



### **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 III

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan Dan Koordinasi

Kerja magang selama 3,5 bulan dilaksanakan di Universitas Tokyo Denki yang berlokasi di 2-1200 MuzaiGakuendai, Inzai, Chiba 270-1382, Jepang berkedudukan sebagai Programmer. Penulis ditugaskan untuk menangani sebuah project berkatitan dengan interaksi robot dan manusia bersama dengan 2 rekan lainnya. Kerja magang diawasi oleh Prof. Naoki Mukawa selaku kepala laboratorium dan dikoordinasi oleh Saito Natsuki selaku ketua *Project*.

Secara keseluruhan, sistem program robot ini merupakan perpaduan program robot, sensor suara, serta kinect selaku sensor wajah yang nantinya dapat digunakan sebagai dasar untuk mengembang project-project yang sejenis nantinya.

#### 3.2 Tugas yang dilakukan

Selama periode magang selama 3,5 bulan, penulis ditugaskan untuk membuat program pendeteksi muka dengan melalui kinect, dan program yang mendeteksi suara yang nantinya akan diintegrasikan dengan robot, serta program yang menghubungkan hasil output dari kinect dengan program robot. Penulis mendapat tanggung jawab yang sama dengan Daniel Halim yang melakukan praktek kerja magang di tempat yang sama sehingga dari tanggung jawab yang diberikan dibagi menjadi 2 sehingga penulis sendiri bertanggung jawab untuk mendeteksi wajah user melalui kinect, menetapkan posisi wajah dalam kamera kinect, mengimplementasikan program penghubung (*socket programming*) ke program robot (*server*), memproses informasi tersebut yang dikirimkan untuk kemudian ditentukan gerakan robot, serta sensor suara untuk mendeteksi suara user . Untuk mendeteksi rotasi wajah dan pengiriman data yang telah dikalkulasi ke *server* akan dikerjakan oleh Daniel Halim.

Prinsip kerja sistem yang akan dibangun adalah kinect bertugas untuk membaca wajah user dan sensor suara yang siap menerima input suara, dari informasi-informasi yang diambil dari user, program nantinya mengenerate output

berupa gerakan robot. Semua program-programnya dibuat dengan menggunakan bahasa pemrograman C++.

Pada pembuatan sistem yang ditugaskan kepada penulis, pembuatan sistem bisa dikategorikan sebagai pengembangan dari detail metode *System Development Life Cycle (SDLC)* dalam implementasinya. Oleh karena itu, untuk menggeneralisasi tugas-tugas yang diberikan selama pembuatan sistem ini, penulis akan mengacu pada fase-fase dalam *System Development Life Cycle (SDLC)*.

Tabel 3.1 timeline sistem

No	Kegiatan	September				Oktober				November				December	
		1	2	3	4	1	2	3	4	1	2	3	4	1	2
1	Studi literatur														
2	<i>Requirement analysis</i>														
3	<i>Design</i>														
4	<i>Development</i>														
5	<i>Integration and Test</i>														
6	<i>Implementation</i>														
7	<i>Operation and Maintenance</i>														
8	Penulisan laporan dan dokumentasi														

### 3.2.1 REQUIREMENT ANALYSIS

Selama periode magang ini penulis mendapat tugas untuk membuat sebuah sistem yang menghubungkan robot dengan kinect dan sensor dimana kinect nantinya akan diumpamakan sebagai mata robot sementara sensor suara diumpamakan sebagai telinga robot sehingga dari hasil output kinect dan sensor suara nantinya akan memberikan perintah bagi robot untuk bergerak.

Untuk sensor kinect penulis diminta untuk mendeteksi wajah yang berada didepan kinect, kinect akan diletakkan di depan robot, setelah itu program harus dapat memutuskan apakah user yang berada di depan kinect sedang melihat kinect atau tidak. Untuk memenuhi requirement ini maka pertama-tama program harus dapat mendeteksi setiap wajah user yang masuk ke dalam area *view* kinect. Setelah itu, program harus mendeteksi posisi wajah, dan yang ketiga adalah mendeteksi rotasi wajah. Dari semua informasi yang didapatkan barulah dapat ditentukan apakah user sedang melihat atau tidak. Program kinect tersebut nantinya akan dibangun dengan bahasa pemrograman C++.

*Requirement* kedua yang diminta adalah sistem diharuskan dapat mendeteksi suara user ketika user sedang berbicara. Untuk memenuhi *requirement* ini tentunya harus membuat sebuah sensor suara(audio) maka dari itu dalam proyek ini diputuskan untuk menggunakan mikrofon yang dihubungkan dengan arduino.

Arduino adalah semacam pengendali mikro single-board yang bersifat *open source*, diturunkan dari Wiring platform, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Berbagai kelebihan arduino seperti harga yang murah, penggunaan yang sederhana, serta *open source* membuat kelompok penulis memutuskan untuk menggunakan arduino sebagai alat bantu untuk menghubungkan mikrofon dengan PC.

Selain dari hasil analisa penggunaan hardware-hardware yang telah ditetapkan, berikut analisis *user requirement* secara teknis, dapat di tarik beberapa kesimpulan yang perlu diterapkan dalam sistem yaitu:

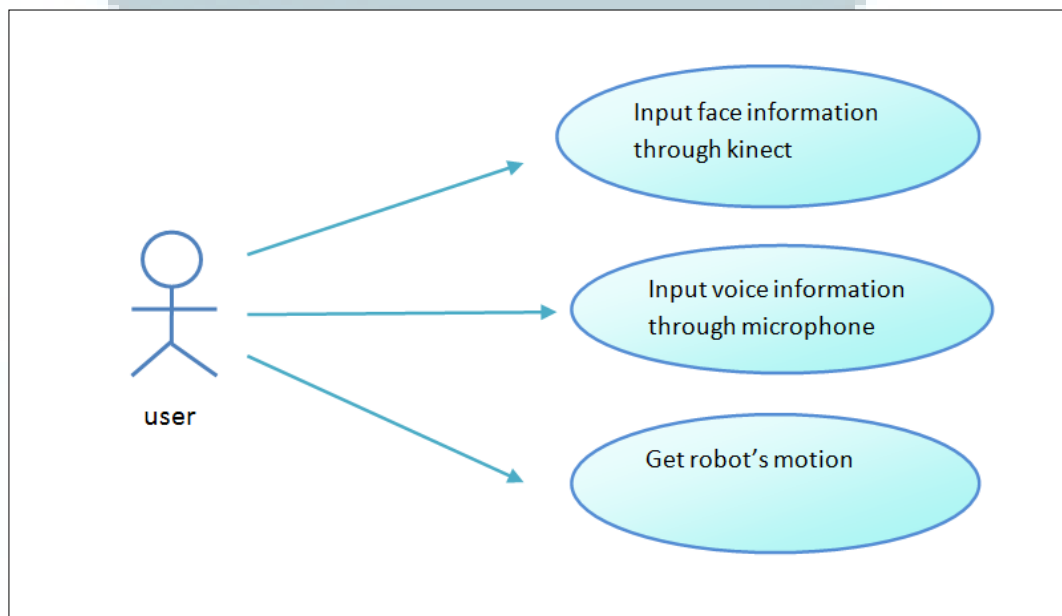
- a. Semua perangkat hardware haruslah terhubung secara *real-time* dalam arti diperlukannya sebuah program yang dapat menghubungkan kinect, sensor suara, dan robot.
- b. Masing-masing program akan memproses data terlebih dahulu sebelum mengirim atau menerima data. Teknik proses dan pengiriman data haruslah efisien dan sama cepat dikarenakan program akan selalu mengambil data dari user dalam jarak waktu yang dekat sehingga untuk menghindarinya terjadinya deadlock.

### 3.2.2 Design

Pada tahap perancangan, dilakukan pembuatan dan analisis algoritma yang dibutuhkan untuk membuat sistem ini. Untuk membuat rancangan suatu sistem, tentunya dibutuhkan gambaran dan rancangan teknis tentang alur kerja dari sistem yang akan dibuat. Pada sistem ini belum terdapat database dikarenakan seluruh proses dari input sampai dengan output dilakukan secara *real-time* sehingga tidak ada data yang perlu ditampung. Dalam pembuatan sistem robot ini digunakan *Unified Modelling Language* (UML) untuk menggambarkan detail sistem secara menyeluruh dimulai dari analisis permintaan sampai dengan dokumentasi serta menjadi patokan selama sistem dibangun. UML adalah bahasa visual yang digunakan untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek-aspek dari sebuah sistem yang dibangun.

Berikut ini adalah gambaran sistem serta algoritma yang dirancang yang nantinya akan digunakan sebagai patokan dalam pembuatan sistem.

#### a. Use Case Diagram

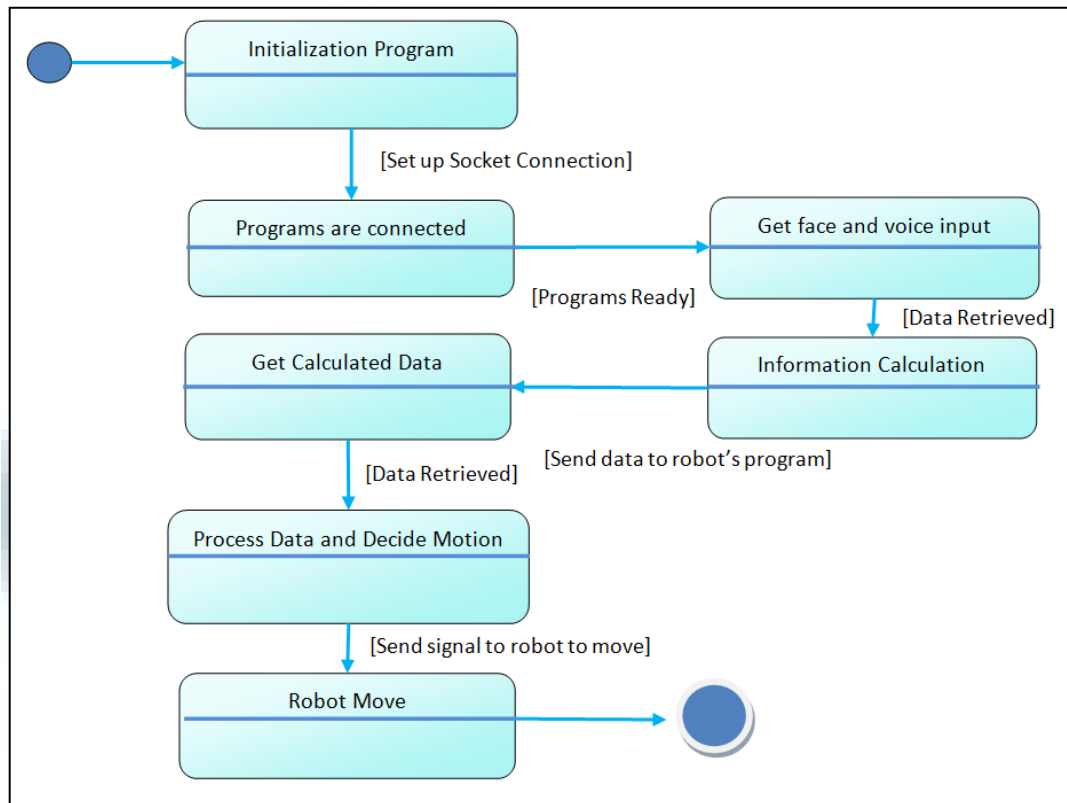


Gambar 3.1 Use case diagram

Sistem ini terdiri atas 3 aktivitas yang dapat dilakukan oleh user yang menggunakan sistem. Berikut adalah penjelasan masing-masing fungsi.

1. Aktivitas pertama adalah user dapat menginput informasi wajah ke dalam sistem. Ketika user masuk ke jarak pandang kinect maka wajah user akan secara otomatis di deteksi oleh kinect. Program kinect akan secara otomatis mengambil informasi tentang posisi dan gerakan kepala user. Nantinya dengan informasi posisi kepala dan rotasi kepala informasi tersebut akan menghasilkan kesimpulan apakah user yang terdeteksi sedang memandang ke arah kinect atau tidak.
2. Sistem juga menyediakan fungsi untuk mendeteksi suara sekitar. Nantinya sistem akan menyediakan sebuah mikrofon yang nantinya akan digunakan untuk menangkap suara user yang tertuju pada mikrofon.
3. User nantinya dapat melihat output yang berupa gerakan robot yang dihasilkan dari informasi kepala dan suara yang diinput dari user. Pada sistem ini gerakan robot hanya terbatas pada kepala saja, jika user melihat ke arah kinect dengan asumsi kinect terletak tepat di depan robot, maka kepala robot nantinya akan melihat ke arah user. Sementara dari suara user yang terdeteksi maka robot akan melakukan gerakan mengangguk.

## b. State Chart Diagram



Gambar 3.2 State chart diagram

State chart diagram diatas menyediakan informasi gambaran sistem yang akan dibuat secara garis besar. Tahapan awal sebelum sistem siap digunakan adalah inisialisasi program-program. Di tahapan ini juga terdapat inisialisasi hubungan antar program yang disambungkan melalui socket. Jadi nantinya program *server* (program robot) akan dijalankan terlebih dahulu barulah program *client* dijalankan. Nantinya akan terjadi komunikasi dimana program *client* akan terkoneksi dengan program *server*.

Setelah sistem siap digunakan maka sistem akan menunggu user yang akan menggunakan sistem. Sistem nanti akan mendeteksi wajah user melalui kinect, jika terdeteksi maka sistem akan mengambil

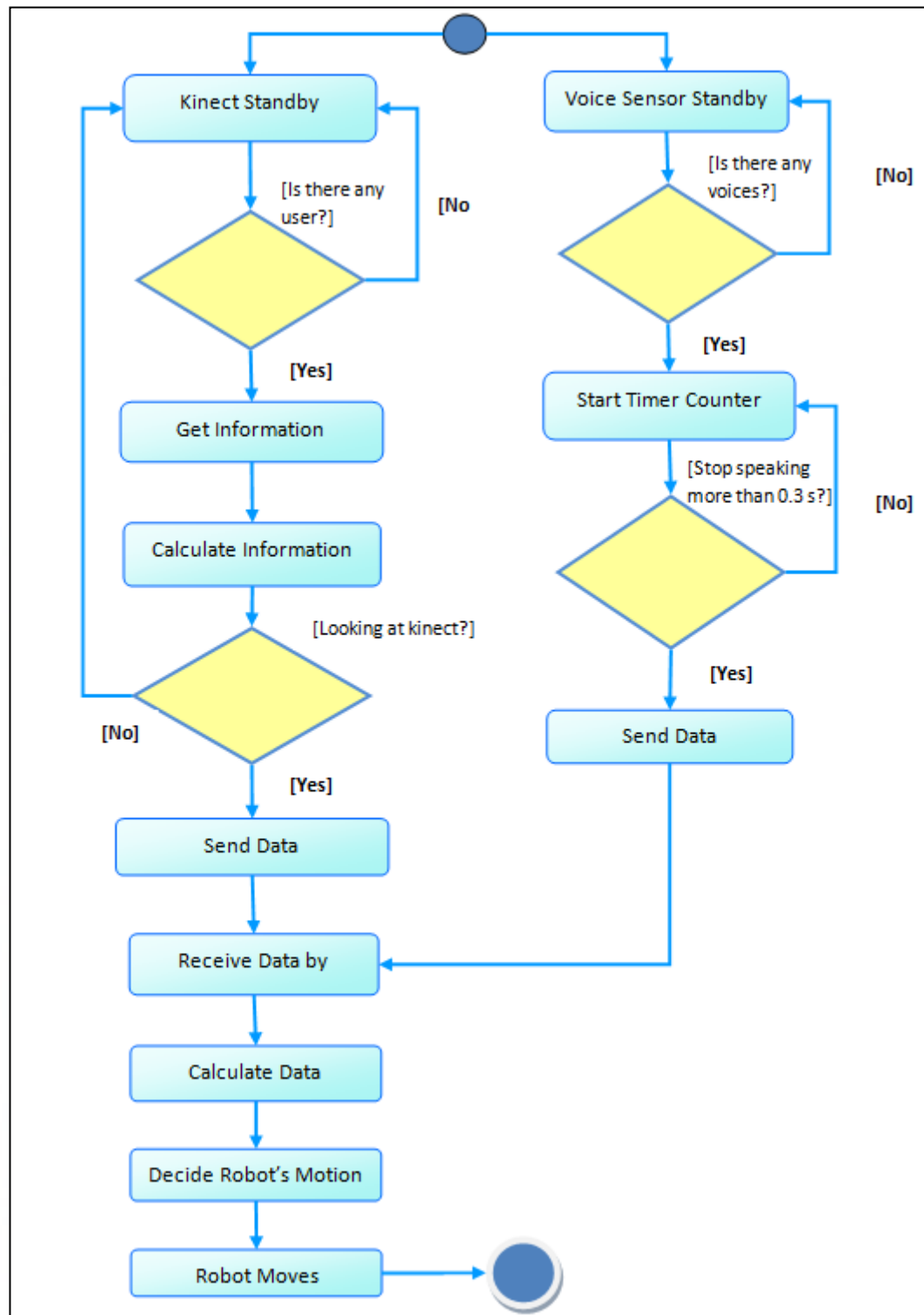
informasi posisi wajah dalam area kinect dan informasi rotasi kepala yang nantinya akan digunakan untuk memutuskan apakah user melihat kinect atau tidak. Beriringan dengan pendeteksian wajah sistem juga akan menunggu inputan suara dari user. Suara user akan terdeteksi melalui mikrofon. Ketika ada suara yang terdeteksi oleh program maka program akan menjalankan timer untuk menghitung jeda percakapan melalui mikrofon, ketika jeda melebihi 0.3 detik maka user diasumsikan telah berhenti berbicara.

Setelah informasi wajah atau suara user selesai diproses maka data tersebut akan dikirimkan ke program robot(*server*) yang telah terkoneksi sebelumnya, program robot nantinya akan selalu menunggu buffer data yang dikirimkan dari program kinect maupun program mikrofon. Setelah informasi didapatkan barulah nanti akan diproses dan ditentukan bagaimana robot harus bergerak. Setelah mejalani proses kalkulasi maka program akan memerintahkan robot untuk bergerak sesuai dengan hasil kalkulasi yang telah didapatkan tersebut.





## c. Activity Diagram



Gambar 3.3 Activity diagram

Berdasarkan *Activity diagram* yang telah dibuat penulis sendiri bertanggung jawab untuk mendeteksi wajah user melalui kinect, menetapkan posisi wajah dalam kamera kinect, mengimplementasikan program penghubung (*socket programming*) ke program robot (*server*), memproses informasi tersebut yang dikirimkan untuk kemudian ditentukan gerakan robot, serta sensor suara untuk mendeteksi suara user .

*Activity diagram* diatas menggambarkan kegiatan yang dilakukan selama sistem dijalankan. Berikut adalah penjabaran dari diagram diatas dan fungsi dari masing-masing activity yang tercantum.

- 1) *Kinect Standby*, pada tahap awal ini. Program kinect dan server telah terhubung dan telah dinyalakan. Kinect telah siap menerima inputan wajah dari setiap user yang berada pada jarak pandang kinect. Jika tidak ada user yang terdeteksi oleh kinect maka kinect akan tetap pada posisi *standby*, namun jika ada program akan mengambil informasi wajah user.
- 2) *Get information*, pada tahap ini wajah user telah terdeteksi oleh kinect. Selanjutnya adalah program kemudian mulai mengambil informasi wajah user seperti koordinat wajah dimana posisi user, dan mengambil titik koordinat hidung berdasarkan rotasi wajah saat itu.
- 3) *Calculate information*, pada tahap ini berdasarkan informasi yang telah diambil oleh kinect, informasi-informasi tersebut kemudian diproses untuk diambil kesimpulan apakah user sedang melihat kearah kinect atau tidak.

Berikut adalah algoritma yang digunakan untuk menentukan apakah user melihat kearah kinect:

a) *Face position*

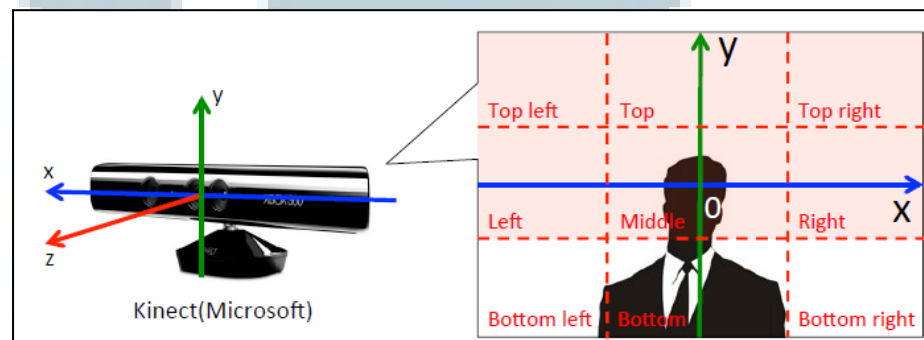
Setiap user yang masuk ke dalam jarak pandang kinect akan terdeteksi secara otomatis. Seperti yang ditunjukkan gambar dibawah



Gambar 3.4 Deteksi wajah oleh kinect

Ada beberapa tahapan untuk mendapatkan posisi wajah user:

1. Mengambil informasi jarak pandang kinect (maksimum dan minimum X Y).
2. Membagi area jarak pandang kinect kedalam 9 area yang telah ditetapkan. Seperti yang ditunjukkan gambar berikut.



Gambar 3.5 Pembagian posisi wajah

3. Membaca koordinat titik tengah wajah user yang sedang dideteksi.
4. Menggunakan *if*, untuk menentukan dimana wajah user berada diantara 9 area yang telah dibagi sebelumnya.

b) *Face rotation*

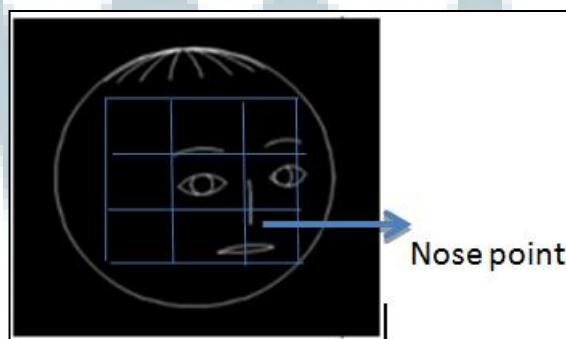
Sensor kinect akan membaca wajah user secara real-time. *Face rotation* ini berguna untuk menentukan kemana arah mata melihat. Tahap *face rotation* ini dikerjakan oleh rekan kerja penulis, Daniel Halim. Ada beberapa tahapan yang diterapkan oleh Daniel Halim untuk mendapatkan informasi arah tatapan mata:

1. *Face rotation* nantinya akan disimulasikan dalam bentuk gambar avatar pada sistem. Seperti pada gambar yang ditunjukkan dibawah



Gambar 3.6 Animasi avatar

2. Dari avatar tersebut kita implementasikan lagi cara yang mirip dengan algoritma *face position* yaitu pembagian area dengan menjadikan titik tengah hidung sebagai patokan, dan dapatkan posisi dimana titik tengah hidung berada.

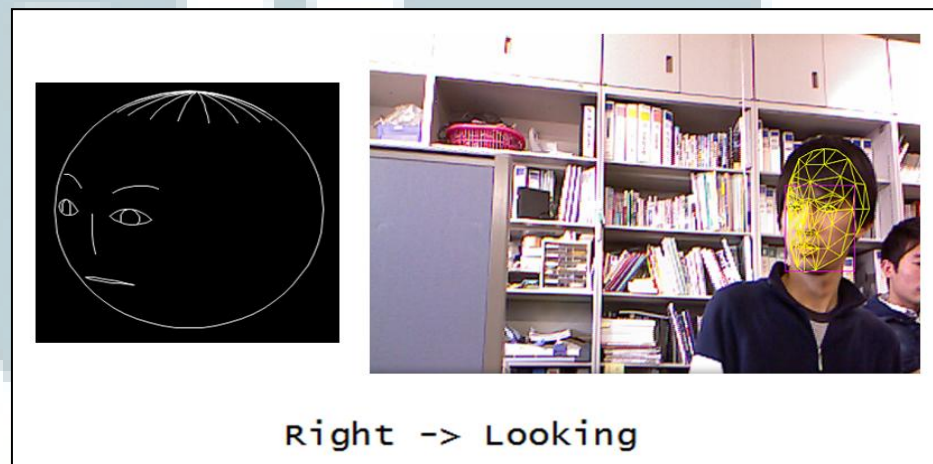


Gambar 3.7 Animasi Avatar dengan pembagian *nose point*

c) Kalkulasi informasi yang telah didapatkan

Setelah mendapatkan informasi dari proses *face position* dan *face rotation*. Kedua informasi tersebut digabung untuk menentukan apakah user melihat kearah kinect atau tidak. Berikut adalah contoh logika yang diterapkan:

1. Jika posisi wajah user berada di kanan kinect maka titik tengah hidung yang didapat dari proses *face rotation* harus berada di posisi kiri, jika terjadi seperti itu maka user tergolong sedang melihat ke arah kinect.



Gambar 3.8 *Right Looking User*

2. Jika posisi wajah user berada pada ditengah, maka posisi hidung juga harus berada di tengah. Pada gambar dibawah posisi wajah user berada pada tengah kinect namun posisi hidung berada di kiri atas, dapat disimpulkan bahwa user tidak melihat kinect.



Gambar 3.9 *Middle not looking User*

- 4) *Send data* (kinect), tahap ini dikerjakan oleh rekan kerja penulis, Daniel Halim. Informasi yang telah diproses sebelumnya kemudian dikirimkan ke program robot (*server*) melalui socket. Pada proyek ini penulis masih menggunakan socket yang terkoneksi dengan *localhost*, dengan IP: 127.0.0.1 dan port: 10003.
- 5) *Voice sensor standby*, pada tahap ini sistem telah siap untuk mendeteksi suara disekitar mikrofon. Jika tidak ada suara yang terdeteksi melalui mikrofon maka sensor ini akan tetap dalam keadaan siap untuk menerima suara, namun jika ada maka sensor akan langsung menyalakan conter.
- 6) *Start timer counter*, tahap ini dimulai ketika sudah ada suara yang terdeteksi oleh mikrofon, timer akan menyala jika terjadi jeda, jika terjadi jeda selama lebih dari 0.3 detik maka program akan mengirimkan data ke *server*, namun jika belum melewati 0.3 detik berarti user masih berbicara pada saat itu, maka akan kembali lagi ke tahapan *counter* dinyalakan.
- 7) *Send data* (mikrofon), tahap ini kesimpulan bahwa user sudah berhenti berbicara selama 0.3 detik kemudian dikirimkan ke program robot (*server*) melalui socket, sama seperti pengiriman oleh program kinect

- 8) *Received data by server*, pada tahap ini informasi yang telah dikirimkan oleh program *client* diambil. Data yang dikirimkan diterima oleh program *socket* yang telah diimplementasikan sebelumnya pada program *server*.
- 9) *Calculate data*, pada tahap ini data yang telah diterima sebelumnya kemudian diproses lagi untuk menentukan gerakan mana yang akan dilakukan oleh robot.
- 10) *Decide robot's motion*, pada tahap ini gerakan robot akan ditentukan. Pada tahap ini penulis hanya tinggal membuat *if* dan memanggil gerakan robot. Gerakan robot telah disediakan sebelumnya oleh teman sekelompok Saito.
- 11) *Robot moves*, pada tahap ini program robot telah memiliki kesimpulan gerakan mana yang perlu dilakukan oleh robot. Kemudian barulah program robot mengirimkan signal kepada robot. Terakhir robot bergerak.

Penulis bertanggung jawab terhadap program kinect, program mikrofon, serta program penghubung yang diterapkan pada program kinect dan program robot. Untuk program robot dikerjakan oleh programmer lain. Tahapan dan cara-cara yang digunakan penulis semasa pengerjaan proyek berpatokan dengan design yang telah dijelaskan diatas.

### 3.2.3 Development

Selama proses pengembangan dari sistem ini, ada beberapa perangkat keras dan perangkat lunak yang akan digunakan oleh penulis. Berikut ini adalah penjabaran perangkat keras yang digunakan oleh penulis.

1. Laptop ASUS K45DR sebagai perangkat yang digunakan selama pengembangan dengan spesifikasi umum sebagai berikut.
  - a. Processor AMD A8-4500 APU (4 cores, 4MB cache, 1,9GHz with Turbo Core up to 2,8GHz)
  - b. VGA AMD Radeon HD7600G + Radeon HD7470 1GB VRAM

- c. RAM 4GB DDR3
  - d. Harddisk dengan kapasitas 500GB
2. Robot dengan jenis Robovie-MR2 dengan CPU VS-RC003/HV
  3. Kinect yang digunakan sebagai sensor wajah
  4. Arduino dan Mikrofon sebagai perangkat keras yang digunakan untuk membaca sensor suara

Beberapa perangkat lunak yang digunakan oleh penulis selama tahap pengembangan sistem antara lain sebagai berikut.

1. Sistem operasi windows 7
2. Visual C++ 2010 express yang digunakan sebagai *environment* untuk membuat aplikasi
3. Roboviemaker 2 yang digunakan untuk startup robot dan membuat gerakan-gerakan robot
4. Software arduino yang digunakan untuk melakukan programming arduino
5. Google Chrome web browser yang digunakan untuk mencari informasi perihal pengembangan sistem
6. Microsoft word 2007 untuk membuat dokumentasi dan laporan kerja magang
7. Microsoft power point 2007 untuk presentasi proyek

#### 3.2.4 Integration and Test

Selama pengembangan sistem, ada beberapa pengujian yang dilakukan terhadap mekanisme kerja dari bagian-bagian sistem yang diuraikan sebagai berikut.

##### 1. *Testing* Kinect

Pengecekan dilakukan dengan cara mencoba langsung menjadi user, mencoba untuk berpindah ke berbagai posisi (masih dalam jarak pandang kinect) beriringan dengan gerakan kepala yang melihat ke arah kinect maupun gerakan kepala yang tidak melihat ke arah kinect. Hasil dari tersting tersebut ditampilkan ke dalam sebuah *console* yang



kemudian barulah dicocokkan dengan gerakan yang dilakukan secara real-time. Selama proses pembuatan program *face detection* ini, terjadi sangat banyak perbaikan dikarenakan penggunaan asumsi letak kordinat titik tengah wajah yang cukup rumit sehingga terkadang masih belum memberikan hasil yang akurat.

## 2. *Testing* Sensor Suara

*Testing* sensor suara cukup simple dan kurang ada perbaikan, dikarenakan prosesnya yang sederhana. Program hanya perlu mendeteksi adanya suara dan menyalakan counter perhitungan 0.3 detik setelah user berhenti berbicara. Namun sedikit masalah yang terjadi adalah kualitas mikrofon yang kurang bagus sehingga agar suara user dapat terdeteksi mulut user harus berada pada jarak kurang dari 1cm dari mikrofon dengan suara yang cukup kuat.

## 3. *Testing* program penghubung (integrasi sistem)

Salah satu bagian sistem terpenting yang dibuat adalah program yang menghubungkan kerja perangkat-perangkat. Program penghubung yang menggunakan socket sebagai media komunikasi ini dilakukan *testing* yang berupa mengecek pengiriman dari program kinect menuju kepada program robot. Data yang sebelum dikirimkan haruslah dipastikan sama dengan hasil ketika program sudah diterima. Pengecekan dilakukan dengan cara menceteak buffer yang diterima oleh program robot. User secara langsung mencoba program kinect yang dijalankan dan melihat langsung hasil dari pendeteksian kinect pada console program robot.

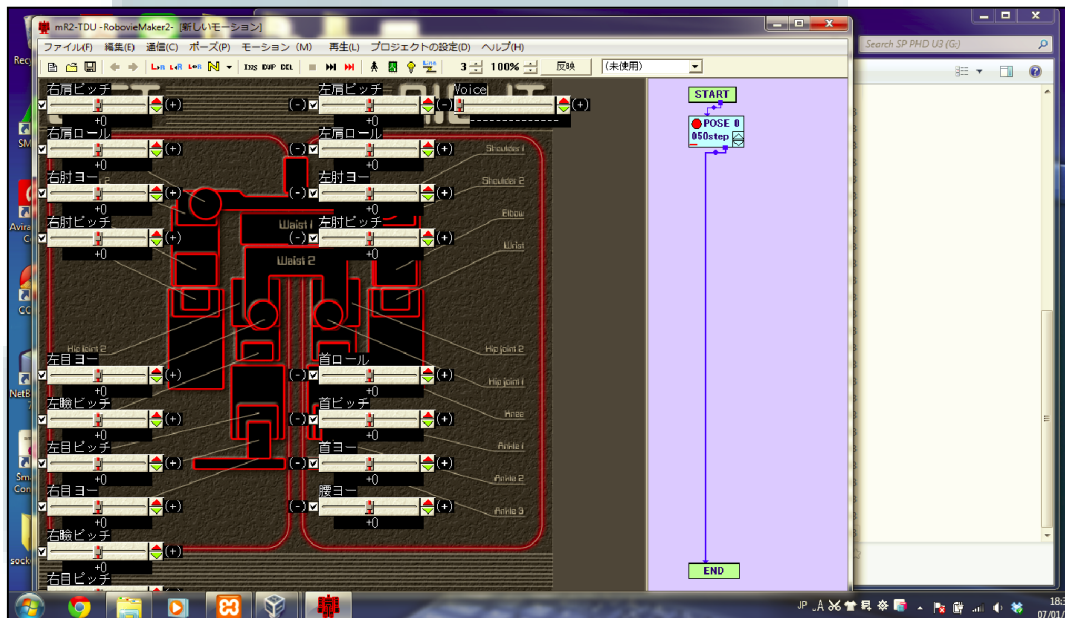
*Testing* dilakukan oleh beberapa pihak. Untuk *testing* yang dilakukan oleh develop *testing* dilakukan beringingan dengan proses development. *Testing* juga dilakukan oleh ketua project dan dilakukan uji coba pada beberapa user yang dimati tolong untuk memberikan saran. Dari hasil *testing* yang dilakukan oleh beberapa user sempat terjadi beberapa kali perbaikan terkait dengan cara kerja dan keakuratan.

### 3.2.5 Implementation

Pada tahap implementasi ini, penulis menggabungkan berbagai sub-sub program yang telah dibuat sebelumnya. Seperti yang telah dijelaskan pada tahap design nantinya program kinect akan dihubungkan dengan program robot. Berikut adalah gambar robot beserta dengan kinect yang digunakan oleh kelompok penulis untuk mengembangkan sistem ini.

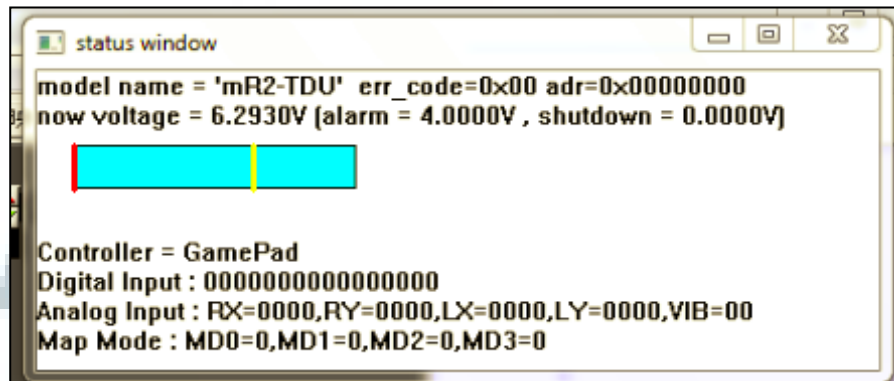
Posisi dimana kinect berada sangat bergantung pada tepatnya gerakan robot. Pada gambar tersebut dapat dilihat bahwa kinect diletakan tepat di depan robot dengan tujuan agar kinect dapat berfungsi sebagai mata robot.

Pada awal ketika robot akan dijalankan robot harus diaktifkan terlebih dahulu melalui software khusus yang bernama Roboviemaker2. Software ini berfungsi untuk mengendailkan robot secara manual. Pada *software* ini terdapat pengendali robot seperti menggerakkan bagian-bagian tubuh robot serta mengaktifkan ataupun mematikan robot. Adapun tampilan *software* pengendali robot seperti berikut.



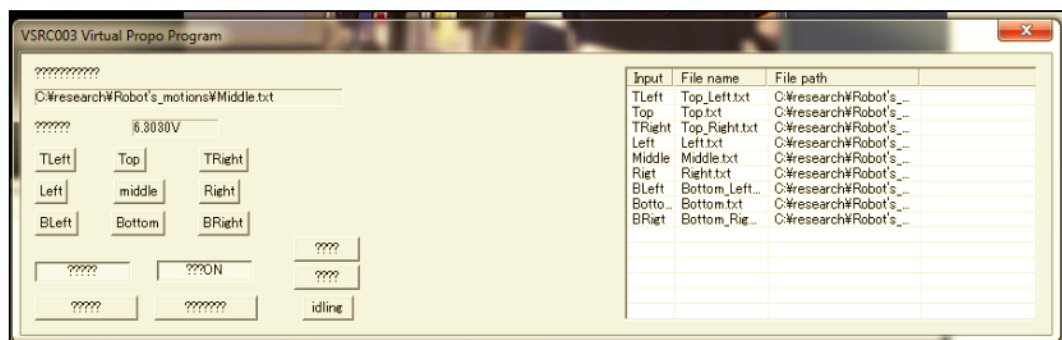
Gambar 3.10 Program Roboviemaker2

Berikut adalah *windows* yang keluar ketika robot dinyalakan, pada *windows* yang keluar ini terdapat voltase robot pada saat itu yang tidak tetap.



Gambar 3.11 Tampilkan voltase robot diaktifkan

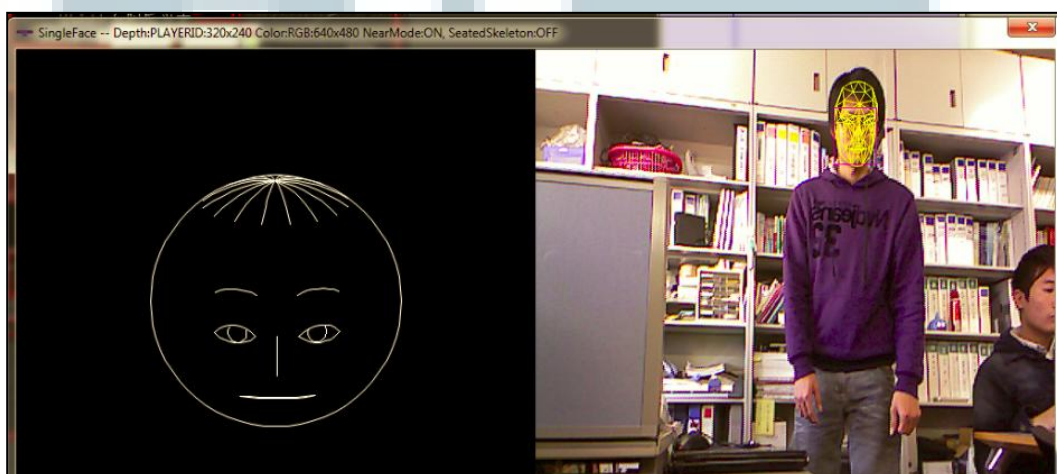
Setelah robot siap untuk digunakan maka program robot terlebih dahulu dijalankan karena program robot bertindak sebagai *server* yang nantinya akan menerima informasi yang dikirimkan oleh sub-sub program. Ketika program robot telah dijalankan maka seorang *programmer* harus melakukan inisialisasi dengan memasukan list-list gerakan robot yang telah dibuat sebelumnya menggunakan software Roboviemaker2. Program robot nantinya akan berjalan dengan otomatis setelah gerakan robot dinisialisasi. Berikut adalah contoh tampilan program robot setelah dijalankan.



Gambar 3.12 User interface program robot

Seperti yang telah dipaparkan pada gambar diatas, terlihat disebelah kanan terdapat list-list gerakan robot yang telah dinisialisasi sebelumnya sehingga nantinya program tinggal memilih gerakan mana yang akan dipanggil.

Setelah program robot telah siap untuk disambungkan dengan sub program, maka kemudian program kinect dijalankan. Pada saat dijalankan program socket yang berada pada program kinect akan terhubung dengan *server* secara otomatis setelah itu program kinect akan menampilkan view kamera beserta *avatar* yang mensimulasikan rotasi wajah user yang sedang terdeteksi. Berikut gambar ketika program kinect dijalankan.

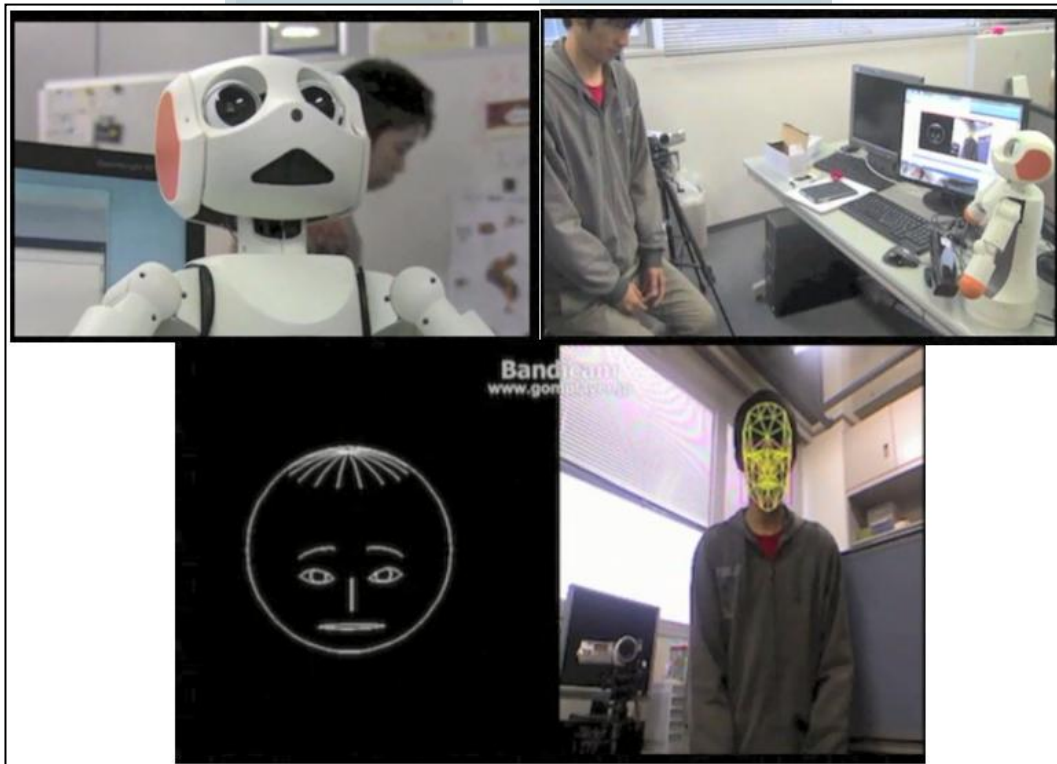


Gambar 3.13 User interface program kinect

User yang wajahnya terlihat oleh kinect akan dideteksi oleh kinect, kemudian program kinect akan menampilkan gerakan rotasi wajah user pada sebuah *avatar* yang dapat terlihat pada sebelah kiri gambar diatas. Ketika user menghadap keatas maka avatar akan menghadap keatas, namun ketika user menghadap ke kiri avatar akan menghadap ke kiri begitu seterusnya.

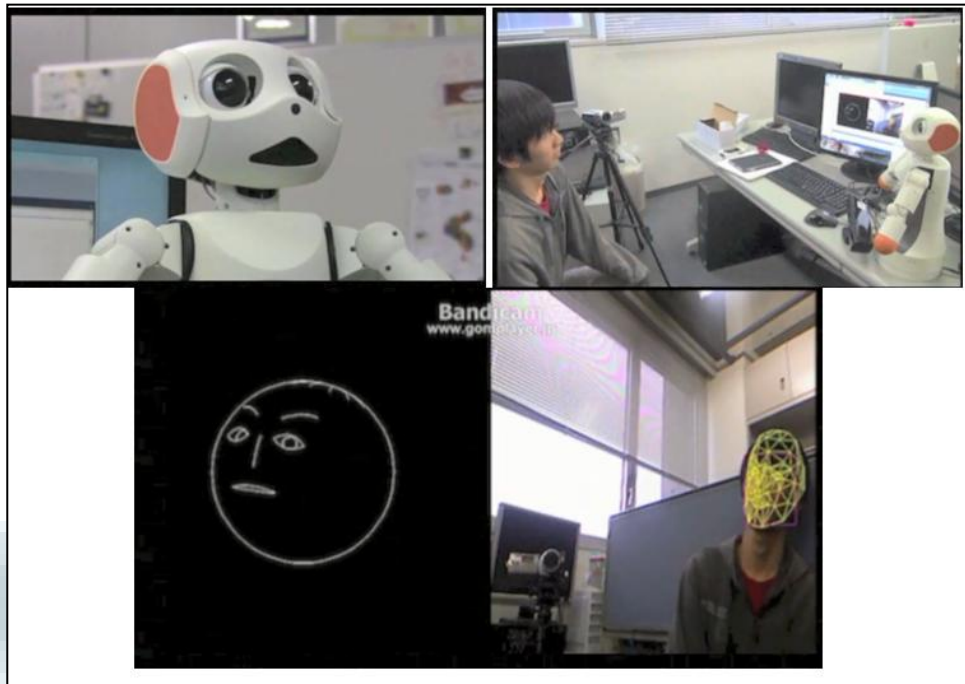
Ketika wajah user terdeteksi oleh kinect nantinya program akan menentukan dimana wajah user berada dan menentukan apakah wajah user menghadap ke arah kinect atau tidak. Kemudian akan program kinect nantinya akan mengirimkan informasi yang telah diproses tersebut kearah kinect melalui

socket yang telah terhubung sebelumnya dengan program robot. Program robot yang telah terhubung, nantinya akan menerima paket data yang telah dikirimkan. Setelah itu program robot akan mengadakan kalkulasi untuk menentukan gerakan robot mana yang akan dijalankan nantinya. Berikut disertakan gambar-gambar yang menunjukkan proses gerakan robot berdasarkan pandangan wajah user.

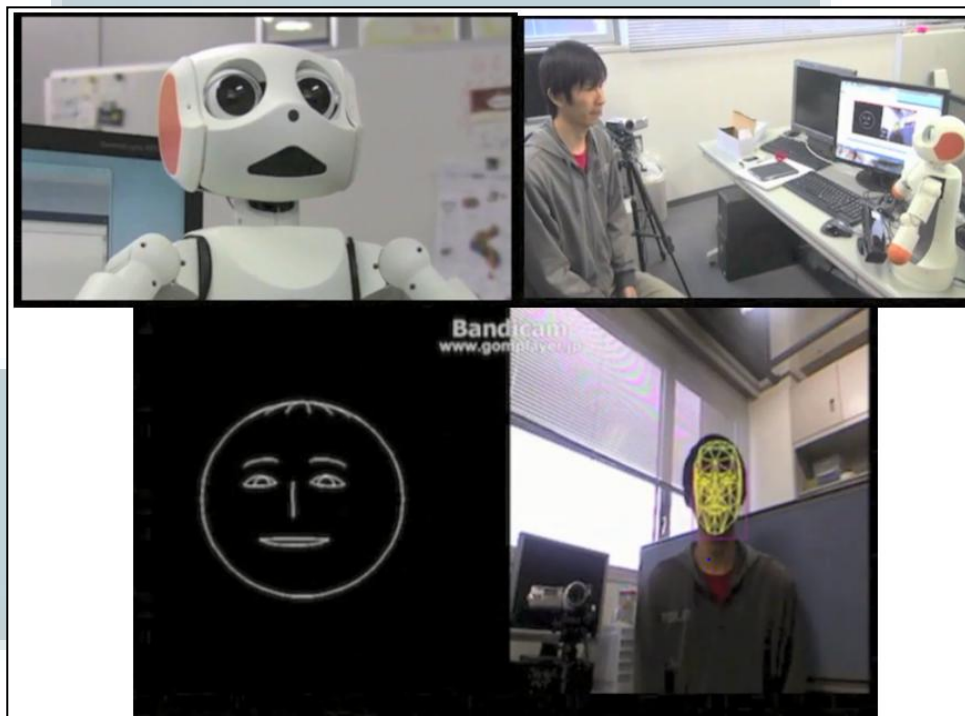


Gambar 3.14 Sistem robot 1

UMMN



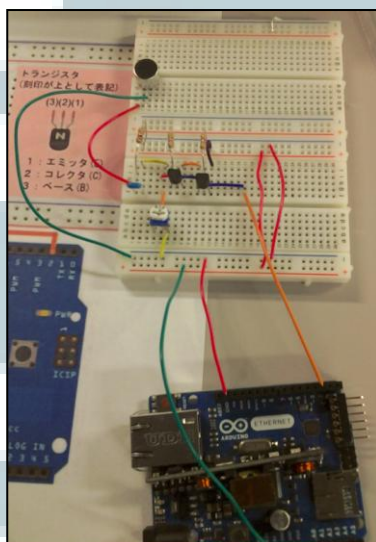
Gambar 3.15 Sistem robot 2



Gambar 3.16 Sistem robot 3

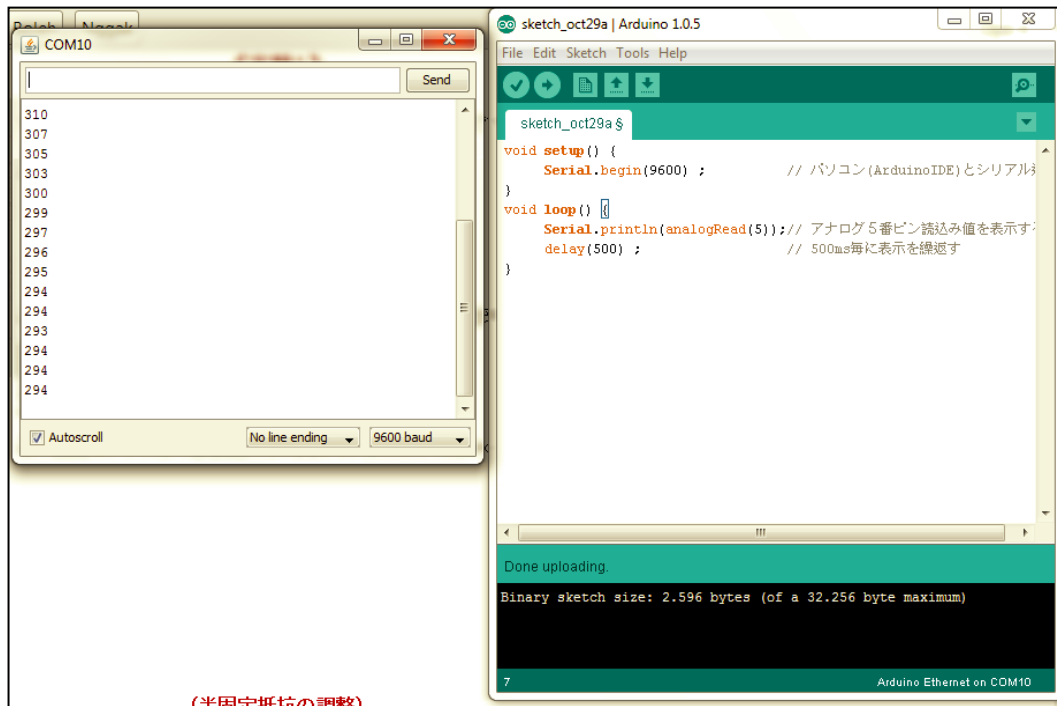
Pada project ini juga penulis membuat sensor suara yang rencananya dihubungkan dengan program robot, namun karena kualitas mikrofon yang kurang maka diputuskan nantinya sensor suara akan diganti dengan mikrofon lain dan tidak menggunakan arduino lagi.

Untuk sistem sensor suara penulis menyiapkan sirkuit yang dirangkai untuk mendeteksi suara. Mikrofon dihubungkan ke sirkuit dan kemudian sirkuit tersebut dihubungkan dengan arduino. Berikut adalah sirkuit yang telah dibuat oleh penulis



Gambar 3.17 Rangkaian sensor suara

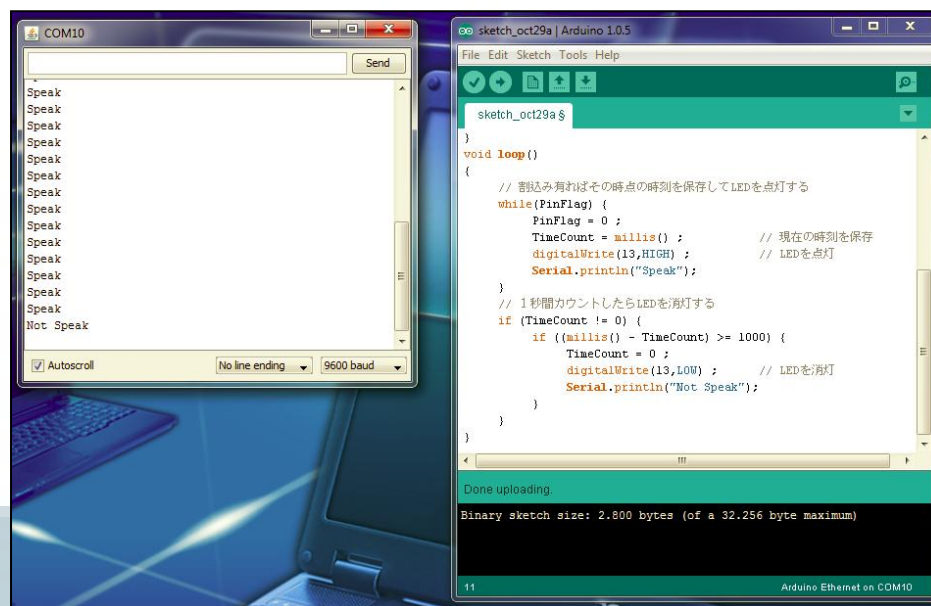
Setelah sirkuit dirangkai, maka dihubungkan dengan computer kemudian dilakukan pengecekan apakah sirkuit telah berfungsi dengan benar atau tidak. Pada gambar diatas, program dibuat untuk membaca signal analog yang dikirimkan oleh mikrofon melalui sirkuit dan arduino. Signal analog tersebut menunjukkan tingkatan suara yang ditangkap disekitar mikrofon.



Gambar 3.18 Program sensor suara

Setelah sirkuit berfungsi dengan baik maka penulis membuat program dengan bahasa pemrograman arduino untuk mendeteksi user yang sedang berbicara ke mikrofon, apabila user berbicara maka akan mencetak “*speak*” namun apabila user berhenti berbicara maka akan mencetak “*not speak*”. Dengan adanya program ini maka dapat dihitung waktu user telah berhenti berbicara dengan jeda 0,3 detik.





Gambar 3.19 Program pendeteksi suara

### 3.2.6 Operation and Maintenance

Tahap *operation* untuk saat ini diserahkan kepada ketua proyek karena penulis hanya bertugas untuk pembuatan program pendeteksi wajah, program sensor suara, serta program penghubung antara kinect dan program robot.

Tahap *maintenance* sudah dua kali dilakukan terutama pada program kinect, *maintenance* pertama dilakukan karena hasil pendeteksi wajah user tidak sesuai harapan sehingga dilakukan perbaikan dengan menerapkan konsep lain yaitu konsep yang dijelaskan pada laporan ini sementara *maintenance* kedua dilakukan perbaikan dengan membatasi jarak pandang kinect hanya sebatas luas windows yang tampil pada layar saja, sebelum dilakukan *maintenance* kedua jarak pandang kinect lebih luas dari pada luas windows yang ditampilkan sehingga tampilan kinect bergeser-geser dan susah untuk digunakan oleh user.

### 3.2.7 Penulisan Dokumentasi dan Laporan Magang

Penulisan dokumentasi dan laporan magang dilakukan selama proses pengembangan sistem berlangsung. Penulis membuat catatan apa saja yang dilakukan dan pencapaian setiap minggu. Selama proses kerja magang berlangsung, laporan magang juga turut dibuat untuk diserahkan ke pihak Universitas Tokyo Denki bagian laboratorium interaksi. Laporan yang telah dibuat dan diserahkan ke pihak Universitas Tokyo Denki dalam format bahasa inggris yang berisikan latar belakang, metode yang digunakan, flow program, penjelasan sistem, serta penjelasan algoritma yang digunakan pada program. Pembuatan laporan dimaksudkan agar nantinya sistem yang telah dibuat dapat digunakan maupun dikembangkan oleh anggota laboratorium. Penulis juga menjelaskan cara kerja sistem dan membuat dokumentasi program kepada ketua proyek.

Sebagai gambaran yang lebih jelas, berikut ini penulis menampilkan detail realisasi kerja magang yang dilaksanakan oleh penulis selama bekerja sebagai *programmer* di Universitas Tokyo Denki dalam bentuk tabel.

Tabel 3.2 Realisasi kerja magang

Minggu	Kegiatan
1	<ul style="list-style-type: none"> <li>- Pengenalan tempat kerja dan anggota</li> <li>- Pembahasan <i>project</i></li> </ul>
2	<ul style="list-style-type: none"> <li>- Penginstallan <i>software</i> robot</li> <li>- Penginstallan visual c++ 2010 express</li> <li>- Pengenalan penggunaan <i>software</i> robot (Roboviemaker 2)</li> </ul>
3	<ul style="list-style-type: none"> <li>- Pengenalan dan pembelajaran koding <i>face detection</i> menggunakan kinect</li> </ul>
4	<ul style="list-style-type: none"> <li>- Mendapatkan koordinat wajah</li> <li>- Mendapatkan koordinat hidung user</li> </ul>
5	<ul style="list-style-type: none"> <li>- Menentukan posisi wajah dalam area pandang kinect</li> </ul>
6	<ul style="list-style-type: none"> <li>- Menentukan apakah user melihat ke arah kinect atau tidak melalui menggabungkan informasi posisi wajah dalam kinect</li> </ul>

Minggu	Kegiatan
7	<ul style="list-style-type: none"> <li>- Pengenalan dan pembelajaran <i>programming</i> arduino</li> <li>- Membuat rangkaian dan kodingan sensor tekanan</li> <li>- Membuat rangkaian dan kodingan sensor suhu</li> </ul>
8	<ul style="list-style-type: none"> <li>- Membuat rangkaian dan kodingan sensor suara melalui mikrofon</li> </ul>
9	<ul style="list-style-type: none"> <li>- Mempersiapkan presentasi untuk melaporkan perkembangan proyek</li> </ul>
10	<ul style="list-style-type: none"> <li>- Perbaiki program kinect, mengganti konsep <i>face rotation</i> yang telah diterapkan sebelumnya</li> </ul>
11	<ul style="list-style-type: none"> <li>- <i>Socket programming</i> membuat program <i>client server</i> untuk mengirimkan data yang telah dihasilkan dari program <i>client</i></li> </ul>
12	<ul style="list-style-type: none"> <li>- Mengimplementasi program socket yang telah dibuat ke program server</li> </ul>
13	<ul style="list-style-type: none"> <li>- Perbaiki kepada program kinect (membatasi jarak pandang kinect menjadi ukuran <i>fix</i> dengan <i>windows</i>)</li> </ul>
14	<ul style="list-style-type: none"> <li>- <i>Final presentation</i></li> </ul>

### 3.3 Kendala dan Solusi

#### 3.3.1 Kendala yang Ditemukan

Berikut kendala-kendala yang ditemukan selama proses kerja magang berlangsung:

- a. Kesulitan dalam memahami permintaan maupun ekspektasi tugas yang harus dicapai, serta interaksi antar anggota laboratorium karena hambatan bahasa.
- b. Kurang pengetahuan terkait pemrograman kinect serta sensor-sensor sehingga mengalami kesulitan dalam mengembangkan sistem.
- c. Kualitas mikrofon yang kurang memadai sehingga sensor yang telah diselesaikan tidak bekerja secara maksimal
- d. Tidak adanya *job description* yang jelas antara sesama mahasiswa magang terkait tugas dan tanggung jawab dikarenakan kurangnya persiapan penerimaan program kerja magang.

### 3.3.2 Solusi Atas Kendala

Solusi yang sekiranya dapat mengatasi kendala diatas:

- a. Menggunakan teknologi *translator* serta mendalami pembelajaran bahasa Jepang untuk memudahkan komunikasi sehingga pesan dapat diterima dengan lebih baik.
- b. Mendalami pengetahuan terkait pemrograman kinect dan sensor-sensor melalui modul-modul maupun mencari informasi tambahan dari sumber lainnya.
- c. Dilakukan diskusi terlebih dahulu terkait kebutuhan hardware yang ingin digunakan sehingga tidak terjadi kegagalan teknis pada produk yang dikerjakan.
- d. Membuat timeline ataupun briefing terkait tanggung jawab dan ekspektasi pekerjaan yang harus dicapai masing-masing mahasiswa magang.

UMMN