

# RANCANG BANGUN ROBOT DENGAN *ELECTRONIC NOSE* UNTUK MENIDENTIFIKASI GAS MENGGUNAKAN MULTI LAYER PERCEPTRON

I Dewa Gede Rai M<sup>1)</sup> Oyas Wahyunggoro<sup>2)</sup> Indah Soesanti<sup>3)</sup>

- 1) Program Studi S1Sistem Komputer, STIKOM Surabaya, email: [dewa@stikom.edu](mailto:dewa@stikom.edu)  
2) Teknik Elektro dan Teknologi Informasi, Univ. Gadjah Mada, email : [oto367@jeti.gadjahmada.edu](mailto:oto367@jeti.gadjahmada.edu)  
3) Teknik Elektro dan Teknologi Informasi, Univ. Gadjah Mada, email: [indah@mti.ugm.ac.id](mailto:indah@mti.ugm.ac.id)

**Abstract:** Research in the field of robotics development equipped with electronic nose is a new challenge. By implementing electronic nose, the robot can identify and mitigate the risks arising from the interaction between humans and harmful gases. This research used robotino that was programmed and equipped with electronic nose to make it more easy and flexible to be controlled by humans to approach the object (toxic gas), as well as designing an artificial intelligence (Multi Layer Perceptron) with Error Back Propagation method to gain the weight. The weight that was obtained will be applied to a robot, so by using feed forward process according to these weigh, the robot was able to precisely and quickly identify gasses that were obtained from the output of TGS array sensors. Experiment result shows that by training 12 sample data, the robot is able to classify gasoline and alcohol with the accuracy of 100%.

**Keywords:** Robot, Electronic Nose, Multi Layer Perceptron, Back Propagation.

Pada Desember 2004 stasiun luar angkasa NASA mengalami kebocoran gas ammonia yang dapat mengancam keselamatan astronot, sehingga para ilmuwan NASA mulai memikirkan bagaimana mengembangkan sensor untuk mendeteksi gas beracun. Hal ini tidak menutup kemungkinan untuk terjadi dalam dunia industri, yang pada proses pengolahan mungkin melibatkan gas-gas yang berbahaya bagi manusia. Robot yang dilengkapi dengan *electronic Nose* dapat menjadi solusi untuk mengetahui potensi gas berbahaya tanpa melibatkan interaksi manusia secara langsung.

Penelitian dalam bidang penerapan jaringan syaraf tiruan dan *E-Nose* saat ini sudah banyak dilakukan, namun untuk penelitian dengan topik penerapan *E-Nose* yang diimplementasikan pada robot dapat dibilang merupakan tantangan baru dalam topik penelitian. Menurut penelitian dengan judul Aplikasi Jaringan Syaraf Tiruan Untuk Identifikasi Aroma Teh Menggunakan *Electronic Nose* (Nugroho Dkk, 2008) menunjukkan bahwa jaringan syaraf tiruan mampu memberikan klasifikasi benar 100% dari 9 set data pengujian. Hasil pelatihan dan pengujian menunjukan bahwa jaringan syaraf tiruan yang dibentuk dapat mengidentifikasi dan

membedakan jenis teh hitam, teh hijau dan teh wangi melati dengan hasil yang baik. Penelitian tersebut dilakukan dengan tahapan pengukuran parameter-parameter yang menentukan aroma teh melalui parameter uji laboratorium menggunakan perangkat *E-Nose*. Peralatan *E-Nose* yang dimaksud didalamnya terdiri atas 4 buah sensor dengan tipe: TGS 880, TGS 826, TGS 822 dan TGS 825. Jaringan syaraf tiruan dibentuk dengan jumlah 4 lapisan, terdiri atas 1 lapisan input dengan 4 sel syaraf masukan, 2 lapisan tersebunyi dengan 8 sel syaraf dan 7 sel syaraf, serta 1 lapisan luaran dengan fungsi aktivasi sigmoid bipolar, algoritma pelatihan menggunakan *back propagation* dengan data pelatihan sebanyak 63 set data.

Penelitian yang dilakukan oleh Nugroho, dkk memiliki keterbatasan antara lain :

- 1) keseluruhan sistem masih sangat tergantung pada manusia terutama pada saat melakukan pengujian terhadap sampel teh yang digunakan, dan
- 2) dinilai dalam bidang keamanan hal ini merupakan kekurangan jika untuk melakukan pengujian masih membutuh interaksi manusia secara langsung terutama jika gas yang akan

diuji merupakan gas yang berbahaya bagi manusia, sehingga akan lebih baik jika memanfaatkan robot untuk melakukan pengujian terhadap sampel tersebut.

Penelitian dengan judul *Simultaneous Estimation of Odor classes and Concentrations Using an Electronic Nose with Function Approximation Ensemble* (Daqi, Wei, 2006) menunjukkan bahwa dengan menerapkan beberapa model fungsi memberikan kemampuan untuk memperkirakan konsentrasi dan klasifikasi gas tertentu. Penelitian tersebut dilakukan dengan menguraikan proses *Multiple Input/Multiple Output (MIMO)* kedalam *Multiple many to one task* secara berurutan. *Many to One task* yang dimaksud disini adalah fungsi perkiraan menggunakan *multivariate logaritmic regression, quadratic multivariate logarithmic regression, multi layer perceptron* dan *support vector machine*. Keempat fungsi tersebut dikombinasikan untuk menentukan klasifikasi dan konsentrasi dari gas tertentu. Penelitian ini juga belum diimplementasikan pada robot sehingga masih ada kemungkinan resiko interaksi manusia dengan gas beracun.

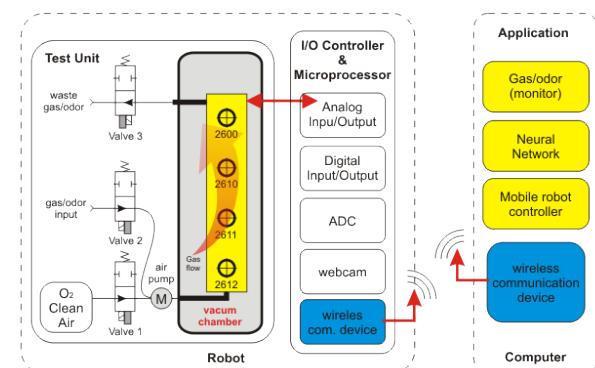
Penelitian lain dengan judul *Building Gas Concentration Gridmaps with a mobile robot* (Lilienthal, Duckett, 2004) menunjukkan bahwa dengan menerapkan *array sensor gas* pada robot yang dilengkapi dengan tanda tertentu pada bagian atas robot (*marker*) mampu merepresentasikan pola distribusi gas dalam ruangan. Metode *Gaussian weighting model* mampu memberikan citra pola distribusi gas dalam ruangan dengan melakukan perhitungan terhadap data luaran *array sensor* yang direkam, digabungkan dengan posisi yang didapat dari 4 buah kamera yang merekam pergerakan robot yang ditempatkan pada keempat sudut dari ruangan. Dalam penelitian ini robot belum memiliki kemampuan untuk mendeteksi secara langsung kebocoran gas dalam ruangan, dan masih memiliki

hambatan jika dalam pergerakan robot dengan pola tertentu terhalang oleh objek lain.

Berdasarkan pada beberapa penelitian tersebut, pada penelitian ini dicoba merancang suatu sistem untuk mengidentifikasi gas berbahaya yang aman terhadap manusia. Hal ini dilakukan dengan merancang sebuah robot yang dilengkapi dengan *electronic nose* yang mudah dan lebih fleksibel dikendalikan oleh manusia untuk mendekati objek (gas beracun), dan merancang kecerdasan buatan (*Multi Layer Perceptron*) dengan menggunakan metode *Back Propagation* yang langsung diaplikasikan ke robot, sehingga robot mampu secara tepat dan cepat dalam melakukan identifikasi objek yang didapatkan dari luaran sensor array TGS.

## METODE

Penelitian ini dilakukan dengan membagi pada dua bagian besar yaitu perancangan *mobile robot* dan perancangan pada komputer sebagai *monitor and operating machine*. Dua bagian ini nantinya akan berkomunikasi menggunakan media *wireless network* untuk mengidentifikasi gas tertentu. Gambar 1 menunjukkan diagram kotak penelitian secara umum.



Gambar 1. Diagram Kotak Penelitian

*Mobile robot* dilengkapi dengan *test unit*, kamera dan *wireless card*. *Test unit* dirancang memiliki kemampuan untuk menghisap bau (*odor*) sebagai objek yang akan diamati,

kemudian gas tersebut dialirkan melalui selang menuju *vacuum chamber* yang didalamnya telah dilengkapi dengan *array sensor* untuk mendeteksi gas. *Array sensor* yang dimaksud adalah rangakain yang terdiri dari beberapa sensor TGS yang memiliki spesifikasi tersendiri yang nantinya menentukan kandungan dalam gas yang akan diuji. O<sub>2</sub> digunakan sebagai gas pembersih dalam *vacuum chamber* setelah melakukan pengujian. Kamera pada robot nantinya akan digunakan sebagai indera penglihatan bagi operator agar tidak berinteraksi secara langsung dengan gas yang akan diidentifikasi.

Komputer digunakan sebagai *monitor and operating machine* yang memungkinkan manusia untuk berinteraksi dengan *mobile robot*. Komputer ini dilengkapi dengan program untuk mengontrol gerakan robot, memvisualisasikan gerakan yang ditangkap oleh kamera yang ada pada *mobile robot*, serta melakukan analisis dengan menggunakan *Artificial Neural Network (Multi Layer Perceptron)* untuk mengidentifikasi gas yang diuji. Sebagai komunikasi maka komputer akan terhubung dengan *mobile robot* menggunakan *wireless card*.

## Perancangan Perangkat Keras

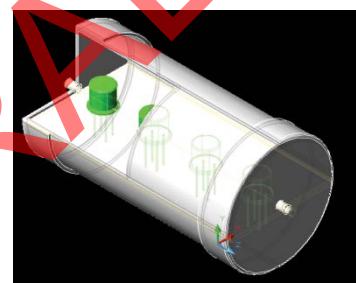
### Pembuatan Prototype E-Nose Pada Robot

Pembuatan *e-nose* dimulai dengan merancang *vacuum chamber*. *Vacuum chamber* ini bekerja seolah-olah sebagai paru-paru yang menyimpan udara yang dihisap oleh *mini air pump* dalam waktu tertentu, untuk membuka dan menutup digunakan *vacuum valve* yang diatur dalam periode tertentu. Waktu untuk membuka dan menutup ditentukan melalui beberapa percobaan.

Gambar 2 merupakan rancangan purwa rupa untuk *vacuum chamber*, seperti yang telah diuraikan sebelumnya bahwa *vacuum chamber* ini terbuat dari pipa PVC yang saling tertutup di kedua ujungnya, dimana didalamnya terdapat *array sensor*. Kedua penutup pipa ini diberi fitting agar selang dapat masuk dengan sempurna tanpa menimbulkan

kebocoran, selang ini terhubung dengan *electronic valve* sebagai saklar on/off yang mengatur aliran udara yang masuk atau meninggalkan *vacuum chamber*.

*Mini air pump* diletakan sebelum *odor intake* agar mampu memberikan volume gas yang cukup pada *vaccum chamber* untuk diidentifikasi, hal ini dikarena udara yang dihisap oleh *mini air pump* didorong untuk masuk kedalam *vacuum chamber* selama waktu tertentu dan *valve* pada *waste odor* dalam keadaan tertutup. Sehingga volume yang cukup yang menempati *vacuum chamber* diharapkan mampu mewakili kondisi gas yang sesungguhnya sesuai dengan kodisi gas yang berada diluar sistem.



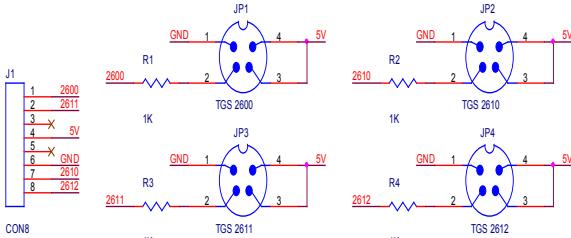
Gambar 2. Rancangan Purwarupa *Vacuum Chamber*

### Perancangan Array Sensor TGS

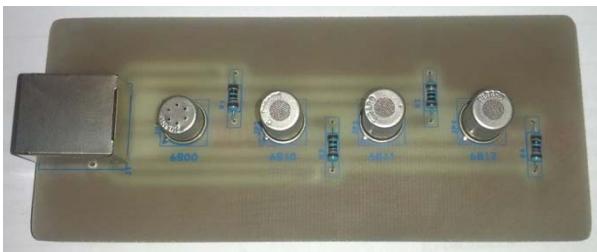
Dalam penelitian ini digunakan 4 buah sensor TGS dari *Figaro* yaitu TGS 2600, 2610, 2611, dan 2612 yang memiliki karakteristik hampir mirip antara yang satu dengan lainnya. Sesuai dengan *datasheet*, karakteristik gas yang dapat terdeteksi yaitu gas yang mengandung *ethanol*, *methane*, *isobutane* dan *propane*.

Agar dapat bekerja dengan baik sensor ini membutuhkan dua tegangan masukan. *Heater Voltage (V<sub>H</sub>)* digunakan sebagai tegangan *heater* dan *Circuit Voltage (V<sub>c</sub>)* merupakan tegangan *supply* rangkaian, keduanya diberikan catu daya sebesar 5 volt DC. Nilai resistor beban ( $R_L$ ) dapat dipilih atau *di-adjust* untuk mengoptimasikan nilai *alarm threshold*, menjaga *power dissipation* ( $P_s$ ) semikonduktor di bawah batas 15mW. *Power dissipation* ( $P_s$ ) akan menjadi sangat tinggi ketika

nilai  $R_S$  adalah sama dengan nilai  $R_L$ . Untuk mengatur  $V_H$  dan pengambilan data analog yang dihasilkan  $V_{RL}$  digunakan rangkaian yang terdiri dari relay, transistor, resistor, serta dioda sehingga dirancang pula modul sensor array TGS seperti Gambar dan gambar 4..



Gambar 3. Skematik Array Sensor



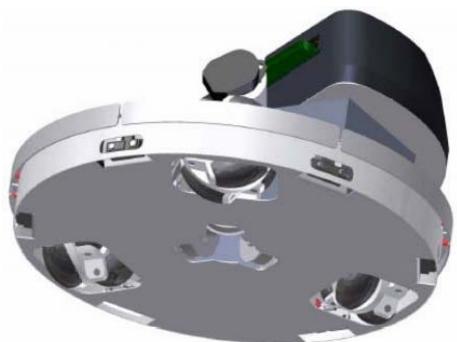
Gambar 4. Array Sensor TGS

## Perancangan Perangkat Lunak

### Perancangan Kendali Robot

Robotino memiliki sistem pergerakan *omni-directional drive* dimana terdapat 3 buah roda yang digunakan untuk menggerakkan Robotino. Berikut gambar sistem *omni-directional drive* Robotino pada Gambar .

Untuk untuk menggerakkan Robotino digunakan fungsi `setVelocity` yang memiliki parameter vx, vy, dan omega. Parameter vx adalah parameter kecepatan pada sumbu x dan vy adalah parameter kecepatan untuk sumbu y, dengan ketentuan parameter vx dan vy dalam satuan mm/s. Omega merupakan parameter kecepatan sudut dengan ketentuan parameter omega dalam satuan deg/s.



Gambar 5. Omni-Directional Drive Pada Robotino

Untuk mengakses fungsi tersebut dibutuhkan deklarasi *header* file yang terletak pada `rec::Robotino::com::OmniDrive.h`, dimana pada awal program harus dideklarasikan terlebih dahulu. Berikut ini adalah *script* untuk menggerakkan robot, dimana *value* harus diisi nilai integer yang mewakili kecepatan motor.

```
OmniDrive omni;
while(( com.isConnected() && false ==
bumper.value() ) && timer.msecsElapsed() <
100000 )
{
    omni.setVelocity(value,value,value);
}
```

### Perancangan Streaming Gambar

Untuk menampilkan data gambar yang sudah tersimpan pada Iplimage kedalam window baru digunakan prototipe fungsi pada *library* OpenCV yaitu `cvShowImage(const char *name, const CvArr *image)`. Dengan ketentuan parameter `const char *name` adalah nama *window* dan `const CvArr *image` adalah Iplimage yang ditampilkan. Berikut program yang digunakan untuk menampilkan data gambar secara *streaming*:

```
cam.setStreaming (true);
cvShowImage( "image" , img1);
c = cvWaitKey(50);
```

Untuk *refresh* gambar yang ditampilkan pada *window* dibutuhkan fungsi *cvWaitKey*. Ini dikarenakan OS memiliki waktu minimum dalam menjalankan *threads* secara bergantian. Fungsi ini tidak memberikan *delay* persis seperti parameter yang telah set, namun *delay* tergantung *threads* yang sedang berjalan pada komputer saat itu. Nilai yang dikeluarkan dari fungsi ini adalah kode untuk penekanan tombol atau -1, apabila tidak ada tombol yang ditekan selama waktu yang ditentukan.

### Perancangan Sampling Gas

Pada sub topik ini membahas tentang bagaimana metode/cara pengambilan sampel gas/odor untuk menjamin pengukuran yang valid dari gas yang akan diidentifikasi dengan menggunakan *array sensor* TGS. Hal ini terkait dengan bagaimana mengatur *electronic valve* dan *mini air pump* agar konsentrasi gas yang masuk kedalam *vacuum chamber* memiliki volume yang cukup akurat untuk digunakan sebagai pengujian sesuai dengan waktu tertentu. *Valve1* dan *valve2* digunakan untuk mengatur gas yang akan masuk ke dalam *vacuum chamber*, *Valve1* digunakan untuk mengontrol gas O<sub>2</sub> sebagai gas pembersih, *Valve2* digunakan untuk mengatur gas/odor dari sampel, *valve3* digunakan untuk mengatur luaran gas dari *vacuum chamber*. *Relay1* terhubung dengan rangkaian Catu daya *array sensor* TGS, sedangkan *relay2* terhubung dengan *mini air pump*. Data gas diperoleh dari luaran 4 buah sensor array TGS yang berupa tegangan. Tegangan tersebut kemudian dikonversi dari analog menjadi digital dengan *ADC Internal* yang dimiliki oleh robotino. Aliran dan pengendalian *valve* terhadap sample gas/odor terlihat di Tabel 1.

Tabel 1. Pengendalian Aliran Gas dan *Electronic Valve*

State	Valve1	Valve2	Valve3	Air Pump	Durasi
<i>Heating Sensor</i>	Off	Off	Off	Off	60 s
<i>Cleaning Vacuum Chamber</i>	On	Off	On	On	180 s
<i>Gas Sampling (Intake)</i>	Off	On	Off	On	120 s
<i>Gas Sampling (data acquisition)</i>	Off	Off	Off	Off	60 s

Berdasar pada Tabel 1 maka berikut ini adalah *script* yang mengatur bagaimana *input* dan *output* digital dari robot bekerja sesuai dengan rancangan.

```
Timer timer;
AnalogInput analogInput0;
AnalogInput analogInput1;
AnalogInput analogInput2;
AnalogInput analogInput3;

DigitalOutput digitalOutput0;
DigitalOutput digitalOutput1;
DigitalOutput digitalOutput2;
Relay relay1;
Relay relay2;

void nose(float & data1, float & data2, float & data3, float & data4)
{
    //Heating sensor for 60'
    relay2.setValue (true);
    Sleep(30000);
    //Clean_Chamber for 180'
    digitalOutput0.setValue (true);
    digitalOutput1.setValue (false);
    digitalOutput2.setValue (true);
    relay1.setValue (true);
    Sleep(60000);
    //Odor Sampling (intake) for 120'
    digitalOutput0.setValue (false);
    digitalOutput1.setValue (true);
    digitalOutput2.setValue (false);
    relay1.setValue (true);
    Sleep(30000);
    //sampling for 60'
    digitalOutput0.setValue (false);
    digitalOutput1.setValue (false);
    digitalOutput2.setValue (false);
    relay1.setValue (false);
    Sleep(2000);

    data1=analogInput0.value();
    data2=analogInput1.value();
    data3=analogInput2.value();
    data4=analogInput3.value();
    //relay2.setValue (false);
}
```

## Perancangan Input dan Output MLP

Bentuk jaringan MLP sangat berkaitan erat dengan jumlah input dan output yang akan digunakan. Bentuk *input layer* dan *output layer* untuk sebuah kasus MLP tidak dapat diubah.

Dalam penelitian ini digunakan 4 buah sensor gas yang membentuk *array sensor*, yaitu TGS2600, TGS2610, TGS2611, dan TGS2612. Masing-masing sensor akan menghasilkan nilai dengan respon yang berbeda-beda ketika mendekksi kandungan tertentu pada gas. Oleh karena itu, jumlah neuron untuk layer input didefinisikan berjumlah 4 neuron. Nilai yang dapat dihasilkan masing-masing sensor adalah tegangan dengan nilai antara 0 – 5 Volt DC.

Sedangkan target dari penelitian ini adalah untuk mengenali gas alkohol dan bensin, yang seharusnya diwakili dengan 2 buah neuron untuk *output layer*. Namun, akan lebih baik jika pada *output layer* ditambahkan 1 buah neuron lagi yang mewakili output untuk gas-gas yang tidak dikenal. Sehingga didapatkan jumlah neuron untuk *output layer* adalah 3 buah neuron. Masing-masing output didefinisikan sesuai dengan Tabel 2.

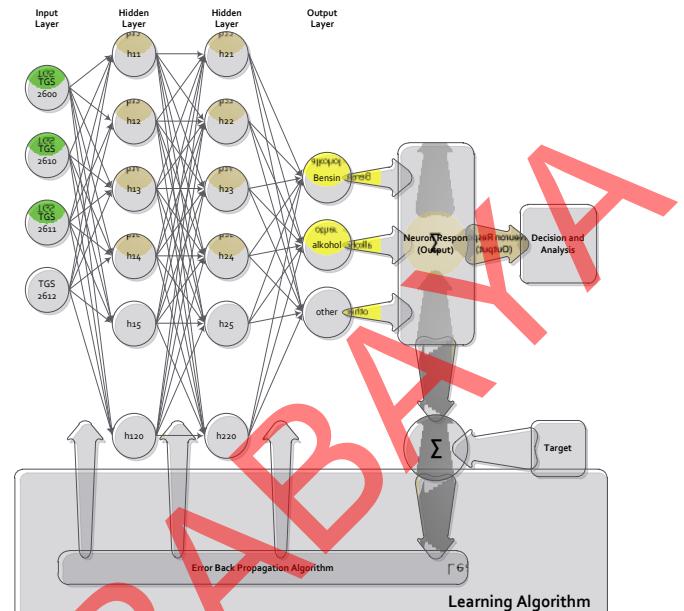
Tabel 2. Korelasi Antara Neuron Output dengan Jenis Gas

Output Layer			Gas yang Dikenali
Neuron 1	Neuron 2	Neuron 3	
1	0	0	Bensin
0	1	0	Alkohol
0	0	1	Lainnya

## Perancangan MLP pada Komputer (Error Back Propagation)

Pada Gambar terdapat *input layer* dengan 4 buah neuron, yang nilainya masing-masing akan diperoleh dari *array sensor* gas (TGS2600, TGS2610, TGS2611, TGS2612) yang tampak pada gambar. Selain itu, pada *output layer* terdapat 3 buah neuron, yang masing-masing neuron mewakili 1 jenis gas yang dideteksi. Sedangkan jumlah neuron pada *hidden layer* yang digunakan tidak terbatas seperti pada gambar, begitu pula halnya dengan jumlah

*hidden layer* itu sendiri. Rancangan yang digunakan pada penelitian ini akan menggunakan 2 *hidden layer* yang masing-masing memiliki 20 buah *neuron*.



Gambar 6. Bentuk Jaringan MLP yang Digunakan

Secara umum maka dapat digambarkan aliran dari proses perancangan seperti yang ditunjukkan pada Gambar.

Nilai iterasi maksimal (*max\_i*) adalah sebuah nilai yang digunakan untuk membatasi *looping* pada proses *learning*, dalam penelitian ini nilai iterasi dibatasi sebanyak  $10^9$  atau 1.000.000.000 iterasi. Satu iterasi adalah melakukan proses *feed forward*, mencari error, dan memperbaiki bobot sebanyak seluruh pola atau data pelatihan yang akan dilatihkan.

Nilai Error minimal atau *least SSE* (*e\_min*) digunakan untuk menentukan bahwa proses MLP tersebut telah *convergen*. *Convergen* adalah kondisi dimana nilai SSE dari *error* yang didapatkan untuk satu iterasi lebih kecil dari *least SSE* yang telah ditentukan. Semakin kecil nilai *least SSE* yang diberikan maka MLP menjadi semakin akurat, tetapi hal ini akan menyebabkan proses *learning* akan membutuhkan waktu yang semakin lama. Dalam penelitian ini nilai *least SSE* yang ditentukan adalah  $10^{-5}$ .

Proses *feed forward* terdiri dari beberapa langkah, yaitu :

- Menghitung  $u$  dengan mengakumulasikan seluruh neuron input dikalikan dengan bobot masing-masing, sesuai dengan

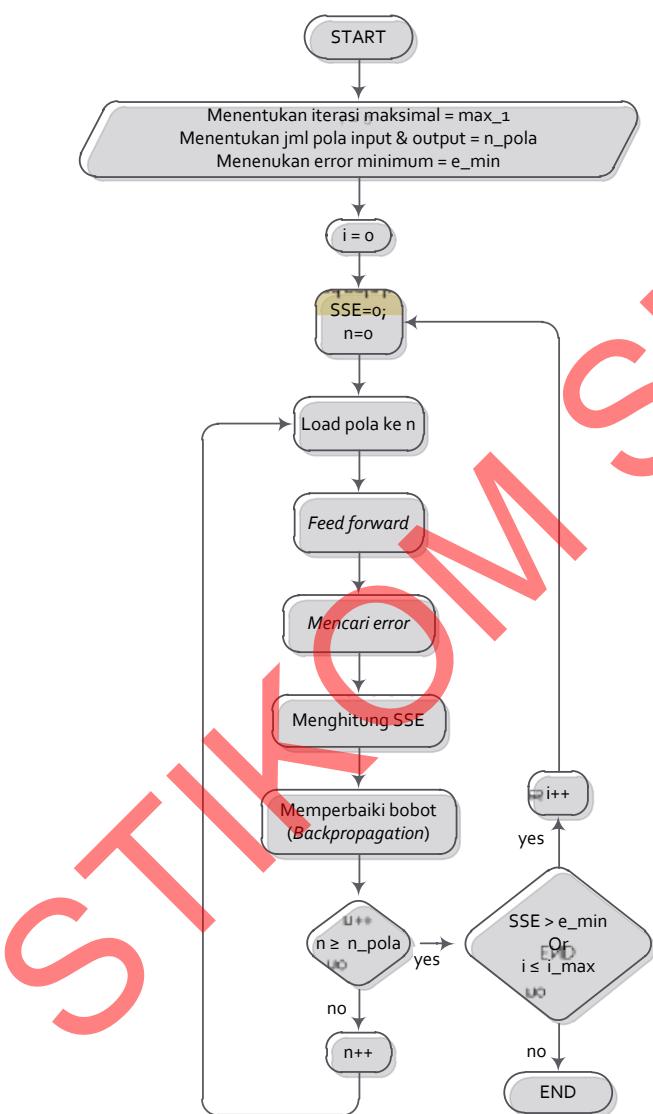
$$u_q = \sum x_j * w_{qj} \quad (1)$$

- Menghitung nilai  $v$  dengan menambahkan hasil akumulasi neuron ( $u$ ) dengan nilai bias sesuai dengan

$$v_q = u_q + b_q \quad (2)$$

- Melakukan aktifasi dengan menggunakan fungsi aktivasi sigmoid bipolar terhadap terhadap fungsi  $v$ , dengan nilai asimtot atau  $\alpha = 1$ .

$$y_q = f_{bs}(v_q) = \frac{1}{1+e^{-\alpha v_q}} \quad (3)$$



Gambar 7. Gambar Aliran Perancangan MLP pada PC Secara Umum

Dari beberapa langkah untuk merancang proses *feed forward* maka berikut ini adalah *script* program yang digunakan

```

'Feed Forward
For i=1 To nLayer
For j=0 To (nNeuron(i) - 1)
sigmaNeuron(i,j)=0
'mencari u
For k=0 To (nNeuron(i-1)-1)
sigmaNeuron(i,j) = sigmaNeuron(i,j) +
neuron((i-1),k)*weight(i-1,k,j)
Next k

'perhitungan bias (mencari v)
sigmaNeuron(i,j) = sigmaNeuron(i,j) +
bias(i,j)

'fungsi aktivasi (mencari y)
neuron(i,j) = 1/(1+Exp(-
1*asimtot*sigmaNeuron(i,j)))
Next j
Next i
  
```

Proses *feed forward* akan menghasilkan sebuah nilai pada neuron-neuron *output*, nilai ini selanjutnya akan dibandingkan dengan nilai target yang bersesuaian dengan *input* yang dilatihkan. Nilai selisih antara target dengan nilai *output* neuron pada *output layer* atau error yang terjadi akan digunakan untuk meneruskan atau menghentikan proses pembelajaran dan juga digunakan untuk menyesuaikan nilai bobot hingga mencapai kondisi yang *convergen*. Nilai bobot tersebut selanjutnya akan disimpan dalam sebuah *file*, jika kondisi *convergen* tidak tercapai dan iterasi sudah mencapai maksimal maka proses *feed forward* dapat diulangi lagi dengan menggunakan kembali nilai bobot yang telah tersimpan dalam *file* tersebut.

Terdapat perbedaan cara menghitung error antara *output layer* dengan layer sebelum *output layer* (*hidden layer*), sehingga dalam perancangan ini untuk menghitung error pada *output layer* dapat menggunakan persamaan 4 yang bisa ditulis dalam script program sebagai berikut

$$\delta_j^{(s)} = (d_{qj} - x_{out,j}^{(s)})g(v_j^{(s)}) \quad (4)$$

```

'Perhitungan dell & error untuk layer output
For i=0 To (nNeuron(nLayer)-1)
dell(nLayer,i)=(desired(i)-
neuron(nLayer,i))*asimtot*neuron(nLayer,
i)*(1-neuron(nLayer,i))
SSE=(SSE+((desired(i)-neuron(nLayer, i))^2))
Next i

```

Sedangkan untuk menghitung error pada layer selain output layer dapat menggunakan persamaan 5 sehingga dapat dituliskan kedalam script program sebagai berikut

$$\delta_j^{(s)} = \left( \sum_{h=1}^{n_{s+1}} \delta_h^{(s+1)} w_{hj}^{(s+1)} \right) g(v_j^{(s)}) \quad (5)$$

```
'Perhitungan error untuk layer yang lain
```

```

For i = (nLayer-1) To 1 Step-1
For j = 0 To nNeuron(i)
dell(i,j) = 0
For k = 0 To nNeuron(i + 1)
dell(i,j) = dell(i,j)+ (dell(i+1,k) *
weight(i,j,k))
Next k
dell(i,j) =
dell(i,j)*asimtot*neuron(i,j)*(1-
neuron(i,j))
Next j
Next i

```

Setelah mendapatkan error dan dell untuk masing-masing neuron pada setiap layer tahap selanjutnya adalah memperbaiki bobot untuk masing-masing *synapses*. Hal ini akan terus menerus dilakukan hingga salah satu atau dua syarat terpenuhi yaitu SSE lebih kecil dari Minimum Error yang diharapkan atau jumlah iterasi telah melebihi nilai iterasi maksimal yang telah ditentukan. Untuk memperbaiki bobot dapat menggunakan persamaan 6, sehingga dapat dituliskan script program sebagai berikut

$$w_{ji}^{(s)}(k+1) = w_{ji}^{(s)}(k) + \mu^{(s)} \delta_j^{(s)} x_{out,i}^{(s-1)} \quad (6)$$

```

'Update Weight
If SSE > MaxError Then
'Update weight & bias
For k = nLayer To 1 Step-1
For i = 0 To (nNeuron(k-1)-1)
For j = 0 To (nNeuron(k)-1)
weight((k-1),i,j) = weight((k-1),i,j) +
(myu*dell(nLayer,j) * neuron((k-1),i))
Next j
bias(k,i) = bias(k,i) + (myu*dell(nLayer,j))
Next i
Next k
End If

```

## Perancangan MLP Pada Robot (Feed Forward)

Perancangan MLP pada robot bertujuan untuk menerapkan NN pada robot agar mampu mengidentifikasi gas yang dijadikan sebagai objek penelitian. Hal ini sesuai dengan tujuan penelitian yang salah satunya adalah mengurangi resiko akibat kontak langsung antara manusia dengan zat berbahaya. Secara umum Perancangan MLP pada robot ini terdiri dari tiga bagian. Pertama adalah memuat bobot yang telah didapatkan pada proses pelatihan, yang kedua adalah proses *feed forward*, yang ketiga adalah proses evaluasi dari masukan sensor gas dengan luaran pada *output layer MLP*. Hasil luaran dari *output layer MLP* ini menetukan jenis gas yang dideteksi oleh robot. Untuk mengidentifikasi jenis gas yang dideteksi dalam perancangan ini disesuaikan dengan Tabel 2 Korelasi Antara Neuron Output dengan Jenis Gas, sehingga tujuan akhir dari penelitian ini dapat tercapai, yaitu robot mampu mengidentifikasi gas dengan menggunakan MLP. Secara umum diagram alir perancangan untuk MLP pada robot ini dapat dilihat pada Gambar.

Berikut ini adalah script pada robot untuk proses *feed forward*

```

void feedforward(double tgs2600, double
tgs2610, double tgs2611, double tgs2612)
{
    nInput = nNeuron[0];
    nOutput = nNeuron[nLayer];
    neuron[0][0] = tgs2600;
    neuron[0][1] = tgs2610;
    neuron[0][2] = tgs2611;
    neuron[0][3] = tgs2612;
}

```

```

for(int i = 1; i <= nLayer; i++)
{
    for(int j = 0; j <= (nNeuron[i]-1); j++)
    {
        sigmaNeuron[i][j] = 0;
    for(int k = 0; k <= (nNeuron[i-1]-1); k++)
    {
        sigmaNeuron[i][j] =
sigmaNeuron[i][j] + neuron[(i-1)][k] *
weight[i-1][k][j];
    }
    sigmaNeuron[i][j] =
sigmaNeuron[i][j] + bias[i][j];
    neuron[i][j] = (double) 1/(1+exp(-1
* asimtot * sigmaNeuron[i][j]));
}
}
}

```

Berikut ini adalah script pada robot untuk proses evaluasi hasil luaran dari *output layer* MLP.

```

int evaluation()
{
    int temp = 0;
    int dua = 1;
for(int i = 0; i <= (nNeuron[nLayer] - 1);
i++)
{
    if(neuron[nLayer][i] > 0.5)
temp = temp + (dua * 1);
    else
temp = temp + (dua * 0);
dua = dua * 2;
}
if(temp == 1)
return(0);
elseif(temp == 2)
return(1);
elseif(temp == 4)
return(2);
else
return(3);
}

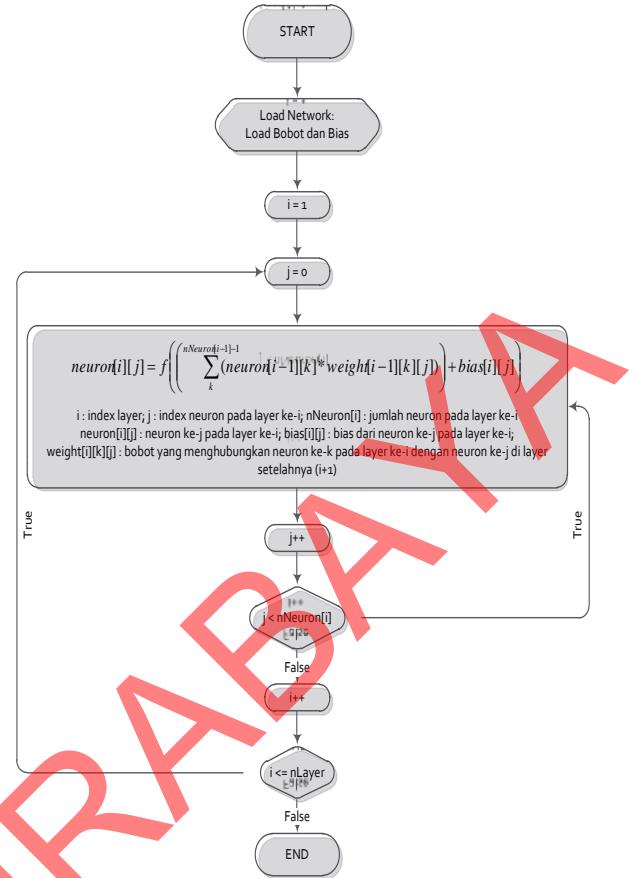
```

Dari hasil evaluasi maka sesuai dengan Tabel 2 maka script dibawah ini digunakan untuk mengidentifikasi gas yang diamati.

```

int hasil = evaluation();
switch(hasil)
{
case 0:
    std::cout << "Bensin"; break;
case 1:
    std::cout << "Alkohol"; break;
case 2:
    std::cout << "Lainnya"; break;
default:
    std::cout << "????"; break;
}

```



Gambar 8. Diagram Alir Perancangan MLP pada Robot

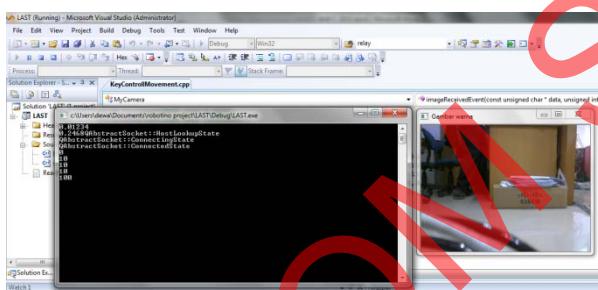
## HASIL

Tabel 3 menunjukkan bahwa robot sepenuhnya berhasil dikendalikan oleh PC melalui program yang telah dirancang, program tersebut membaca karakter yang telah ditentukan dari keyboard. Penekanan terhadap karakter tertentu pada keyboard mengaktifkan prosedur didalam program yang mengatur arah serta kecepatan dari pergerakan robot.

Tabel 3. Kendali Pergerakan Robot

<b>Input</b>	<b>Hasil</b>
Keyboard key "↑" (scan code 0x48)	BERHASIL (robot bergerak maju)
Keyboard key "↓" (scan code 0x50)	BERHASIL (robot bergerak mundur)
Keyboard key "←" (scan code 0x4B)	BERHASIL (robot bergerak ke kiri)
Keyboard key "→" (scan code 0x4D)	BERHASIL (robot bergerak ke kanan)
Keyboard key "q" (scan code 0x71)	BERHASIL (robot berputar ke kiri)
Keyboard key "w" (scan code 0x77)	BERHASIL (robot berputar ke kanan)
Keyboard key "+" (scan code 0x2B)	BERHASIL (kecepatan gerakan robot bertambah)
Keyboard key "-" (scan code 0x2D)	BERHASIL (kecepatan gerakan robot berkurang)
Keyboard key "space" (scan code 0x20)	BERHASIL (robot berhenti)
Keyboard key "enter" (scan code 0x0D)	BERHASIL (prosedur sampling gas berjalan)

Pada Gambar menunjukkan bahwa PC mampu menampilkan data gambar streaming yang ditangkap oleh webcam yang terpasang pada Robot. Gambar streaming ditampilkan melalui window Gambar Warna.

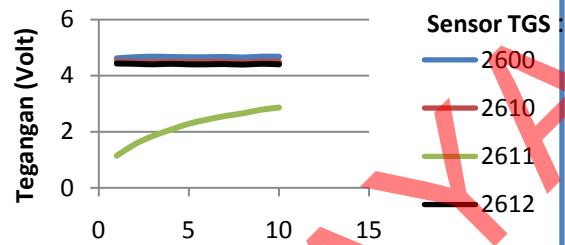


Gambar 9. Hasil Streaming Yang Ditampilkan Oleh PC

Gambar menunjukkan grafik respon dari array sensor TGS terhadap bensin, sedangkan Gambar menunjukkan respon array sensor TGS terhadap Alkohol, Gambar adalah respon array sensor terhadap metana, dan gambar 13 merupakan respon array sensor terhadap solar. Data luaran dari array sensor yang mewakili respon untuk masing-masing gas nantinya akan digunakan sebagai data latih pada proses back propagation. Sebagai hasil pelatihan adalah sebuah bobot untuk masing-masing neuron sesuai dengan desain dari nural network yang

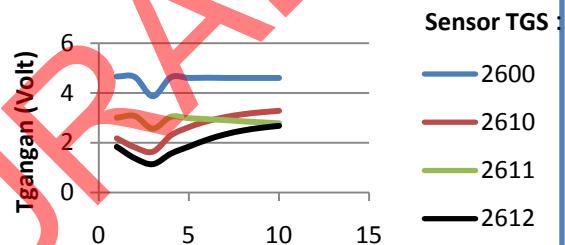
sebelumnya. Selanjutnya bobot tersebut akan digunakan sebagai acuan pada proses feed forward pada robot untuk mengidentifikasi gas.

### Grafik Respon TGS Terhadap Bensin



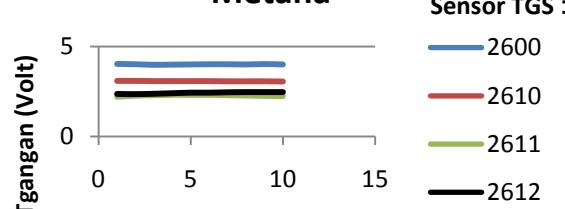
Gambar 10. Grafik Respon TGS Terhadap Bensin

### Grafik Respon TGS Terhadap Alkohol



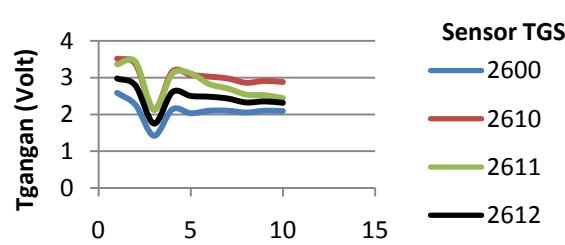
Gambar 11. Grafik Respon TGS Terhadap Alkohol

### Grafik Respon TGS Terhadap Metana



Gambar 12. Grafik Respon TGS Terhadap Metana

### Grafik Respon TGS Terhadap Solar



Gambar 13. Grafik Respon TGS Terhadap Solar

MLP yang telah drancang pada robot mampu mengidentifikasi gas yang diuji. Pengujian dilakukan dengan menempatkan Bensin, solar, Alkohol, dan Metana kedalam sebuah cawan dengan ukuran yang sama, robot bergerak menuju cawan dan memulai proses penyamplingan gas. Hasilnya adalah seperti yang ditunjukkan pada Tabel 4 dan 5, yang pada Tabel 4 pengujian dilakukan secara berurutan, yaitu dengan 10 sampel bensin, 10 sampel alkohol, 10 sampel metana dan 10 sampel solar. Sebagai hasil, MLP mampu mengidentifikasi gas apa yang terdapat di dalam cawan. Pada Tabel 5 pengujian dilakukan dengan mengacak keempat sampel tersebut, masing-masing disampel sebanyak 10 kali, sehingga pengambilan data keseluruhan dilakukan sebanyak 40 kali. Hasil terlihat di Tabel 5, yang dapat diambil kesimpulan bahwa MLP telah berhasil 100% membedakan bensin dan alkohol, serta selain bensin dan alkohol diidentifikasi sebagai “Lainnya”. Hal ini membuktikan bahwa bobot yang didapatkan pada saat proses pelatihan setelah diterapkan pada robot mampu memberikan kemampuan robot untuk mengidentifikasi gas. Bobot dihasilkan pada saat pelatihan hingga kondisi konvergen dicapai dengan iterasi sebanyak 781.338.964 dan SSE sebesar 9,99989696745863e-05.

Tabel 4. Pengujian Identifikasi Gas Secara Berurutan

Pengujian Ke	Gas yg diuji	Luaran MLP (Diidentifikasi)
1	Bensin	Bensin
2	Bensin	Bensin
3	Bensin	Bensin
4	Bensin	Bensin
5	Bensin	Bensin
6	Bensin	Bensin
7	Bensin	Bensin
8	Bensin	Bensin
9	Bensin	Bensin
10	Bensin	Bensin
11	Alkohol	Alkohol
12	Alkohol	Alkohol
13	Alkohol	Alkohol
14	Alkohol	Alkohol
15	Alkohol	Alkohol
16	Alkohol	Alkohol
17	Alkohol	Alkohol
18	Alkohol	Alkohol
19	Solar	Lainnya
20	Bensin	Bensin
21	Solar	Lainnya
22	Solar	Lainnya
23	Solar	Lainnya
24	Solar	Lainnya
25	Solar	Lainnya
26	Solar	Lainnya
27	Solar	Lainnya
28	Solar	Lainnya
29	Solar	Lainnya
30	Solar	Lainnya
31	Metana	Lainnya
32	Metana	Lainnya
33	Metana	Lainnya
34	Metana	Lainnya
35	Metana	Lainnya
36	Metana	Lainnya
37	Metana	Lainnya
38	Metana	Lainnya
39	Metana	Lainnya
40	Metana	Lainnya

Pengujian Ke	Gas yg diuji	Luaran MLP (Diidentifikasi)
19	Alkohol	Alkohol
20	Alkohol	Alkohol
21	Solar	Lainnya
22	Solar	Lainnya
23	Solar	Lainnya
24	Solar	Lainnya
25	Solar	Lainnya
26	Solar	Lainnya
27	Solar	Lainnya
28	Solar	Lainnya
29	Solar	Lainnya
30	Solar	Lainnya
31	Metana	Lainnya
32	Metana	Lainnya
33	Metana	Lainnya
34	Metana	Lainnya
35	Metana	Lainnya
36	Metana	Lainnya
37	Metana	Lainnya
38	Metana	Lainnya
39	Metana	Lainnya
40	Metana	Lainnya

Tabel 5. Pengujian Identifikasi Gas Secara Random

Pengujian Ke	Gas yg diuji	Luaran MLP (Diidentifikasi)
1	Bensin	Bensin
2	Solar	Lainnya
3	Alkohol	Alkohol
4	Bensin	Bensin
5	Bensin	Bensin
6	Metana	Lainnya
7	Solar	Lainnya
8	Alkohol	Alkohol
9	Metana	Lainnya
10	Bensin	Bensin
11	Metana	Lainnya
12	Bensin	Bensin
13	Solar	Lainnya
14	Alkohol	Alkohol
15	Metana	Lainnya
16	Metana	Lainnya
17	Bensin	Bensin
18	Alkohol	Alkohol
19	Solar	Lainnya
20	Bensin	Bensin
21	Solar	Lainnya
22	Metana	Lainnya
23	Alkohol	Alkohol
24	Solar	Lainnya
25	Bensin	Bensin
26	Bensin	Bensin
27	Alkohol	Alkohol
28	Solar	Lainnya
29	Metana	Lainnya
30	Alkohol	Alkohol
31	Bensin	Bensin
32	Solar	Lainnya
33	Bensin	Bensin

Pengujian Ke	Gas yg diuji	Luaran MLP (Diidentifikasi)
34	Solar	Lainnya
35	Alhokol	Alkohol
36	Metana	Lainnya
37	Alkohol	Alkohol
38	Bensin	Bensin
39	Metana	Lainnya
40	Solar	Lainnya

## SIMPULAN

Berdasar hasil penelitian dapat disimpulkan sebagai berikut.

1. Dengan menerapkan *electronic nose* pada robot yang secara *remote* dikendalikan oleh manusia dapat mengurangi resiko yang ditimbulkan akibat interaksi secara langsung antara manusia dan gas berbahaya.
2. Multi layer perceptron yang dirancang 100% berhasil membedakan gas alkohol bensin, serta gas lain yang tidak dikenal diklasifikasikan sebagai gas lainnya.

## RUJUKAN

- Doroftei , Ioan, (dkk.). 2007. “*Omnidirectional Mobile Robot – Design and Implementation*”. (Online).[http://www.intechopen.com/source/pdfs/465/InTech-omnidirectional\\_mobile\\_robot\\_design\\_and\\_implementation.pdf](http://www.intechopen.com/source/pdfs/465/InTech-omnidirectional_mobile_robot_design_and_implementation.pdf), diakses 26 Januari 2012
- Ham, M Fredric., Kostanic, Ivica. 2001, *Principles of Neurocomputing for Science and Engineering*. McGraw Hill.
- Graham, Brian Barkley. 2000. *Using an Accelerometer Sensor To Measure Human Hand Motion*. Department of Electrical Engineering and Computer Science . Massachusetts Institute of Technology . Massachusetts.
- Karras, Ulrich. 2011. *Robotino – An Open Learning Mobile Robot System for Robocup*. (online). ([http://www.festodidactic.com/ov3/media/customers/1100/festo\\_robocup.pdf](http://www.festodidactic.com/ov3/media/customers/1100/festo_robocup.pdf), diakses tanggal 25 Januari 2012).
- Loutfi, Amy,. Coradeschi, Silvia. 2002. Relying on an electronic nose for odor localization. IEEE pages 46 – 50.
- Purbo, Onno W. 2011. *Jaringan Komputer Menggunakan Protokol TCP / IP*. (Online). (<http://onno.vlsm.org/v09/onno-ind-1/network/jaringan-komputer-menggunakan-protokol-tcpip-01-1992.rtf>, diakses tanggal 16 Agustus 2011). Canada.
- Rajagukguk, Agatha Rebecca. 2009. *Inovasi dan Kreatifitas Layanan Penelusuran, Penggunaan Internet dan Jurnal Online pada Perpustakaan Universitas Sumatera Utara*. (Online). (<http://repository.usu.ac.id/bitstream/123456789/13596/1/09E02383.pdf>, diakses tanggal 25 Januari 2012).
- Robotino.2010. *OpenRobotino API How To* (Online). (<http://doc.openRobotino.org/documentation/OpenRobotinoApiHowTo/HTML/index.html>). Diakses tanggal 1 September 2011
- ROS. 2010. *Robots Using ROS Robotino*. (Online). (<http://www.ros.org/news/2010/06/robots-using-ros-Robotino.html>). Diakses 29 September 2011
- Syam, Syamsul. 2011. “*Perancangan Robot Line Follower Menggunakan Omniwheel Berbasis Mikrokontroler BS2P40*”. (Online). <http://elib.unikom.ac.id/gdl.php?mod=browse&op=read&id=jbptunikompp-gdl-syamsulsyam-26361&newlang=indonesian>, diakses 26 Januari 2012