacam-macam. Namun, sebagian besar, 'state' atau keadaan yang dijadikan pedoman uji tetap menggunakan uji dua kondisi tersebut.



Gambar 1. Sensor Ultrasonik

### 2.3. Android

*Android* adalah sistem operasi untuk telepon seluler yang berbasis Linux. *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri sehingga dapat digunakan oleh bermacam peranti bergerak. Awalnya Google Inc. Membeli *Android Inc.* pendatang baru yang membuat software (perangkat lunak) untuk telepon genggam. Kemudian untuk mengembangkan *Android* di bentuklah *Open Handset Alliance* yang merupakan gabungan dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi termasuk *Google, HTC, Intel, Motorola, Qualcomm, TMobile, dan NVidia.*

*Android* memiliki berbagai keunggulan sebagai software yang memakai basis kode komputer yang bisa didistribusikan secara terbuka (*open source*) sehingga pengguna bisa membuat aplikasi baru di dalamnya. *Android* memiliki aplikasi native *Google* yang

terintegrasi seperti *pushmail Gmail, Google Maps, dan Google Calendar.*

Para penggemar *open source* kemudian membangun komunitas yang membangun dan berbagi *Android* berbasis *firmware* dengan sejumlah penyesuaian dan fitur-fitur tambahan, seperti *FLAC lossless audio* dan kemampuan untuk menyimpan download aplikasi pada *microSD card*. Mereka sering memperbaharui paket-paket *firmware* dan menggabungkan elemen-elemen fungsi *Android* yang belum resmi diluncurkan dalam suatu *carrier-sanction firmware.*

### 2.4. Arduino

*Arduino* adalah kit elektronik atau papan rangkaian elektronik *opensource* yang di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan jenis *AVR* dari perusahaan *Atmel.* *Arduino* dapat digunakan untuk mengembangkan objek interaktif, mengambil masukan dari berbagai *switch* atau sensor, dan mengendalikan berbagai lampu, motor, dan *output* fisik lainnya. Proyek *arduino* dapat berdiri sendiri, atau dapat berkomunikasi dengan perangkat lunak yang berjalan pada komputer. *Board* dapat dirakit sendiri atau dengan cara membeli, dan *IDE opensource* dapat didownload secara gratis.

*Arduino* mikrokontroler adalah komentar kecil untuk perintah kontrol. Dengan beberapa baris kode membuat *Arduino* mengubah cahaya atau mematikan, membaca nilai sensor dan menampilkannya di layar komputer, atau bahkan menggunakannya untuk membangun sebuah sirkuit buatan sendiri untuk memperbaiki alat yang rusak. Karena fleksibilitas dari *Arduino* dan dukungan besar tersedia dari *online* komunitas pengguna *Arduino,* telah menarik generasi baru penggemar elektronik yang belum pernah sebelum menyentuh mikrokontroler.

Bahasa pemrograman *arduino* merupakan implementasi dari wiring, sebuah platform komputasi fisik, yang didasarkan pada lingkungan pemrograman *processing multimedia.*

Ada banyak mikrokontroler lain dan platform mikrokontroler yang tersedia untuk komputasi fisik. *Parallax Basic Stamp, Netmedia BX-24, Phidget, Handyboard MIT,* dan banyak lagi lainnya yang menawarkan fungsionalitas yang sama. Semua alat ini menggunakan pemrograman mikrokontroler yang berantakan dan membingkusnya dalam

paket yang mudah digunakan. Mikrokontroler itu sendiri adalah chip atau IC (*integrated circuit*) yang bertugas membaca dan memproses input dari sebuah rangkaian elektronik sehingga menghasilkan *output* sesuai dengan yang diinginkan. Mikrokontroler ini bisa diprogram menggunakan komputer.

Berikut adalah beberapa kelebihan dari Arduino :

- **Hardware dan Softwarena Open Source**

Dari sini kita bisa membuat tiruan *board* yang kompatibel dengan *board* Arduino tanpa harus membeli *board* asli buatan Itali dan kita juga tidak akan dianggap membajak selama kita tidak menggunakan trademark "Arduino".

- **Fasilitas chip yang cukup lengkap**

Arduino menggunakan chip AVR ATmega 168/328 yang memiliki fasilitas PWM, komunikasi serial, ADC, timer, interupt, SPI dan I2C. Dengan fasilitas chip yang demikian, Arduino bisa digabungkan dengan modul atau alat lain walaupun protokol yang digunakan berbeda-beda.

- **Proses Upload tidak memerlukan chip programmer**

Chip pada Arduino sudah dilengkapi dengan *bootloader* yang akan menangani proses *upload* dari komputer. Dengan begitu kita tidak memerlukan chip programmer kecuali untuk menanamkan *bootloader* pada chip yang masih *blank*.

- **Ukuran board kecil**

Ukuran *board* Arduino yang kecil ini mudah di bawah kemana-mana atau dimasukkan ke dalam saku atau tas yang kecil.

- **Koneksi menggunakan Port USB**

Ini akan memudahkan kita jika menghubungkan Arduino ke PC atau laptop yang tidak memiliki port serial/paralel.

- **Bahasa pemrograman yang mudah**

Bahasa pemrograman Arduino adalah bahasa C yang sudah dipermudah menggunakan fungsi-fungsi yang sederhana sehingga dapat dipelajari dengan mudah.

- **Library gratis**

*Library-library* ini dapat di download gratis di website Arduino.

- **Pengembangan aplikasi lebih mudah**

Pengembangan aplikasi ini menjadi lebih mudah karena didukung oleh bahasa yang mudah dipelajari serta adanya *library* dasar yang lengkap.



Gambar 2. Arduino Uno

## 2.5. Driver Motor

Motor DC tidak dapat dikendalikan secara langsung oleh mikrokontroler, karena kebutuhan arus listrik yang besar pada motor DC sedangkan arus keluaran pada mikro sangat kecil. Driver motor merupakan pilihan alternatif yang harus digunakan untuk mengendalikan motor DC pada robot beroda. Ada beberapa driver motor yang sering digunakan pada aplikasi robotika, yaitu menggunakan rangkaian H-Bridge transistor, H-Bridge MOSFET, dan IC driver motor.

Motor DC adalah suatu piranti elektronik yang berfungsi untuk mengubah energi listrik menjadi energi mekanik. Pada aplikasi robotika pergerakan robot beroda umumnya menggunakan motor DC sebagai alat penggerakannya, karena jenis motor ini lebih mudah untuk dikendalikan. Kecepatan yang dihasilkan oleh motor DC berbanding lurus dengan potensial yang diberikan.

Pengaturan arah putaran motor dilakukan dengan mengubah arah polaritas yang mengalir melalui motor



Gambar 3. Driver Motor

## 2.6. DC Motor

DC Motor umum yang menggunakan sikat (*brush*) yang menggunakan lilitan pada rotor dan menggunakan magnet tetap pada sisi stator, pada dasarnya dapat dianggap sebagai suatu beban yang dapat dihubungkan langsung ke rangkaian *switch*-

ing arus DC. Oleh karena itu, pemilihan yang tepat cukup diperoleh dengan memperhatikan besar kebutuhan arus untuk memutar DC Motor secara nominal. Lilitan pada DC Motor dapat didentikkan dengan lilitan pada kumparan relay sehingga rangkaian relatifnya sama.

Pada beberapa kasus, seringkali diperlukan arah putaran DC Motor yang berubah-ubah. Prinsip dasar untuk mengubah arah putarannya ini adalah dengan membalik polaritas pada catu tegangannya.

Pada DC Motor, daerah kumparan medan yang dialiri arus listrik akan menghasilkan medan magnet yang melingkupi kumparan jangkar dengan arah tertentu. Konversi dari energi listrik menjadi energi mekanik (motor) maupun sebaliknya berlangsung melalui medan magnet, dengan demikian medan magnet disini selain berfungsi sebagai tempat untuk menyimpan energi, sekaligus sebagai tempat berlangsungnya proses perubahan energi.

Agar proses perubahan energi mekanik dapat berlangsung secara sempurna, maka tegangan sumber harus lebih besar daripada tegangan gerak yang disebabkan reaksi lawan. Dengan memberi arus pada kumparan jangkar yang dilindungi oleh medan maka menimbulkan perputaran pada motor.

Dalam memahami sebuah motor, penting untuk mengerti apa yang dimaksud dengan beban motor. Beban dalam hal ini mengacu kepada keluaran tenaga putar / *torque* sesuai dengan kecepatan yang diperlukan.



Gambar 4. DC Motor

## 2.7. App Inventor

App Inventor adalah sebuah tool untuk membuat aplikasi android, yang menyenangkan dari tool ini adalah karena berbasis visual *block programming*.

*Framework visual programming* ini terkait dengan bahasa pemrograman Scratch dari MIT, yang secara spesifik merupakan implementasi dari Open Block yang didistribusikan oleh MIT Scheller Teacher Education Program yg diambil dari riset yang dilakukan oleh Ricarose Roque. App Inventor

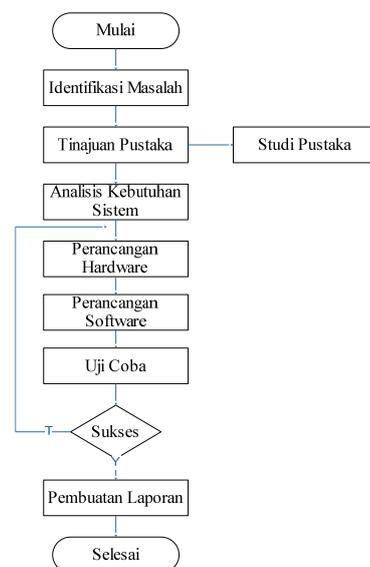
menggunakan Kawa Language Framework dan Kawa's dialect – yg di develop oleh Per Bothner dan di distribusikan sebagai bagian dari GNU Operating System oleh Free Software Foundation sebagai *Compiler* yang mentraslate *visual block programming* untuk diimplementasikan pada *platform Android*.

Dalam App Inventor dikenal tiga istilah, yaitu :

1. *Designer*  
*Designer* adalah *workspace* mendesain tampilan aplikasi android. *Designer* berjalan di *web browser*, cara menggunakan *designer* yaitu *click and drag*.
2. *Block Editor*  
*Block Editor* adalah bisa digambarkan seperti *Event Handler* dan digunakan untuk membuat program kendali android.
3. *Emulator*  
*Emulator* memungkinkan untuk mengembangkan aplikasi *mobile* dan menguji di sebuah emulator yang memiliki tampilan dan fungsi mirip dengan ponsel sebenarnya. namun, ada beberapa catatan yang perlu ketahui. Emulator hanyalah sebuah emulator dengan segala keterbatasannya, sehingga tidak bisa berpatokan penuh dengan hasil yang ditampilkan di emulator sama jika sudah diexport dan menginstall aplikasi di ponsel Android.

## 3. Metodologi Penelitian

Secara umum metode penelitian dapat dilihat pada diagram alir dibawah ini:



Gambar 5 Kerangka Penelitianan

Proses dimulai dengan Tahap Mulai yang menandakan kegiatan akan dilaksanakan. Yang dilanjutkan dengan Pengumpulan Data dan Informasi.

Kemudian dilakukan Persiapan Komponen Hardware & Software. Setelah itu dilanjutkan dengan Perancangan yang dibagi menjadi Aplikasi dan Pembuatan Prototype.

Kemudian dilakukan Uji Coba & Evaluasi, jika ada masalah maka proses akan kembali ke Perencanaan dan jika tidak ada masalah maka proses akan dilanjutkan dengan Dokumentasi dan Penulisan Laporan.

### 3.1. Persiapan Software

Persiapan *software* yang ada berupa:

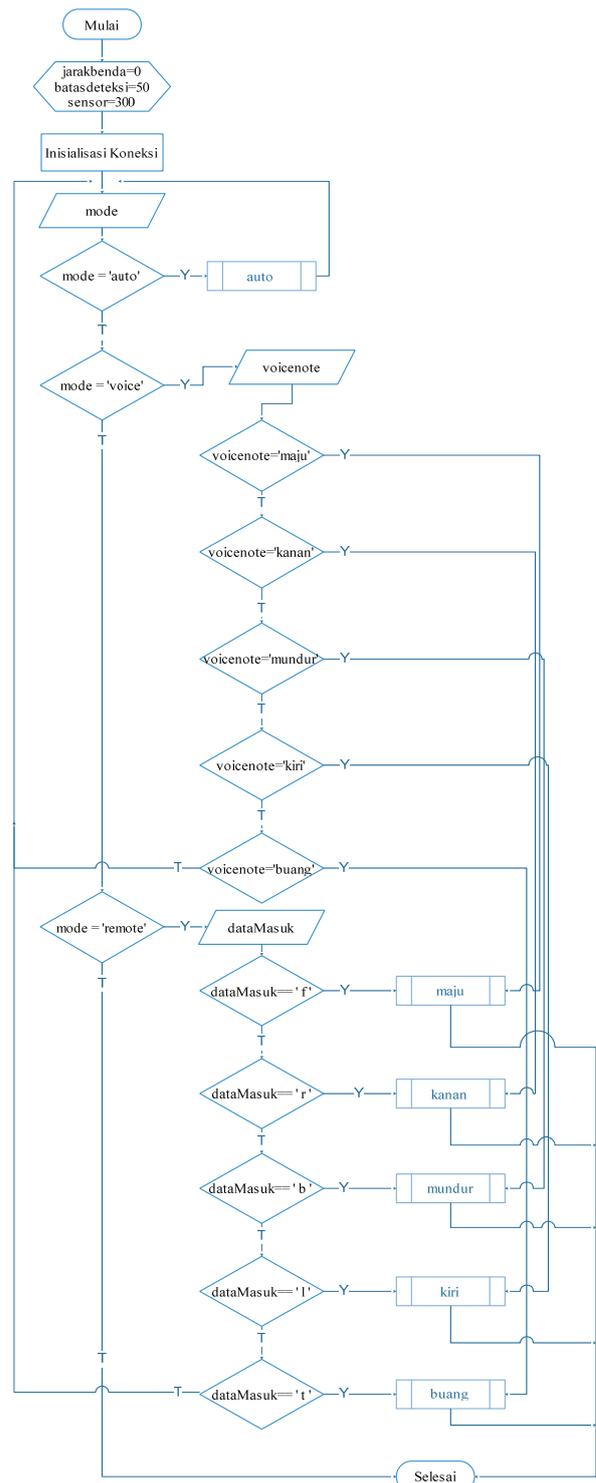
1. Arduino IDE 1.5  
Arduino IDE digunakan untuk membuat program pada Arduino.
2. App Inventor  
Digunakan untuk membuat aplikasi pengontrol robot untuk android.
3. Microsoft Office Excel  
Digunakan untuk menghitung matriks yang akan digunakan untuk menghitung arah robot.
4. Matlab 7.6  
Digunakan untuk membuat jaringan syaraf tiruan sebagai arah kerja robot

### 3.2. Persiapan Hardware

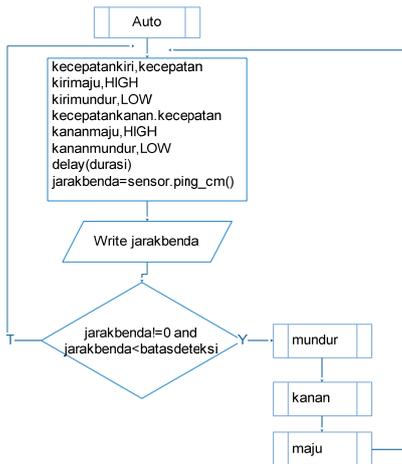
Persiapan *hardware* yang ada berupa:

1. Arduino Uno  
Digunakan sebagai mikrokontroler menggerakkan seluruh kinerja robot yang dikontrol.
2. Sensor Ultrasonik  
Sebagai sensor pendeteksi jarak dan penghalang yang ada didepan robot.
3. Motor Servo  
Digunakan sebagai penggerak untuk menuang sampah.
4. Bluetooth  
Digunakan sebagai koneksi antara handphone dengan robot.
5. Driver Motor  
Digunakan untuk menggerakkan motor DC.
6. Motor DC  
Digunakan untuk menggerakkan robot
7. Handphone OS Android  
Digunakan untuk mengontrol gerak robot

Algoritma pengoprasian gerak robot



Gambar 6. Flowchart Gerak Robot 1



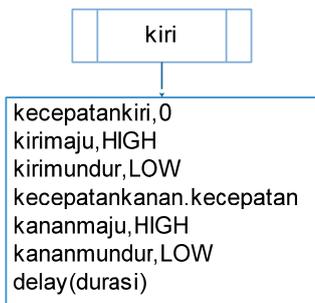
Gambar 7. Flowchart Gerak Robot 2



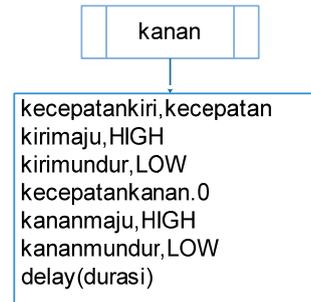
Gambar 8. Flowchart Gerak Robot 3



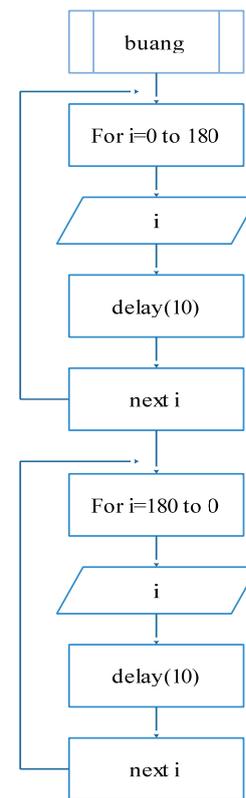
Gambar 9. Flowchart Gerak Robot 4



Gambar 10. Flowchart Gerak Robot 5



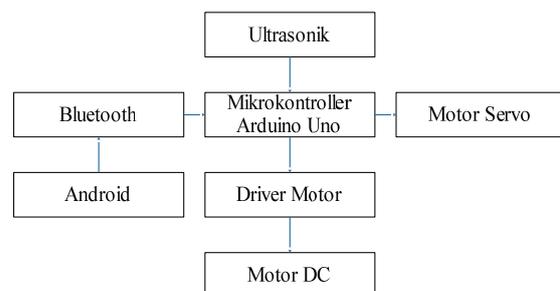
Gambar 11. Flowchart Gerak Robot 6



Gambar 12. Flowchart Gerak Robot 7

### 3.3. Perancangan Hardware

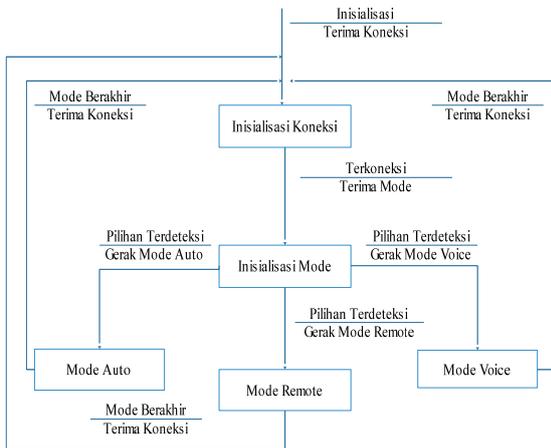
Pada perancangan ini penulis membuat suatu tampilan perancangan koneksi antar komponen yang digunakan :



Gambar 13. Rancangan Interkoneksi hardware

Pembuatan rancangan di atas bertujuan untuk mensimulasikan pembuatan rangkaian sesungguhnya sehingga hasilnya sesuai dengan yang diinginkan.

State Transition Diagram Android

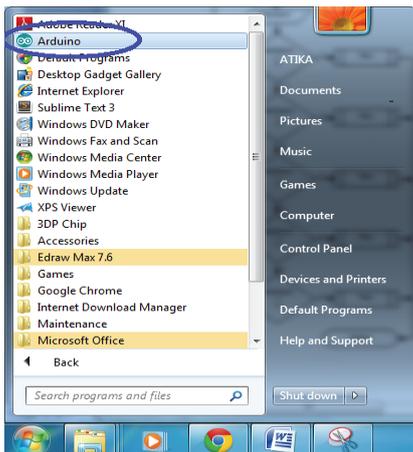


Gambar 14. diagram STD Menu Android

### 3.4. Konfigurasi Arduino 1.5

Konfigurasi pertama kali adalah melakukan instalasi program untuk menghubungkan Arduino dengan komputer (OS Windows). Langkahnya ialah:

- Install aplikasi Arduino.
- Setelah selesai diinstal, maka akan muncul logo aplikasi Arduino IDE seperti gambar berikut:



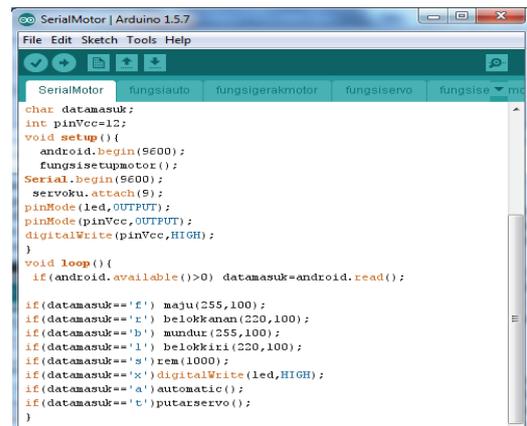
Gambar 15. Tampilan Aplikasi Arduino IDE

### 3.5. Pembuatan Program Arduino Uno

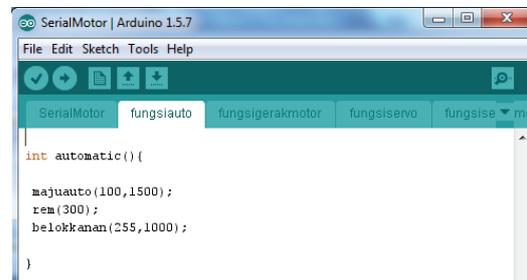
Proses ini, dilakukan dengan menyambungkan PC ke Arduino UNO menggunakan kabel USB. Adapun langkah-langkahnya adalah sebagai berikut:

1. Hubungkan port kabel USB dengan port Arduino.
2. Buka program Arduino 1.5 IDE.
3. Buka file *serialmotor.ino* yang sudah dibuat sebelumnya.
4. Ubah tipe board nya menjadi *Arduino Uno*
5. Ubah juga Serial Port nya menjadi Port yang sesuai dengan *driver arduino*.

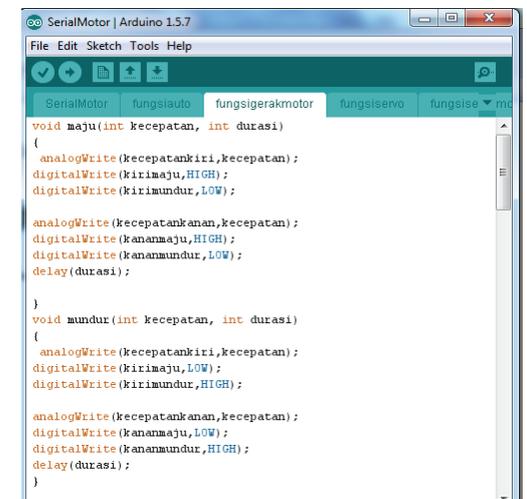
Kemudian klik "Upload" atau tekan tombol Ctrl+U.



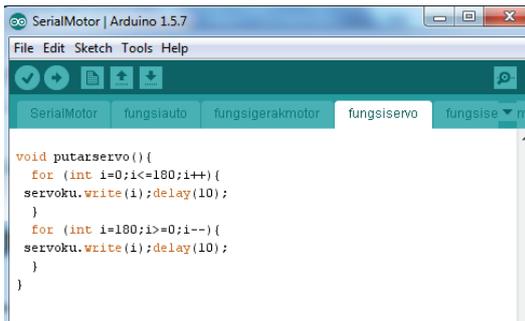
Gambar 16. Program serialmotor.ino 1



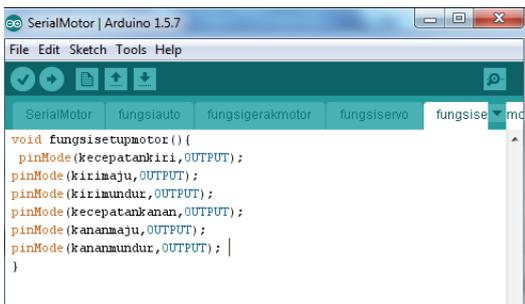
Gambar 17. Program serialmotor.ino 2



Gambar 18. Program serialmotor.ino 3



Gambar 19. Program serialmotor.ino 4



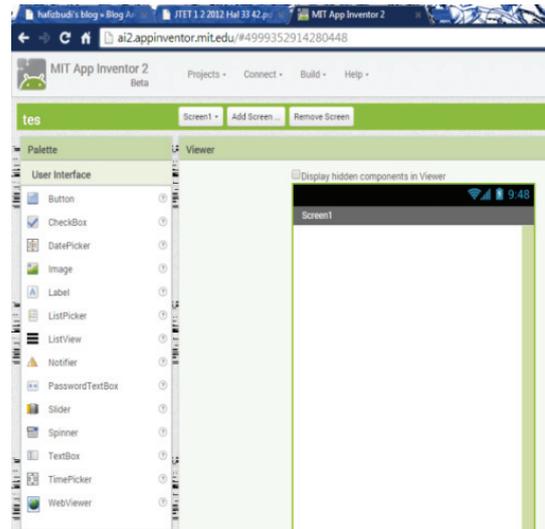
Gambar 20. Program serialmotor.ino 5



Gambar 21. Program serialmotor.ino 6

### 3.6. Konfigurasi App Inventor

Konfigurasi untuk membuka aplikasi inventor dan melakukan pengkodean adalah dengan masuk melalui akun google plus, dan selanjutnya masuk ke <http://www.ai2.appinventor.mit.edu> sehingga akan menampilkan konfigurasi berikut :



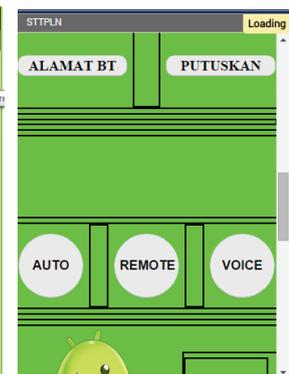
Gambar 22. Tampilan pembuat program android

### 3.7. Pembuatan Program Android

Proses ini dilakukan dengan membuat desainer terlebih dahulu sehingga menghasilkan tampilan seperti berikut :



Gambar 23. Desain android 1



Gambar 24. Desain android 2



Gambar 25. Desain android 2

Selanjutnya adalah membuat baris program yang menggunakan kode block seperti berikut:

This screenshot shows several blocks of code in Android Studio. At the top, there's a 'do' block that sets a list picker to Bluetooth addresses and names. Below it, a 'when ListPicker1 AfterPicking' block calls BluetoothServer1 to accept a connection, sets a text field to 'Connected', and enables a disconnect button. To the right, a 'when Screen1 Initialize' block sets a global screen to 'Cover' and calls 'lakukanScreen'. At the bottom, a 'when Screen1 BackPressed' block checks if a screen cover is visible and either closes the application or toggles the screen cover's visibility.

Gambar 26. Kode Block Android 1

This screenshot displays code blocks for managing screen visibility. It features multiple 'if' and 'else if' conditions that check the visibility of various screen elements like 'ScreenVoice', 'ScreenAuto', 'ScreenRemote', 'ScreenUtama', and 'ScreenCover'. When a screen is visible, it sets other screens to be invisible and vice versa. A 'do' block at the top right is labeled 'lakukanScreen' and contains logic to compare global screen states and set the visibility of different screen components.

Gambar 27. Kode Block Android 2

This screenshot shows code blocks for speech recognition and timer control. A 'when SpeechRecognizer1 AfterGettingText' block checks for specific words like 'MAJU', 'MUNDUR', and 'kanan', and sends text to BluetoothClient1. To the right, a 'when Clock2 Timer' block checks if a global variable is 0 and enables a timer to open another screen. Another 'when Clock1 Timer' block checks if BluetoothClient1 is connected and sets the visibility of labels and text fields.

Gambar 28. Kode Block Android 3

This screenshot displays code blocks for speech recognition and button actions. It includes a 'then' block that sets a label to 'BELOK KIRI' and sends text to BluetoothClient1. An 'else if' block checks for the word 'stop' and sends text to BluetoothClient1. A 'when btnSpeak Click' block sends text to SpeechRecognizer1. A 'when btnStop Click' block sends text to BluetoothClient1. A 'when btnKiri TouchDown' block sends text to BluetoothClient1. A 'when btnMaju TouchDown' block is also present.

Gambar 29. Kode Block Android 4

This screenshot shows code blocks for button actions and timer control. It includes a 'when btnEnter Click' block that sets a global screen to 'Utama' and calls 'lakukanScreen'. A 'when btnSelesai Click' block sends text to BluetoothClient1. A 'when btnVoice Click' block sets a global screen to 'voice' and calls 'lakukanScreen'. A 'when btnMundur TouchUp' block sends text to BluetoothClient1. A 'when btnMundur TouchDown' block sends text to BluetoothClient1. A 'when btnStopAuto Click' block sets a timer to always fire and enables it. A 'when btnDiscon Click' block disconnects the Bluetooth server and client and sets the Bluetooth text to 'Disconnected'.

Gambar 30. Kode Block Android 5

#### 4. Hasil dan Uji Coba

Setelah melakukan analisa dan perancangan sistem, penulis akan mencoba memaparkan sebagian dari hasil penelitian mengenai *prototype* alat yang dibuat yang bertujuan untuk mengetahui kinerja alat yang dibuat.

##### 4.1. Interface Android

Tampilan atau interface yang dihasilkan dalam pemrograman android sebagai pengontrol gerak robot adalah sebagai berikut :



**Gambar 31.**  
Tampilan slash android



**Gambar 32.**  
Tampilan utama android



**Gambar 37.**  
Tampilan remote



**Gambar 38.**  
Tampilan voice



**Gambar 33.**  
Tampilan mode android



**Gambar 34.**  
Tampilan alamat BT

#### 4.2. Interface Robot

Tampilan atau interface yang dihasilkan dalam komponen robot :



**Gambar 39.**  
Robot Tampak Depan



**Gambar 40.** Robot Tampak Samping Kanan



**Gambar 35.**  
Tampilan BT Terkoneksi



**Gambar 36.**  
Tampilan auto android



**Gambar 41.** Robot Tampak Samping Kiri



**Gambar 42.** Robot Tampak Atas



Gambar 43. Robot Membuang Sampah

#### 4.3. Uji Coba

Tabel. 1 Pengujian kapasitas berat sampah

No	Berat	Kecepatan gerak	Keterangan
1	1 Kg	Normal	Mampu
2	2 Kg	Normal	Mampu
3	3 Kg	Lambat	Mampu
4	4 Kg	Lambat	Mampu
5	5 Kg	Lambat	Tidak Mampu

Untuk berat sampah hingga 5 Kg menyebabkan kecepatan gerak robot menjadi lambat dan membuat robot tidak bergerak dikarenakan adanya kurang supply daya untuk robot.

Tabel 2. Pengujian Jarak Bluetooth untuk robot

No	Jarak	Kecepatan Gerak	Keterangan
1	1 M	Normal	Mampu
2	2 M	Normal	Mampu
3	3 M	Normal	Mampu
4	4 M	Normal	Mampu
5	5 M	Normal	Mampu
6	6 M	Normal	Mampu
7	7 M	Normal	Mampu
8	8 M	Normal	Mampu
9	9 M	Normal	Mampu
10	10 M	Normal	Mampu
11	11 M	Normal	Mampu
12	12 M	Normal	Mampu
13	13 M	Normal	Mampu
14	14 M	Normal	Mampu
15	15 M	Normal	Mampu
16	16 M	Lambat	Mampu
17	17 M	Lambat	Mampu
18	18 M	Lambat	Tidak Mampu

Tabel 3. Pengujian jenis kapasistas sampah

No	Alat yang digunakan	Keterangan
1	Gabus	Mampu
2	Tripleks	Mampu
3	Tempat Sampah tinggi 22 cm	Tidak Mampu

#### 5. Kesimpulan

Dari uraian di atas, penulis dapat menarik kesimpulan yaitu:

1. Aplikasi android untuk mode voice hanya bisa digunakan jika ada koneksi data
2. Durasi akses data ke robot tergantung koneksi jaringan *bluetooth*
3. Koneksi android dengan robot hanya bisa dilakukan satu akun dalam waktu yang bersamaan
4. Supply daya yang dibutuhkan tergantung berat benda yang diangkut
5. Robot menggunakan satu sensor untuk mode otomatis dan membutuhkan google voice untuk mode voice.

#### DAFTAR PUSTAKA

1. Ardiansyah, F. (2011). *Pengenalan Dasar Android Programming*. Biraynara.
2. Budiharto, W. (2010). *Robotika Teori dan Implementasi*. Yogyakarta: Andi.
3. Desiani, A., & Arhami, M. (2006). *Konsep Kecerdasan Buatan*. Yogyakarta: Andi.
4. Mifbakhuddin, Salawati, T., & Kasmudi, A. (2010). Gambaran Pengelolaan Sampah Rumah Tangga Tinjauan Aspek Pendidikan, Pengetahuan, dan Pendapatan Perkapita di RT 6 RW 1 Kelurahan Pedurungan Tengah Semarang. *jurnal.unimus.ac.id* , Volume 6. Nomor 1.
5. Pitowarno, E. (2005). *Mikroprosesor dan Interfacing*. Yogyakarta: Andi.
6. Pitowarno, E. (2006). *Robotika, Desain, Kontrol dan Kecerdasan Buatan*. Yogyakarta: Andi.