

BAB III

METODE PENELITIAN

Metode penelitian yang digunakan pada pembuatan perangkat keras dan perangkat lunak yaitu dengan studi kepustakaan. Dengan cara ini penulis berusaha untuk mendapatkan dan mengumpulkan data-data, informasi, konsep-konsep yang bersifat teoritis dari buku bahan-bahan kuliah dan referensi dari internet yang berkaitan dengan permasalahan.

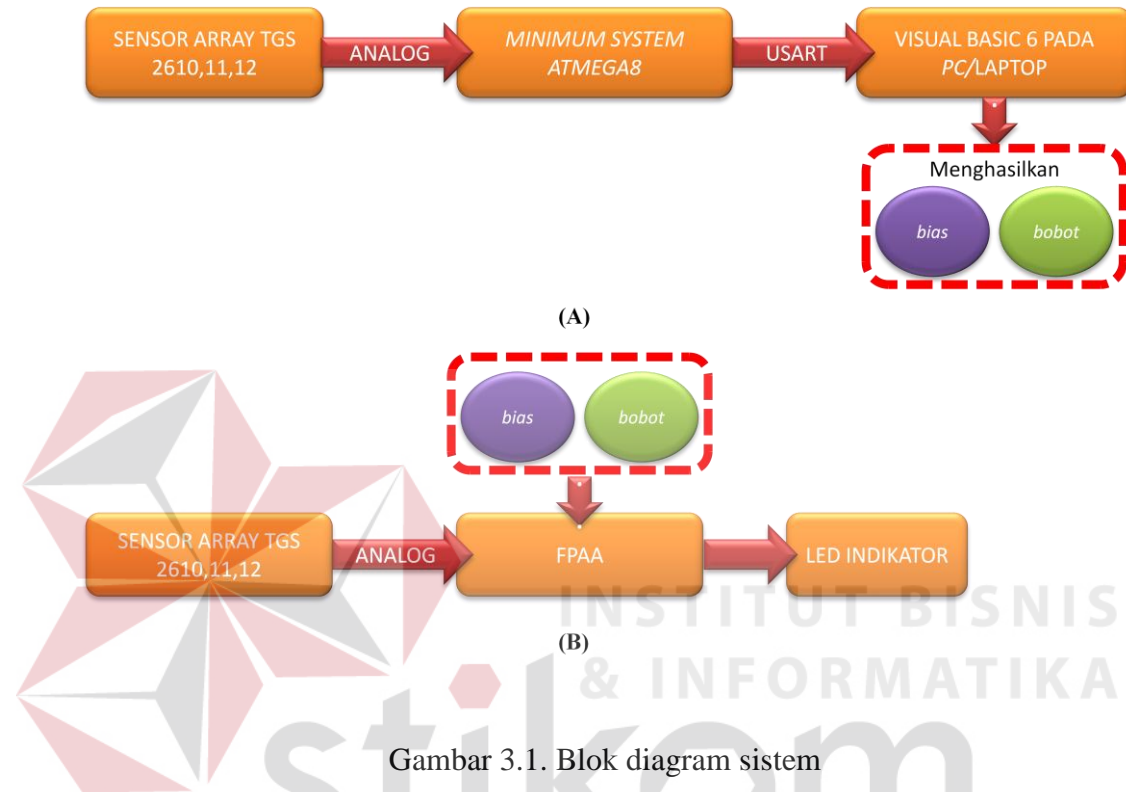
Dari data-data yang diperoleh maka dilakukan perencanaan rangkaian perangkat keras. Dalam perangkat keras ini, penulis akan melakukan pengujian perangkat keras dengan program-program yang telah dibuat, Pembuatan perangkat lunak adalah tahap selanjutnya. Terakhir adalah penggabungan perangkat keras dengan kerja perangkat lunak yang telah selesai dibuat.

Pada BAB III dibahas mengenai masalah yang timbul dalam perencanaan dan pembuatan perangkat keras (*hardware*) maupun perangkat lunak (*software*). Dari kedua bagian tersebut akan dipadukan agar dapat bekerja sama untuk menjalankan sistem yang baik.

Perencanaan ini diperlukan sebelum proses pembuatan *system* tersebut, Perancangan ini berguna agar pengerjaan tahapan selanjutnya berjalan dengan lancar. Tahapan-tahapannya meliputi tahap pembuatan perangkat keras, perangkat lunak dan menggabungkan keduanya.

3.1. Diagram Blok

Dalam pembahasan tentang proses keseluruhan yang dapat di jelaskan pada diagram blok seperti Gambar 3.1.



Gambar 3.1. Blok diagram sistem

Dari blok diagram pada Gambar 3.1. Gambar bagian (A) memperlihatkan blok pelatihan jaringan saraf tiruan, sedangkan Gambar bagian (B) memperlihatkan pengujian jaringan saraf tiruan. Sistem ini yang terdiri dari berbagai modul antara lain modul sensor *array*, modul *minimum system* ATmega8, pemrograman yang terdapat pada komputer, modul FPAA AN231K04. Proses pertama sensor *array* TGS menerima *input* berupa gas-gas kimia. Kemudian *output* dari sensor *array* TGS berupa data *analog* akan diterima *microcontroller* dan dikelola sesuai program yang ada hingga mengirimkan data digital kepada komputer melalui komunikasi serial (UART). Data-data yang

diterima komputer akan disimpan dalam *database* dan diproses oleh program jaringan saraf tiruan hingga mendapatkan nilai bobot dan bias yang sesuai, nilai bobot dan bias yang didapatkan dari proses pelatihan akan digunakan sebagai parameter bobot dan bias pada jaringan saraf tiruan di FPAA. Data dari sensor *array* TGS akan diujikan pada jaringan saraf tiruan di modul FPAA sehingga menghasilkan *output* tentang jenis gas yang telah terdeteksi. Indikator hasil keputusan FPAA disajikan dalam bentuk LED.

3.2 Perancangan perangkat keras

3.2.1. Rangkaian *array* sensor TGS

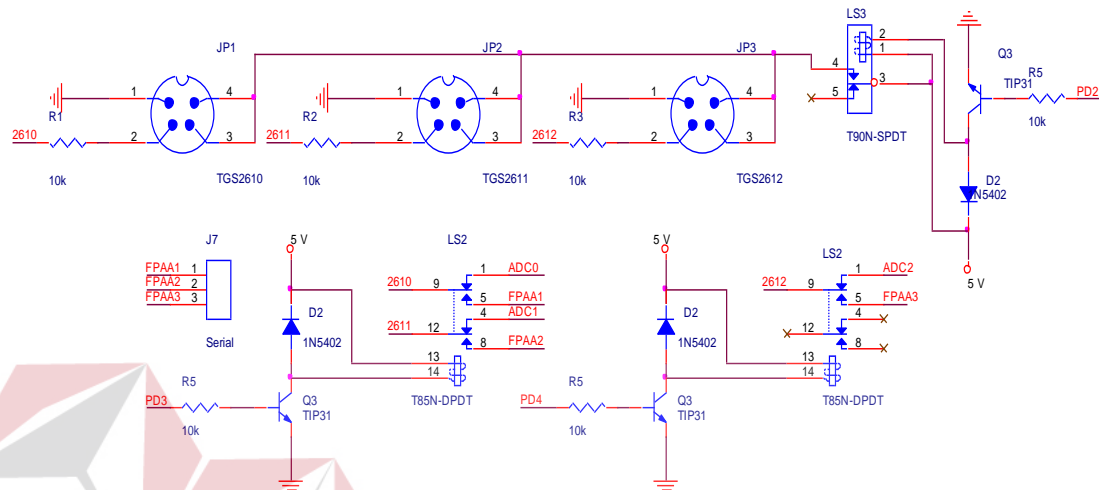
Dalam tugas akhir ini digunakan 3 buah sensor TGS dari *Figaro* yaitu TGS 2610, 2611, dan 2612 yang memiliki karakteristik hampir mirip antara yang satu dengan lainnya. Sesuai dengan *datasheet*, karakteristik gas yang dapat terdeteksi yaitu gas-gas yang mengandung *ethanol*, *methane*, *iso-butane* dan *propane*.

Agar dapat bekerja dengan baik sensor ini membutuhkan dua tegangan masukan. *Heater Voltage* (V_H) digunakan sebagai tegangan *heater* dan *Circuit Voltage* (V_C) 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 . Nilai *power dissipation* (P_S) dan hambatan sensor (R_S) dapat dihitung dengan persamaan (3.1) dan (3.1).

$$P_S = \frac{(V_C - V_{RL})^2}{R_S} \quad (3.1)$$

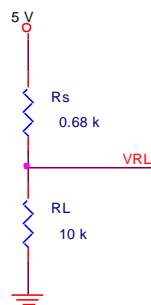
$$R_S = \frac{V_C - V_{RL}}{V_{RL}} \times R_L \quad (3.2)$$

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 3.2.



Gambar 3.2. Rangkaian modul sensor *array* TGS

Pada Gambar 3.2, satu *relay* yang digunakan untuk memilih tegangan V_C dan V_H dan dua *relay* yang digunakan untuk memilih *output* dari TGS, yaitu untuk diarahkan ke *microcontroller* atau ke FPAA. Resistansi yang digunakan pada nilai *resistor* beban (R_L) adalah 10 k Ω sedangkan nilai hambatan sensor (R_S) pada *datasheet* rata-rata 0,68 k Ω sampai 6,8 k Ω . Gambar 3.3 adalah ilustrasi rangkaian saat TGS menghasilkan nilai R_S minimal yaitu 0,68



Gambar 3.3. Rangkaian sensor dengan $R_S = 0,68 \text{ k}$

Perhitungan rangkaian pada Gambar 3.3 adalah sebagai berikut:

$$R_{Smin} = 0,68 \text{ k}\Omega$$

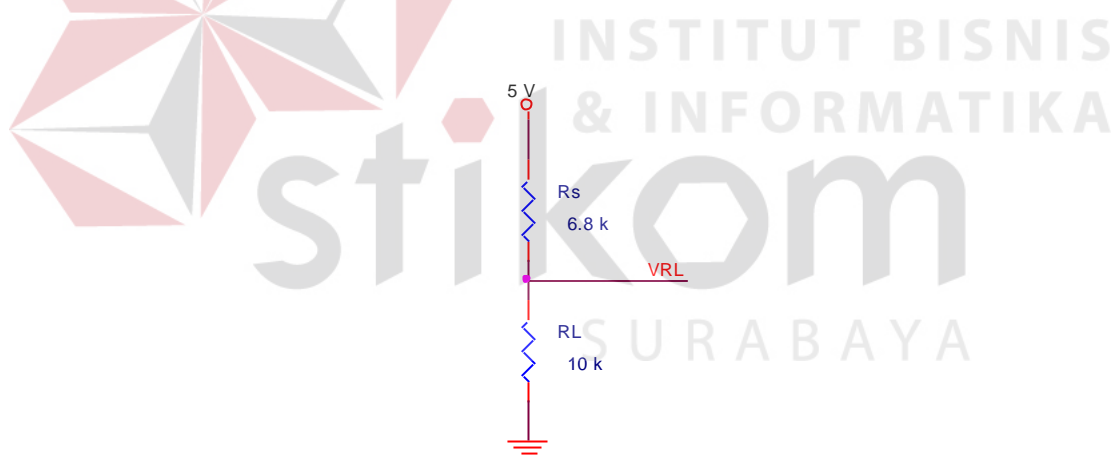
$$R_L = 10 \text{ k}\Omega$$

$$V_{RL} = \frac{R_L}{R_{Smin} + R_L} \times V_C \quad (3.3)$$

$$V_{RL} = \frac{10000}{680 + 10000} \times 5 = 4,682 \text{ Volt}$$

$$P_S = \frac{(V_C - V_{RL})^2}{R_S} = \frac{(5 - 4,682)^2}{680} = 1,4871 \times 10^{-4} \text{ Watt}$$

Perhitungan diatas menghasilkan nilai $P_S = 1,4871 \times 10^{-4}$ sehingga masih di bawah 15 mW, kemudian rangkaian saat R_{Smax} terlihat pada Gambar 3.4.



Gambar 3.4. Rangkaian sensor dengan $R_S = 6,8 \text{ k}$

Perhitungan rangkaian pada Gambar 3.4 adalah sebagai berikut:

$$R_{Smax} = 6,8 \text{ k}\Omega$$

$$R_L = 10 \text{ k}\Omega$$

$$V_{RL} = \frac{R_L}{R_{Smax} + R_L} \times V_C \quad (3.4)$$

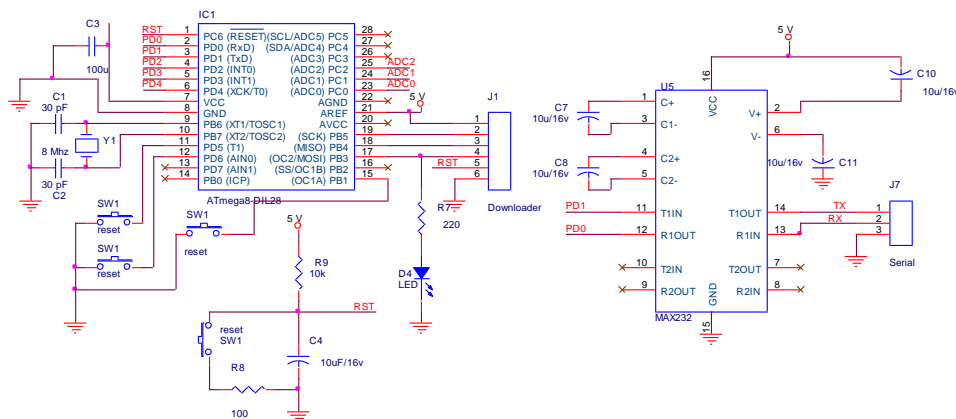
$$V_{RL} = \frac{10000}{6800 + 10000} \times 5 = 2,9762 \text{ Volt}$$

$$P_S = \frac{(V_C - V_{RL})^2}{R_S} = \frac{(5 - 2.9762)^2}{6800} = 6,0232 \times 10^{-4} \text{ Watt}$$

Perhitungan diatas menghasilkan nilai $P_S = 6,0232 \times 10^{-4}$ sehingga masih dibawah 15 mW dan dapat disimpulkan dengan resistor beban (R_L) = 10 k Ω adalah cukup optimal.

3.2.2. Minimum system

Minimum system ini dirancang untuk *microcontroller* ATmega8(L), dalam perancangannya ini memerlukan beberapa komponen pendukung seperti kristal, resistor dan variabel resistor, dan kapasitor. Rangkaian ini dalam istilah lainnya disebut *minimum system* ATmega8(L). *Microcontroller* berfungsi untuk mengatur pengambilan data *analog* dari sensor *array* TGS menuju *personal* komputer maupun FPAA. Di dalam *microcontroller* ATmega8 sudah dilengkapi dengan ADC yang terletak di pin PC0 – PC5, pin yang digunakan untuk membaca hasil keluaran dari sensor *array* TGS. Gambar *minimum system* ATmega8 dapat dilihat pada Gambar 3.5, sedangkan Tabel 3.1 adalah rincian alokasi pemakaian *port-port* I/O.



Gambar 3.5. Minimum system ATmega8

Tabel 3.1. Alokasi *port I/O* pada *microcontroller*

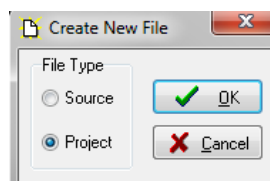
| Port | Alokasi |
|----------------|--|
| PC0-PC2 | Jalur ADC <i>channel</i> 0 sampai <i>channel</i> 2 untuk pembacaan sensor <i>array</i> TGS |
| PD0 dan PD1 | Jalur pengiriman data serial (RX dan TX) yang disalurkan ke PC |
| PD2,3 dan 4 | Jalur untuk mengontrol relay |
| PD5,6 dan PB 1 | Jalur <i>input</i> tombol |

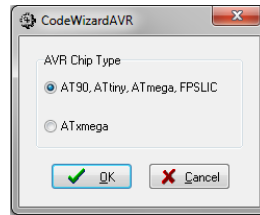
Pin VCC diberi masukan tegangan operasi berkisar antara 4,5 volt sampai dengan 5,5 volt. Pin RESET berfungsi untuk masukan reset program secara otomatis atau manual. Sedangkan pin MOSI, MISO, dan SCK digunakan untuk keperluan pemrograman *microcontroller*. Nilai kapasitor yang digunakan adalah 30 pF. Frekuensi kristal yang dipakai adalah 8 MHz.

a. *Setting software codevision AVR*

Sebelum menggunakan *software codevision AVR* sebagai *downloader*, pertama-tama harus melakukan penyetelan pada *software* ini. Berikut adalah langkah-langkahnya:

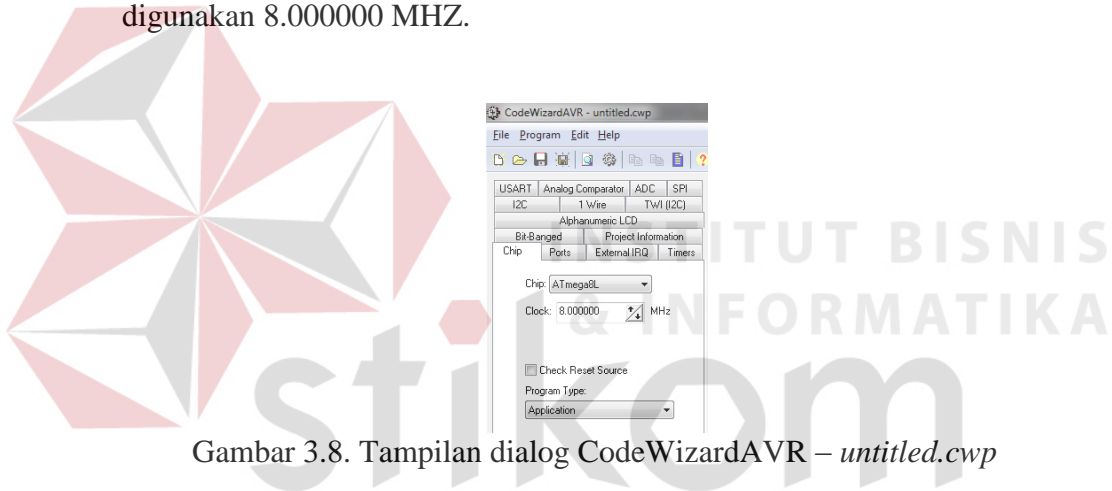
1. Pada tampilan awal *software* terdapat menu bar pada bagian atas.
2. Pilih *file* => *new*, selanjutnya akan muncul dialog *Create New File*, pilih *project* => OK. Dialog *Create New File* dapat dilihat pada Gambar 3.6. Kemudian akan muncul dialog *AVR Chip Type* dengan dua pilihan seperti yang terlihat pada Gambar 3.7, lalu pilih ATMega karena sesuai dengan jenis yang digunakan dalam tugas akhir ini.

Gambar 3.6. Tampilan dialog *Create New File*

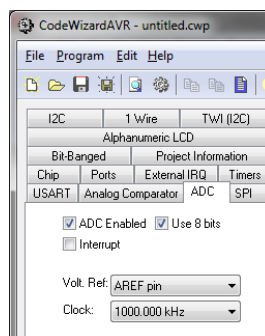


Gambar 3.7. Tampilan dialog AVR Chip Type

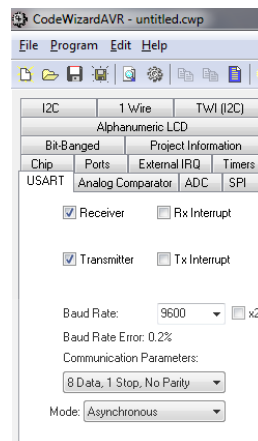
3. Kemudian tampak dialog CodeWizardAVR – *untitled.cwp*. Dialog CodeWizardAVR – *untitled.cwp* dapat dilihat pada Gambar 3.8.
4. Ubah bagian tab Chip, pilih seri *microcontroller* yang sesuai dengan yang digunakan, ATmega8L. Nilai Clock (komponen kristal) yang digunakan 8.000000 MHZ.

Gambar 3.8. Tampilan dialog CodeWizardAVR – *untitled.cwp*

5. Untuk mengaktifkan ADC pada tab ADC pilih *ADC Enabled* dan *Use 8 bits*. *Volt. Ref* => AREF pin. *ADC Clock* => 1000.00 KHz. Berikut adalah tampilan *setting* ADC pada Gambar 3.9.

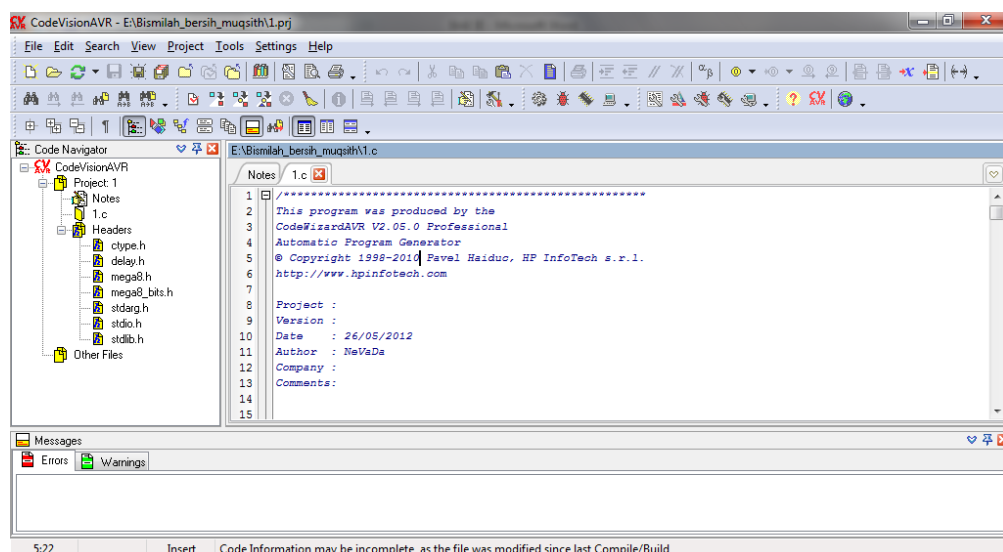
Gambar 3.9. Tampilan *setting* ADC

6. Karena menggunakan komunikasi serial maka buka tab USART, lalu pilih *receiver* dan *transmitter* kemudian *setting baudrate* 9600, komunikasi parameter 8 Data, 1 Stop, No Parity lalu mode *Asynchronous*. Berikut adalah tampilan *setting* ADC pada Gambar 3.10.



Gambar 3.10. Tampilan *setting* USART

7. Pada menu bar pilih Program, pilih *Generate, Save and exit*. Ketiga-tiganya simpan dengan nama yang sama.
8. Selanjutnya tampak kode program pada *software codevision AVR*. Dapat dilihat pada Gambar 3.11.

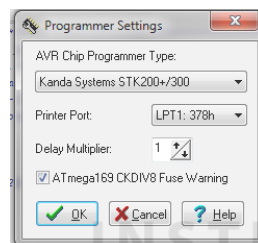


Gambar 3.11. Tampilan kode program

b. *Download* program dari komputer ke *microcontroller*

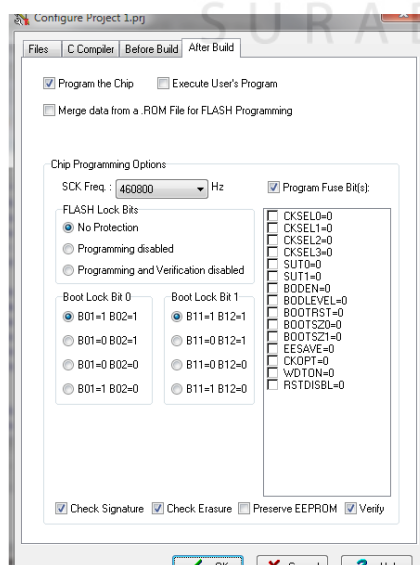
Sebelum *download* program dari komputer, lakukan *setting* pada *software* CVAVR dengan cara sebagai berikut:

1. Pilih menu *Setting* => *Programmer*.
2. Tampak kotak dialog *Programmer Setting*. Ubah tipe pada *AVR Chip Programmer Type* untuk *microcontroller* AVR ATmega8 (L) "Kanda System STK200+/300", kemudian OK, tampilan *Programmer Setting* dapat dilihat pada Gambar 3.12.



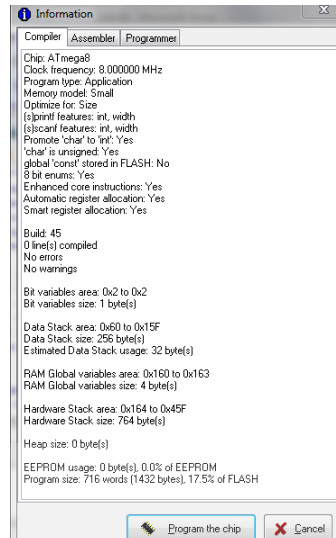
Gambar 3.12. *Setting* downloader Kanda System STK200+/300

3. Pilih menu *Project* => *Configure* => *Tab After Build* => pilih *Program the Chip* => OK. Seperti yang ditunjukkan pada Gambar 3.13.



Gambar 3.13. Tampilan *Dialog Configure Project*

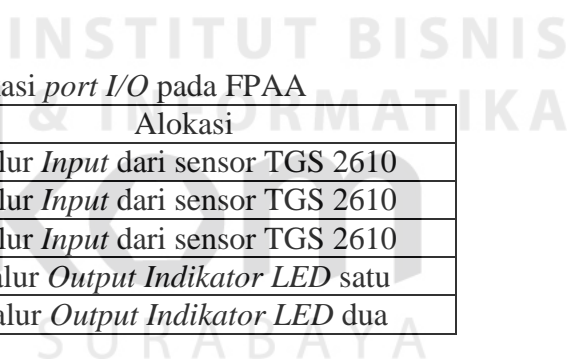
4. Pilih menu *Project => Build (Shift+F9)*, tampil *dialog Information =>* pilih Program, *dialog information* dapat dilihat pada Gambar 3.14.



Gambar 3.14. Tampilan *dialog Information*

3.2.3. Interface RS232

Interface RS232 merupakan suatu jembatan dalam metode komunikasi serial. Dalam perancangannya komponen yang digunakan adalah IC MAX232 dimana komponen pendukungnya lima buah kapasitor dengan nilai 10uF yang terhubung pada pin C1, C2, V+, V-. Penggunaan komponen ini dimaksudkan untuk komunikasi serial antara rangkaian demodulasi FSK dengan komputer. MAX232 ini akan mengubah level tegangan TTL data serial RX dan TX. Rangkaian Interface RS232 dapat dilihat pada Gambar 3.15.



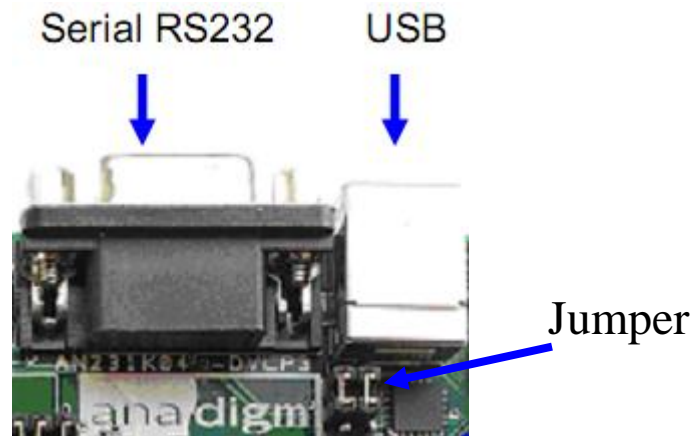
Tabel 3.2. Alokasi *port I/O* pada FPAA

| |
|---|
| <p>asi <i>port I/O</i> pada FPAA</p> <p>Alokasi</p> |
|---|

| |
|---------------------------------------|
| dur <i>Input</i> dari sensor TGS 2610 |
| dur <i>Input</i> dari sensor TGS 2610 |

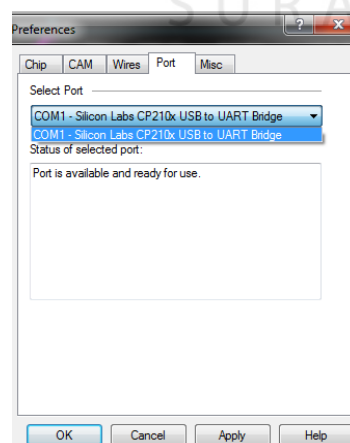
| |
|---|
| Nilai <i>Input</i> dari sensor TGS 2610 |
| Nilai <i>Input</i> dari sensor TGS 2610 |
| Nilai <i>Output Indikator LED</i> satu |
| Nilai <i>Output Indikator LED</i> dua |

tegangan minimal 4 volt sampai dengan

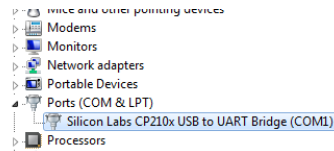


Gambar 3.16. Konfigurasi *Jumper* Serial RS232 dan USB

Selain merancang, program AnadigmDesigner[®]2 juga dapat disimulasikan sebelum di *download* kedalam *Hardware* FPAA. dengan cara pilih menu ***Simulate*** => ***Begin Simulation***, atau tekan ***F5***. Kemudian sebelum *download* program, pastikan FPAA sudah terhubung dengan *interface* PC/Laptop. Dan samakan *setting* COM Port pada AnadigmDesigner[®]2 dengan COM Port pada *device manager*. Untuk AnadigmDesigner[®]2 dengan cara pilih menu ***Setting*** => ***Preferences*** => ***Port***. Pada Gambar 3.17 tampilan pengaturan COM pada AnadigmDesigner[®]2 dan Gambar 3.18 *device manager* pada PC/Laptop.



Gambar 3.17. Tampilan *setting* COM Port pada AnadigmDesigner[®]2.



Gambar 3.18. Tampilan *setting COM Port* pada *device manager*.

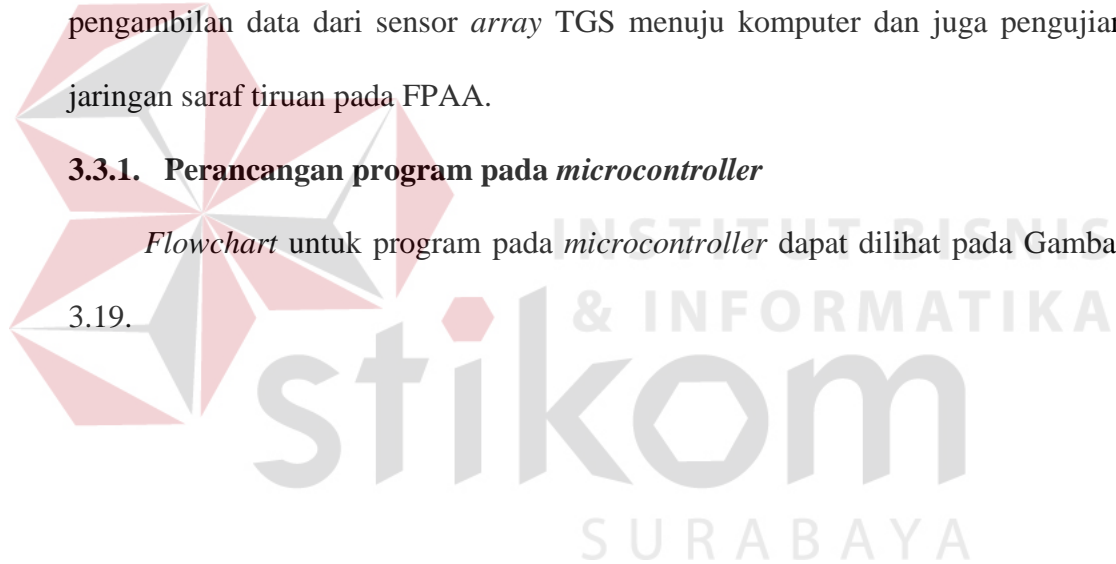
3.3. Perancangan perangkat lunak

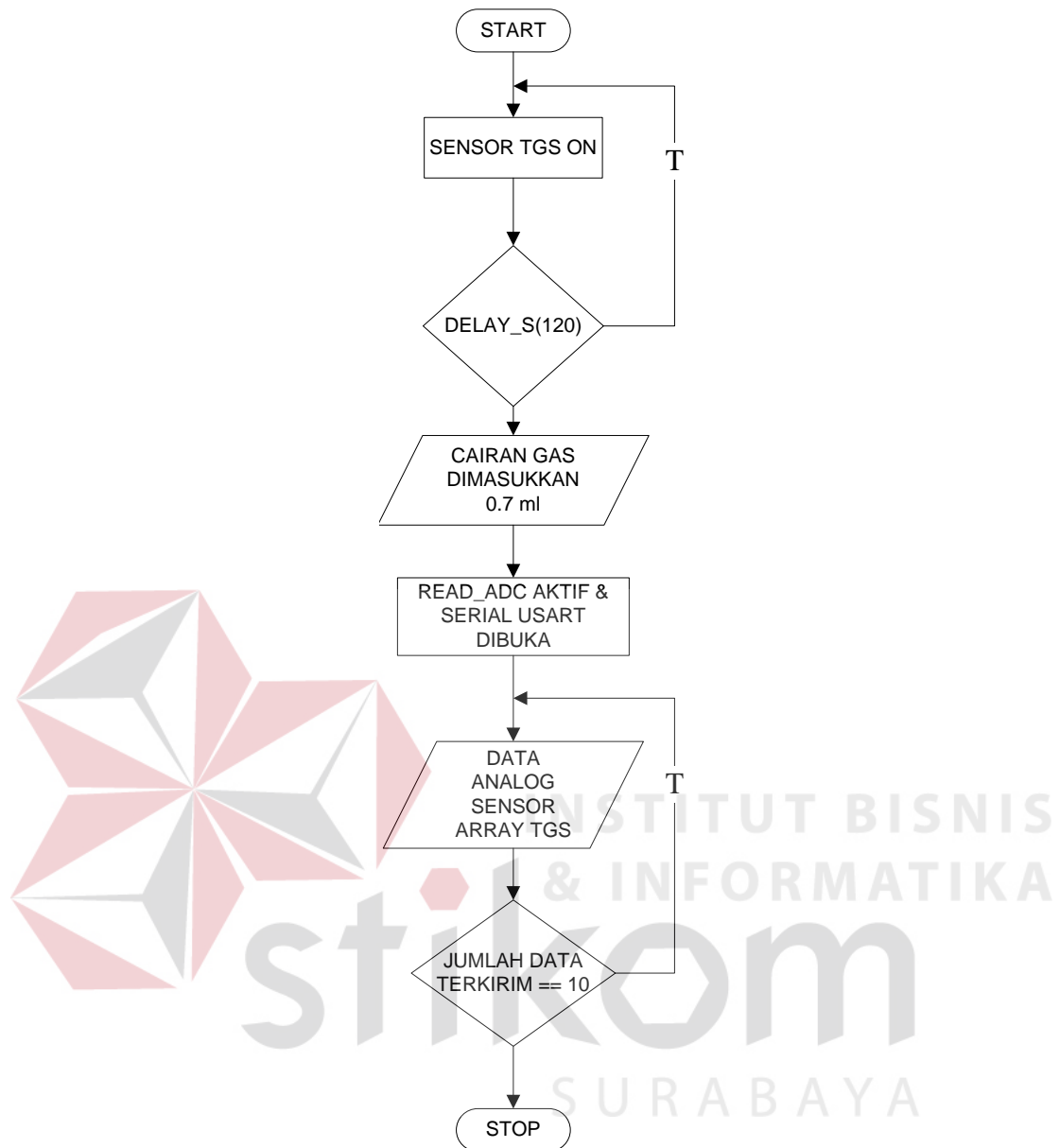
Selain *hardware* yang diperlukan pada perancangan dan pembuatan alat ini, juga diperlukan *software/* program pada *microcontroller*, komputer dan juga FPAA untuk dapat bekerja sesuai dengan fungsinya. Perancangan perangkat lunak pada *microcontroller* dirancang agar dapat mengatur proses pengambilan data dari sensor *array* TGS menuju komputer dan juga pengujian jaringan saraf tiruan pada FPAA.

3.3.1. Perancangan program pada *microcontroller*

Flowchart untuk program pada *microcontroller* dapat dilihat pada Gambar

3.19.





Gambar 3.19. Flowchart program pada *microcontroller*

Adapun penjelasan dari bagan alir di atas adalah sebagai berikut:

1. Proses awal dimulai dengan mengaktifkan sensor *array* TGS selama 120 detik untuk proses pemanasan sensor.
2. Setelah sensor sudah panas, gas dimasukan pada tabung yang telah disediakan, kemudian program akan mengaktifkan fungsi ADC pada *microcontroller* dan proses pengambilan data tegangan *analog* dimulai.

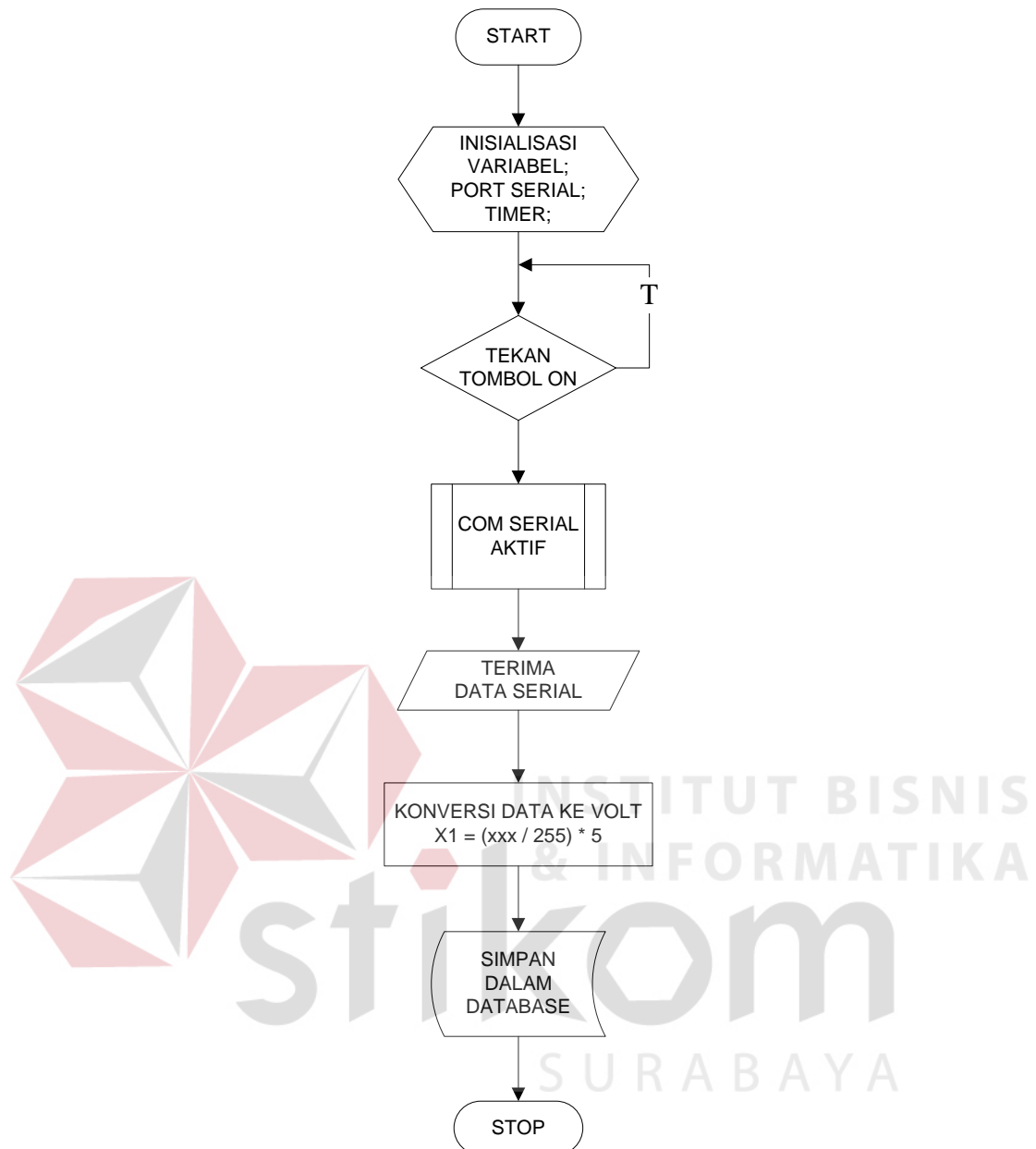
3. Data dari sensor *array* TGS diambil sebanyak 10 data berturut-turut kemudian dikirim secara serial menuju PC/laptop.

3.3.2. Perancangan program *Visual Basic 6* pada komputer

Dalam tugas akhir ini dirancang dua buah program visual, yang pertama program dirancang agar dapat menerima data yang telah dikirim secara serial oleh *microcontroller* kemudian disimpan ke dalam *database*. Program kedua merupakan pelatihan jaringan saraf tiruan yang menggunakan metode *perceptron*. Program dirancang agar dapat mengenali pola sesuai masukan sensor dan keluaran yang telah ditentukan sebelumnya sehingga menghasilkan nilai bobot dan bias yang sesuai dan selanjutan nilai-nilai tersebut digunakan pada FPAA.

- a. Program pengambilan dan penyimpanan data ke dalam *database*

Seperti yang sudah dijelaskan sebelumnya, pada sub ini adalah program pertama yang difungsikan untuk mengambil data. dan pada Gambar 3.20 adalah *flowchart* dari isi program.



Gambar 3.20. *Flowchart* pengambilan dan penyimpanan data ke dalam *database*

Proses awal adalah proses inialisasi PORT serial pada komputer sebagai *input*, tujuannya agar *microcontroller* dapat berkomunikasi dengan baik dengan komputer. Selanjutnya apabila ada penekanan tombol ON maka yang dilakukan pertama kali adalah mengaktifkan jalur COM serial, terima data dari sensor, data bilangan bulat 0 – 255 *dikonversi* menjadi 0 – 5 seperti

tegangan ADC dalam format tipe data *string*, kemudian data disimpan kedalam *database* untuk selanjutnya digunakan dalam pelatihan jaringan saraf tiruan.

Berikut ini adalah langkah pembuatan program pada komputer dengan menggunakan program *visual basic 6*:

i. Inisialisasi komunikasi serial

Pada saat melakukan inisialisasi komunikasi serial di pemrograman visual khususnya program VB ada komponen yang harus digunakan adalah komponen *Microsoft comm control 6.0*. lebih jelasnya lihat Gambar 3.21 perlu diketahui bahwa komponen ini masih tersimpan di dalam data base komponen program VB. Untuk memunculkannya yang harus dilakukan adalah:

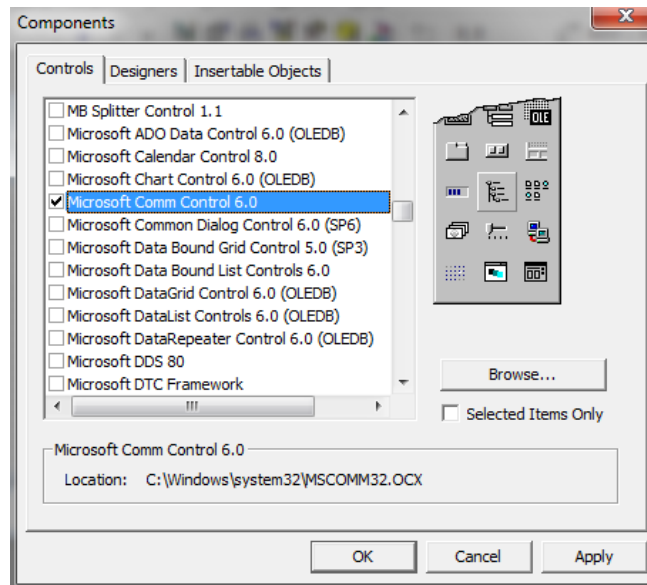
1. Pilih kanan pada *Toolbox general* (lihat Gambar 3.22), pilih *component* kemudian akan muncul *dialog components* lihat Gambar 3.23. pada *tab control* pilih *Microsoft comm control 6.0*. Kemudian pilih *apply* dan *close*.



Gambar 3.21. Komponen *Microsoft comm control 6.0*.



Gambar 3.22. *Toolbox general*



Gambar 3.23. Dialog components Microsoft comm control 6.0.

2. Setelah itu komponen *Microsoft comm control 6.0*. ini akan muncul pada *Toolbox general* lihat Gambar 3.24.



Gambar 3.24. Komponen *MSComm* muncul pada *toolbox general*

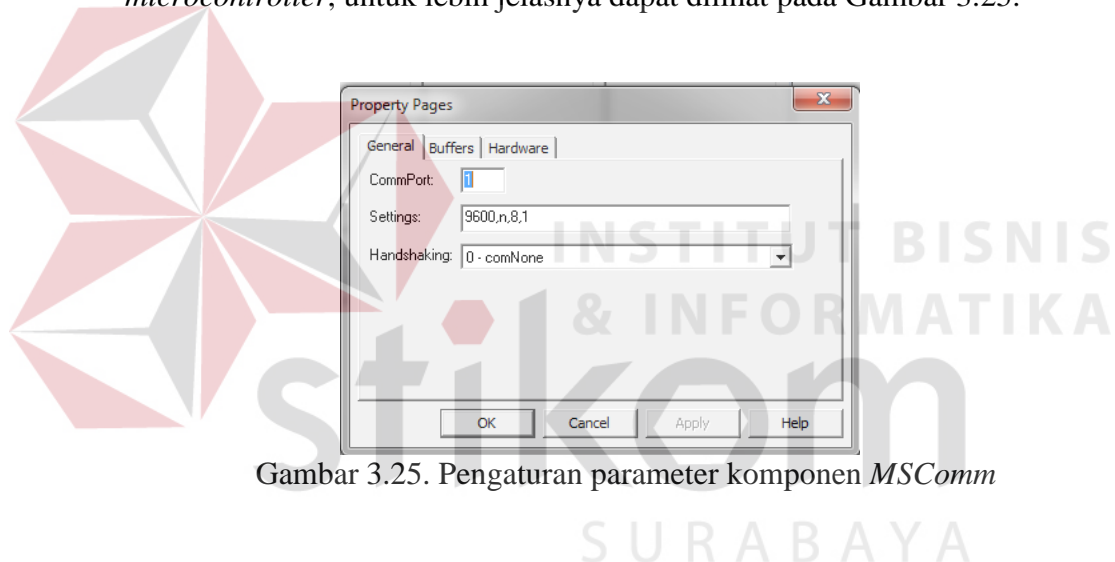
3. Penulisan intruksi ke dalam program

```

.....
MSComm1.PortOpen = True
.....

```

Maksudnya komponen *MSComm* telah aktif dan siap melakukan komunikasi dengan aplikasi dari luar. Selanjutnya pengaturan parameter dari *MSComm*, dalam hal pengaturan parameter harus mengerti beberapa hal yang harus disesuaikan dengan *hardware* yang digunakan misalnya kebutuhan akan nilai dari *baudrate* yang digunakan, jumlah *bit* dalam satu paket data, *parity check* dan *stop bit* yang digunakan. Parameter serial yang digunakan adalah *baudrate* 9600bps, 8 *bit* data, *no parity check* dan menggunakan 1 *stop bit* yang telah disesuaikan dengan modul *microcontroller*, untuk lebih jelasnya dapat dilihat pada Gambar 3.25.



Gambar 3.25. Pengaturan parameter komponen *MSComm*

4. Pengaturan pada proses penerimaan data menggunakan *MSComm* berikut adalah potongan programnya.

```

.....
data = MSComm1.Input
.....

```

Maksud dari potongan program tersebut adalah jalur *MSComm* pada jalur *input* mempersilahkan data yang masuk untuk diterima kemudian disimpan pada variabel data.

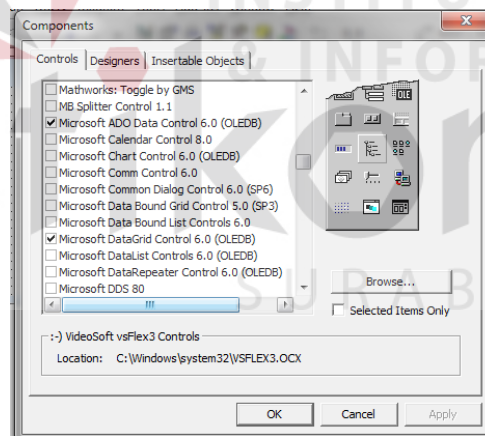
ii. Membuat *Database*

Pada saat melakukan inisialisasi komunikasi serial di pemrograman *visual* khususnya program VB ada komponen yang harus digunakan adalah komponen *Adodc* dan *DataGrid* untuk lebih jelasnya lihat Gambar 3.26 perlu diketahui bahwa komponen ini masih tersimpan di dalam *database* komponen program VB maka dari itu untuk memunculkannya dilakukan cara yang sama seperti pada komponen *Microsoft comm control 6.0*.



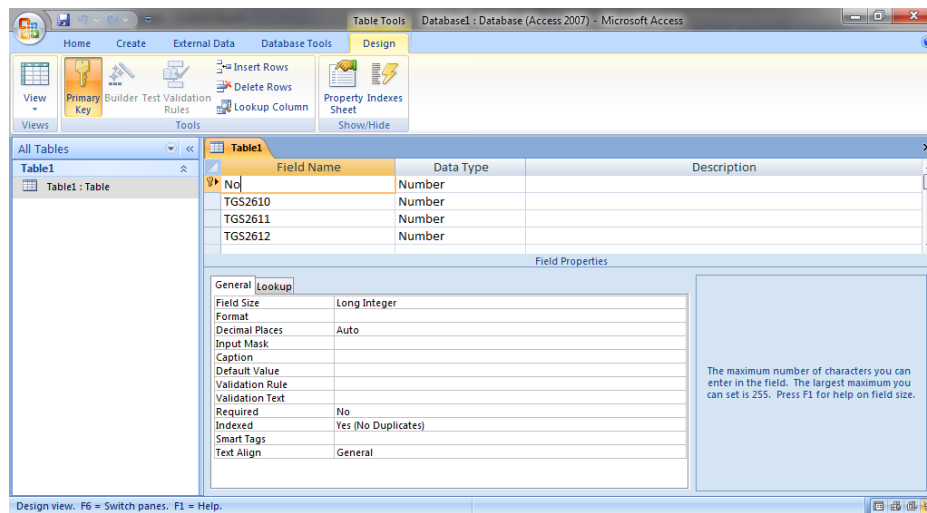
Gambar 3.26. Komponen *Adodc* dan *DataGrid*

Gambar 3.27. Adalah daftar komponen yang dipilih.

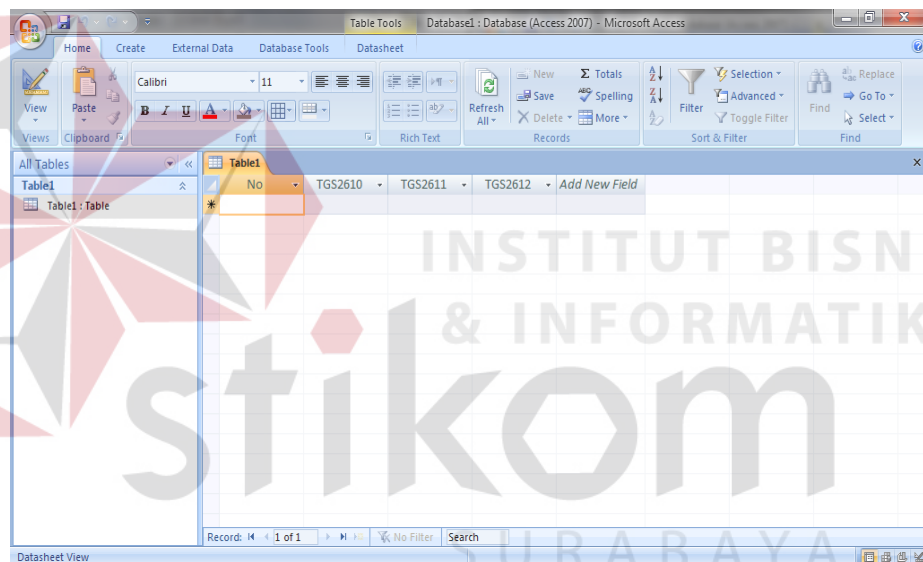


Gambar 3.27. Dialog components *Microsoft ADO Data* dan *Data Bound Grid*

Langkah selanjutnya yaitu membuat *database* dalam bentuk tabel, dan dalam membuat tampilan tabel dengan menggunakan fasilitas dari *Microsoft Office Access* yang di-load ke dalam program, untuk proses pembuatannya adalah pertama merancang *design view*, untuk lebih jelasnya dapat dilihat pada Gambar 3.28 dan Gambar 3.29 merupakan *datasheet view*.

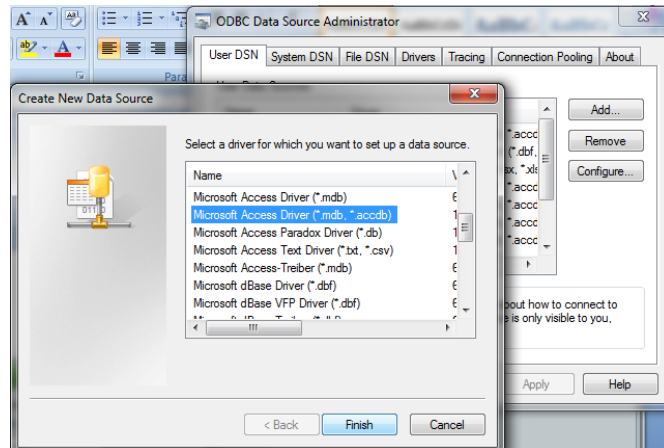


Gambar 3.28. Design view



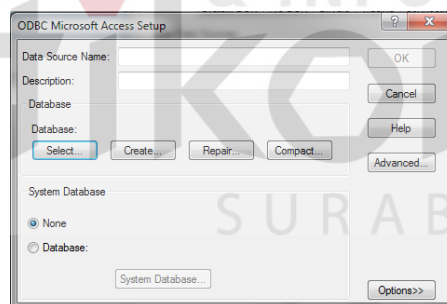
Gambar 3.29. Datasheet view

file.accdb dari Microsoft office access disimpan dengan nama *Database1*, file dipindahkan ke folder program, pada control panel => Administrative Tools => Data Sources (ODBC) => Add => Microsoft Access Driver (*.mdb, *.accdb) => Finish. Hasil akhirnya terlihat pada Gambar 3.30.

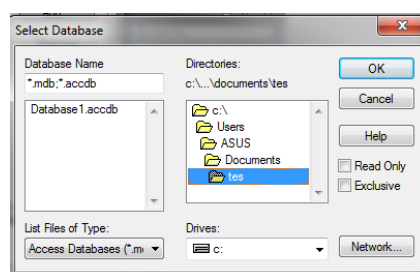


Gambar 3.30. Setting Microsoft Access Driver pada Control Panel

Setelah selesai akan muncul jendela *ODBC Microsoft Access Setup* seperti pada Gambar 3.31, pilih *Select* untuk mencari file *.accdb* yang telah dibuat dan disimpan pada *directori* tertentu. Pilih file tersebut lalu pilih OK, untuk lebih jelasnya dapat dilihat pada Gambar 3.32 pada jendela sebelumnya pilih OK pada menu bar.

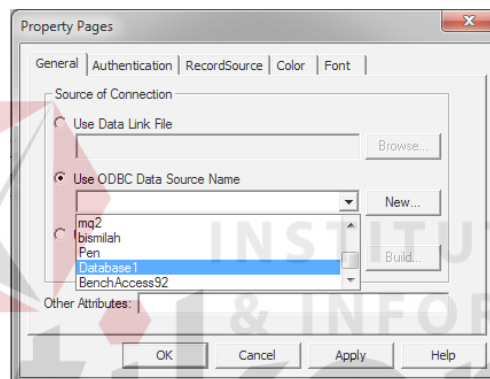


Gambar 3.31. Tampilan ODBC Microsoft Access Setup

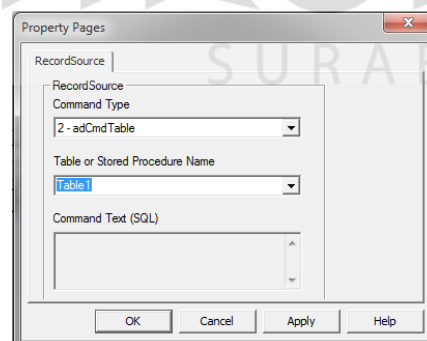


Gambar 3.32. Select Database

Setelah pengaturan pada *control panel*, dilanjutkan pada program *visual basic 6* yaitu pada komponen *Adodc1*, dengan menekan tombol kanan pada *mouse* => *ADODC properties*. Pilih *Use ODBC Data Source Name* dan pilih nama *file* yang telah dibuat. Pada Gambar 3.33 dipilih karena *database1* tersebut yang dibuat oleh penulis, lalu tekan tombol OK. Selanjutnya pilih *Tab RecordSource* pada *properties Adodc* dan akan membuka jendela seperti pada Gambar 3.34, pada *command type* dipilih *2-adCmd Table* pada *Table or Stored Procedure Name* dipilih *table1*.



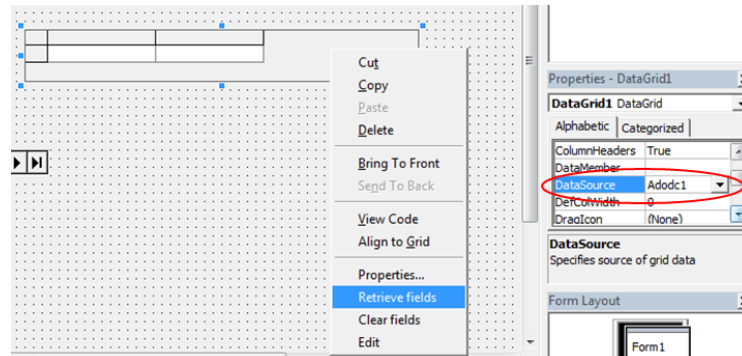
Gambar 3.33. ADODC Property Pages



Gambar 3.34. Record Source ADODC

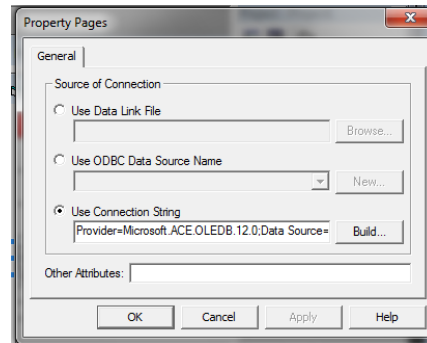
Langkah selanjutnya setelah pengaturan ADODC adalah pengaturan pada *DataGrid*. Pada *DataSource* yang terdapat pada *propeties* dari *DataGrid* dipilih *Adodc1* => lalu pilih kanan pada komponen *DataGrid1* => pilih *Retrieve fields*.

Untuk lebih jelasnya terlihat pada Gambar 3.35, saat proses berhasil tabel pada *DataGrid1* akan sesuai dengan tabel pada *access*.

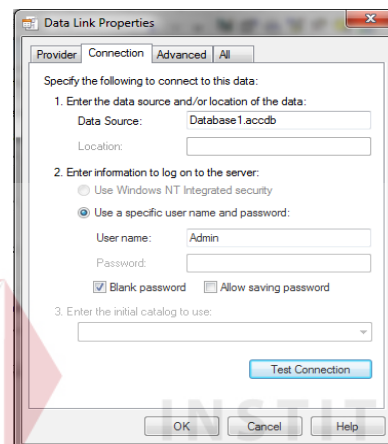


Gambar 3.35. Pengaturan *Adodc1* dan *Retrive fields*

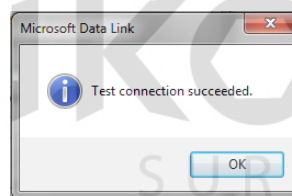
Pengaturan yang telah dilakukan sampai saat ini agar *database* yang dibuat pada *Access* dapat terkoneksi dengan program di *Visual Basic 6*, tetapi jika direktori *file* pada kedua program ini berubah maka koneksinya tidak dapat bekerja sebagai mana mestinya. Untuk itu dilakukan pengaturan *connection* yang diawali pada komponen *Adodc1*. Pilihlah *connection string* pada *properties Adodc* lalu pilih *Use Connection String => pilih Build* seperti pada Gambar 3.36 sehingga muncul *Data Link Properties* dan pada *tab provider* pilih *Microsoft Office 12.0 Access* lalu *next*, setelah itu pada *Data Source* diisi dengan nama *file access* yang telah dirancang seperti pada Gambar 3.37 setelah itu *pilih Test Connections*. Jika sudah berhasil terkoneksi akan muncul seperti pada Gambar 3.38 program siap digunakan.



Gambar 3.36. Use Connection String



Gambar 3.37. Pengaturan Data Link Properties



Gambar 3.38. Koneksi data sukses

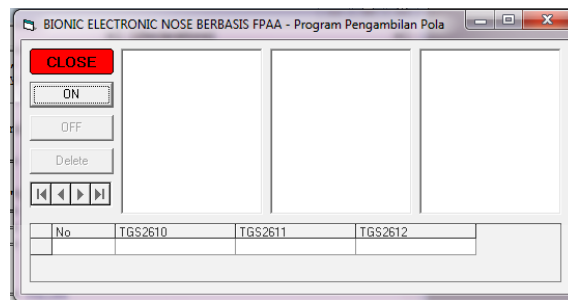
Saat semua pengaturan sudah dilakukan hingga berhasil, tabel dapat diisi sesuai dengan format dan rancangan yang sesuai, dengan perintah di dalam kode program:

```
Adodc1.Recordset.AddNew
    Adodc1.Recordset.Fields("No") = no
    Adodc1.Recordset.Fields("TGS2610") = x1
    Adodc1.Recordset.Fields("TGS2611") = y1
    Adodc1.Recordset.Fields("TGS2612") = z1
Adodc1.Recordset.Update
```

Penjelasan program adalah pada data *fields no* diisi sesuai dengan no urut, kemudian pada *fields* TGS2610 sampai *fields* TGS2612 di isi dengan data yang dikirim oleh sensor *array* TGS.

iii. Tampilan Program

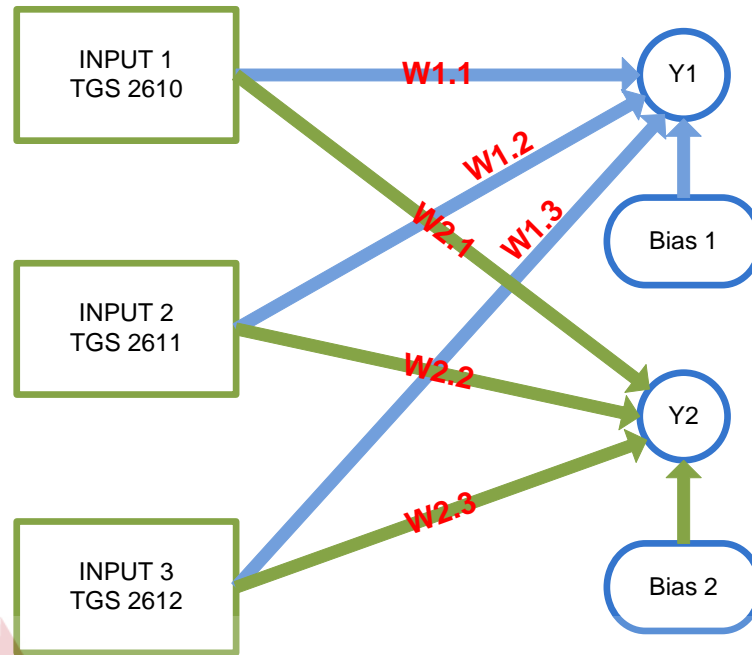
Tampilan program Pengambilan dan *database* terlihat pada Gambar 3.39.



Gambar 3.39. Program pengambilan dan *database*

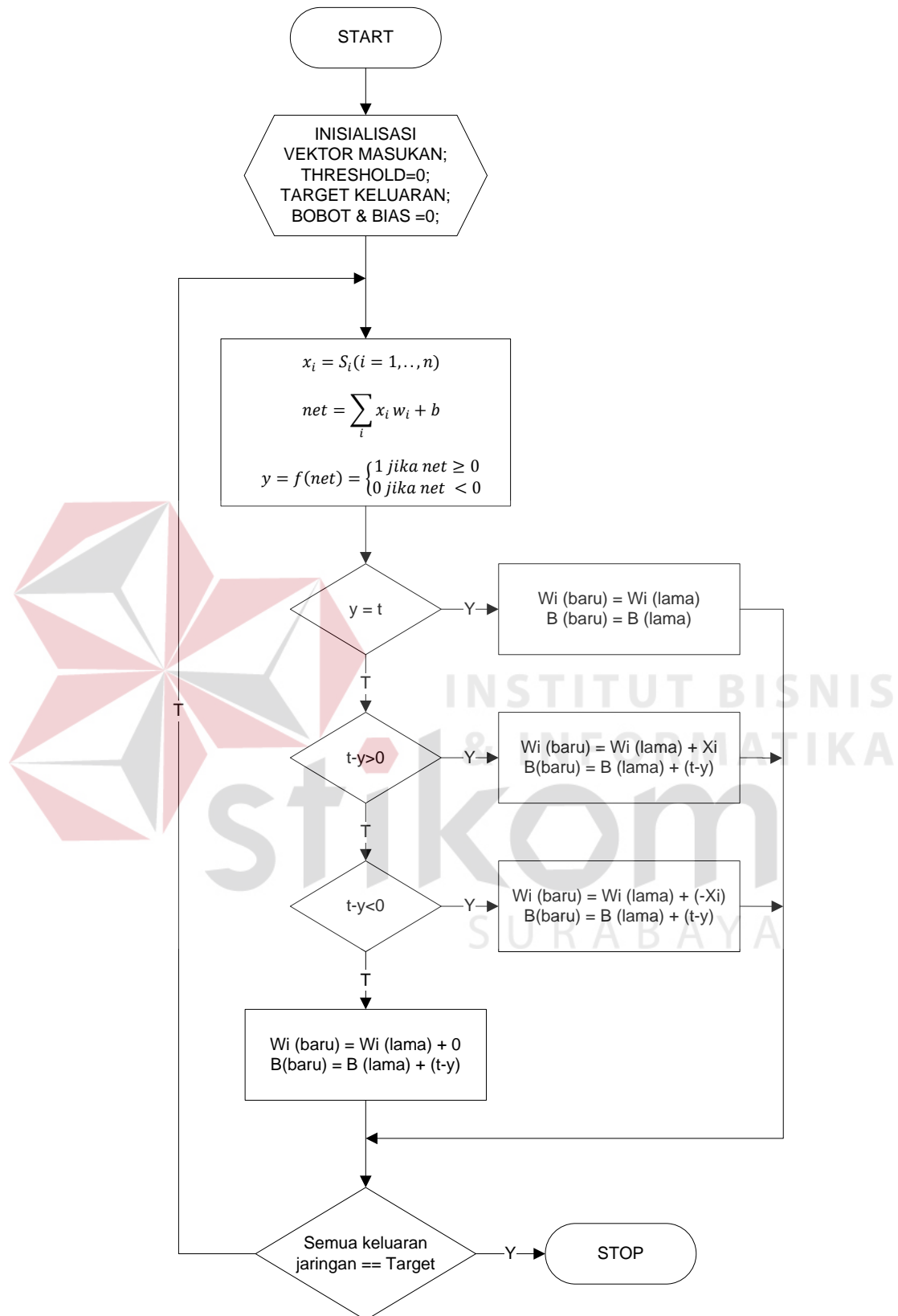
b. Program pelatihan jaringan saraf tiruan menggunakan metode *perceptron*

Data-data sensor *array* TGS yang telah diperoleh akan diproses dalam suatu program yang dapat mengenali jenis gas tertentu sesuai dengan polanya. Pola tersebut diolah dengan menggunakan salah satu metode jaringan saraf tiruan yaitu *perceptron*. Gambaran umum dari *neuron layer* yang dirancang dapat dilihat pada Gambar 3.40.



Gambar 3.40. Rancangan *Neuron Layer*

Pada Gambar 3.40 dapat dilihat bahwa rancangan neuron layer terdiri dari tiga unit masukan dan buah bias serta dua buah unit keluaran (Y1 dan Y2). Tiga buah nilai *inputan* berasal dari sensor *array* TGS yang telah disimpan pada program sebelumnya sedangkan dua buah *output* memiliki nilai yang nantinya telah ditentukan sebelumnya untuk dijadikan target. Gambar 3.41 merupakan *flowchart* dari isi program pelatihan.



Gambar 3.41. Flowchart pelatihan perceptron

Adapun penjelasan dari Gambar 3.41 adalah sebagai berikut:

1. Proses awal dimulai dengan inisialisasi semua variabel pendukung, yaitu nilai *input* yang berasal dari program pengambilan *database* dan nilai target yang telah ditentukan sebelumnya yang dipakai berbentuk bebas (tidak harus biner/bipolar), *threshold*, bias dan bobot yang memiliki nilai nol pada awal proses.
2. Selanjutnya set aktivasi unit masukan $x_i = s_i$ ($i=1,...,n$) dan hitung respon unit keluaran dengan rumus (3.5).

$$net = \sum_i x_i w_i + b \quad (3.5)$$

Dan fungsi aktivasi yang memiliki output biner seperti pada persamaan (3.6).

$$y = f(net) = \begin{cases} 1 & \text{jika } net \geq 0 \\ 0 & \text{jika } net < 0 \end{cases} \quad (3.6)$$

3. Kemudian perubahan bobot diubah berdasarkan *error* yang terbentuk dari selisih antara target yang diinginkan dengan keluaran jaringan ($=f(net)$). Perubahan bobot bukan merupakan hasil kali antara target dengan masukan. Misalkan $p = (x_1 \ x_2 \ ... \ x_n)^t$ adalah masukan yang diberikan ke jaringan dengan target = t , dan $a = f(net)$. Perubahan bobot yang dilakukan adalah :

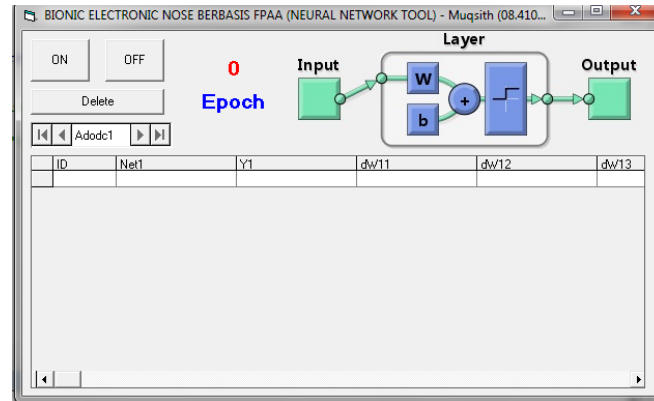
$$\Delta w = \begin{cases} P & \text{jika } (t - a) > 0 \\ -P & \text{jika } (t - a) < 0 \\ P & \text{jika } (t - a) = 0 \end{cases} \quad (3.7)$$

$$w_{baru} = w_{lama} + \Delta w \quad (3.8)$$

$$b_{baru} = b_{lama} + (t-a) \quad (3.9)$$

Perubahan bobot hanya dilakukan jika target \neq keluaran jaringan. Setelah proses pelatihan selesai, nilai bobot dan bias yang dihasilkan, diambil untuk kemudian diimplementasikan ke dalam FPAA. Kode program dapat dilihat pada

halaman lampiran, sedangkan tampilan program terlihat seperti pada Gambar 3.42.



Gambar 3.42. Tampilan Program Pelatihan JST dengan metode *perceptron*

3.3.3. Rancangan Neuron Network pada FPAA

Setelah program pelatihan menghasilkan nilai bobot dan bias yang sesuai, langkah selanjutnya adalah merancang jaringan saraf tiruan menggunakan *software* AnadigmDesigner[®]2 yang nantinya program akan di *download* pada FPAA. Terdapat *IO cell* yang dapat difungsikan sebagai *input* dan *output* jaringan dan setiap *IO cell* memiliki dua pin yaitu pin negatif dan positif. Sesuai dengan pengaturan yang dilakukan, dengan cara *double pilih* pada *IO cell* yang dimaksud, lalu lakukan *konfigurasi*. Dan pada *IO cell* terdapat dua pengkondisi sinyal yaitu *differential* dan *common-mode* yang memiliki rumus seperti pada persamaan (3.10) dan persamaan (3.11).

$$V_{Differential} = V_{Positive} - V_{Negative} \quad (3.10)$$

$$V_{CM} = \frac{V_{Positive} + V_{Negative}}{2} \quad (3.11)$$

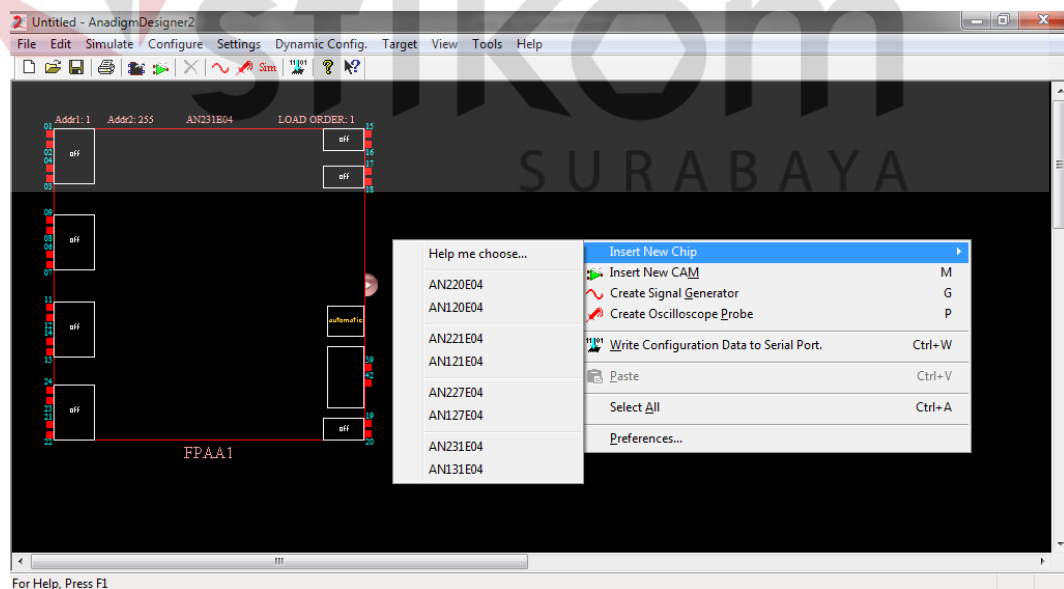
Terdapat beberapa mode yang dapat dipilih pada *IO Cell*, dan pada Tugas Akhir ini penulis menggunakan mode *bypass*, dimana sinyal *input* disalurkan

langsung melalui sel, melewati semua elemen sirkuit aktif. Sedangkan sinyal *output* dialihkan langsung melalui sel, melewati semua elemen sirkuit aktif. Pin *output* di dorong langsung oleh CAM yang memberikan masukan kepada *output cell*. Pada Gambar 3.43 merupakan *IO cell* yang menggunakan mode *bypass*.



Gambar 3.43. *IO cell* dengan *bypass mode*

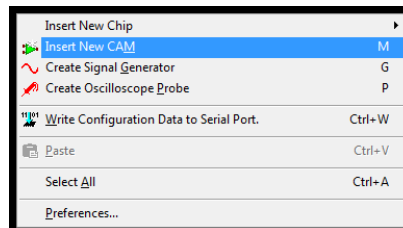
Perancangan jaringan saraf tiruan yang dimulai dari pemilihan *chip* FPAA yang terdapat pada program AnadigmDesigner[®]2, *chip* FPAA yang dipilih adalah AN231K04. Karena disesuaikan dengan *hardware* yang akan digunakan. Untuk memilihnya dengan cara pilih tombol kanan *mouse* pada halaman *project* yang berwarna hitam, kemudian *Insert new chip* => AN231E04. Untuk lebih jelasnya terlihat pada Gambar 3.44.



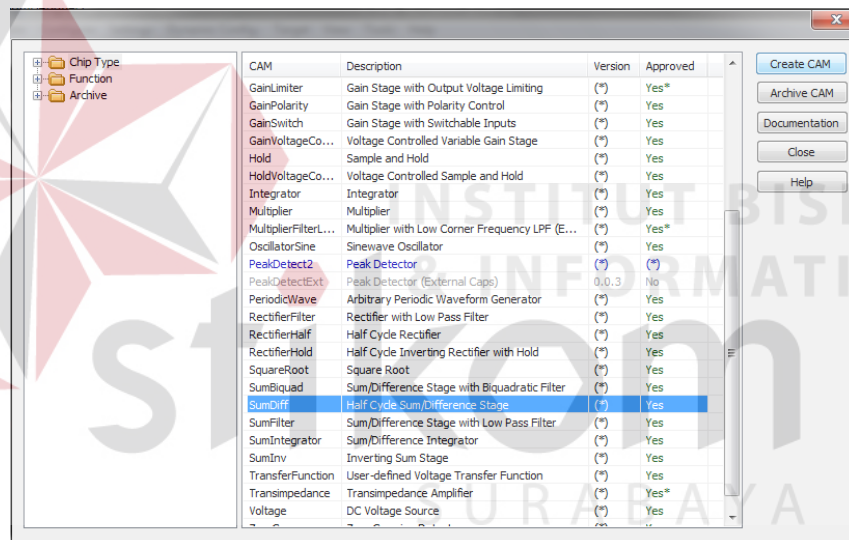
Gambar 3.44. Pengaturan *chip* FPAA

1. Perancangan komponen-komponen jaringan saraf tiruan, dengan pilih kanan => pilih *Insert New CAM*, terlihat seperti pada Gambar 3.45 kemudian

muncul jendela baru seperti pada Gambar 3.46. Pilih => *sumdiff* => *Create CAM* dan letakkan pada FPAA1, setelah itu akan muncul *Set CAM Parameter* dari komponen yang dipilih atau bisa dengan melakukan *double pilih* pada komponen yang dimaksud.



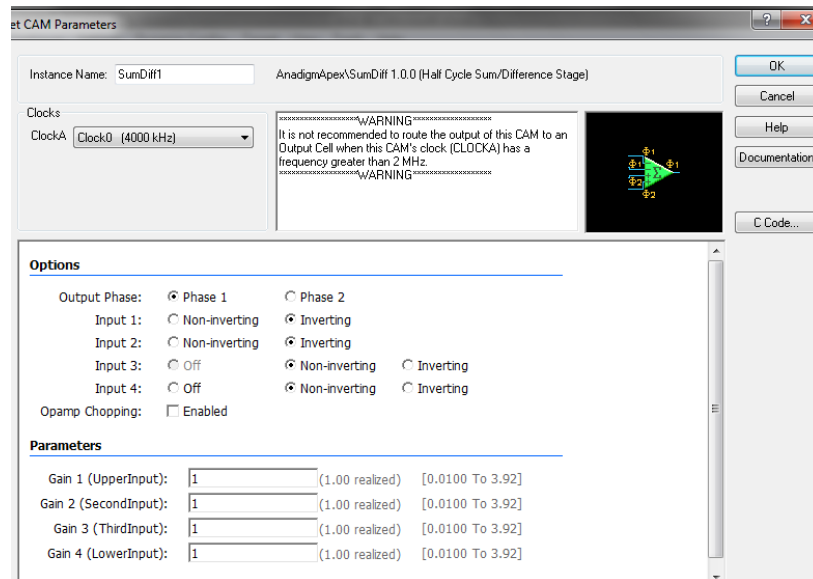
Gambar 3.45. *Insert New CAM*



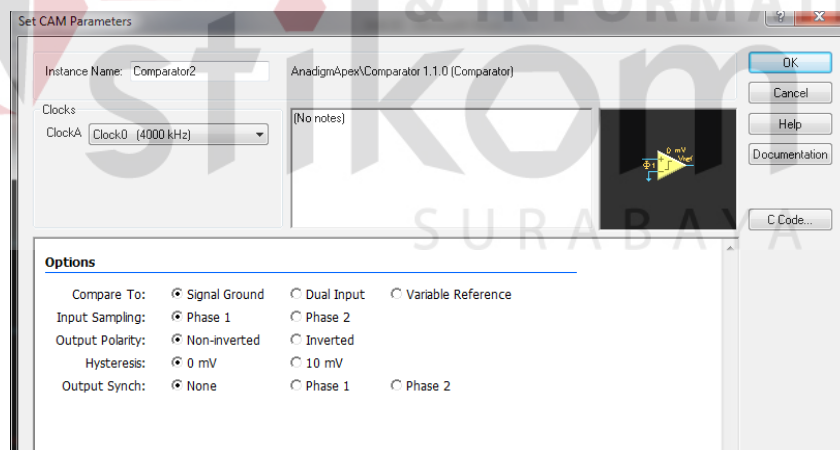
Gambar 3.46. Pemilihan komponen *SumDiff*

Pada pengaturan *sumdiff*, ada beberapa pilihan dan parameter yang dapat diubah-ubah sesuai dengan kebutuhan, seperti yang terlihat pada Gambar 3.47. Pada Gambar 3.47 terdapat dua *input inverting* dan *non-inverting* serta memiliki *Gain=1* di setiap *input*nya. Gain pada *sumdiff* yang terdapat pada chip AN231E04 memiliki *range* 0,01 – 3,92. dan *sumdiff* memiliki rumus sesuai dengan persamaan (3.12).

$$V_{out} = \pm G_1 V_{Input1} \pm G_2 V_{Input2} \pm G_3 V_{Input3} \pm G_4 V_{Input4} \quad (3.12)$$

Gambar 3.47. Pengaturan *Sumdiff1*

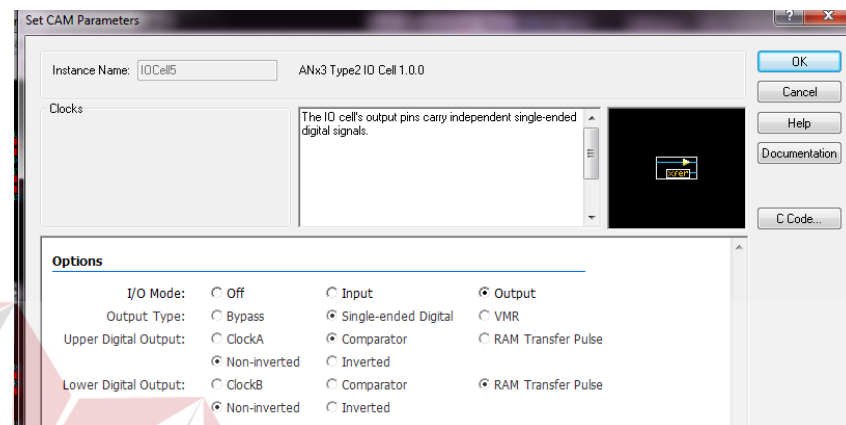
2. Hasil penjumlahan dari *sumdiff1* akan diteruskan menuju komponen komparator yang merupakan persamaan dari fungsi aktivasi, yang juga memiliki pilihan konfigurasi seperti yang terlihat pada Gambar 3.48.

Gambar 3.48. Beberapa konfigurasi pada *comparator*

3. Komponen berikutnya yang mendukung pembuatan jaringan saraf tiruan adalah *Voltage*, dimana cara pengambilan komponennya sama dengan *sumdiff1* dan opsi pengaturannya hanya *polarity* tegangan positif dan negatif

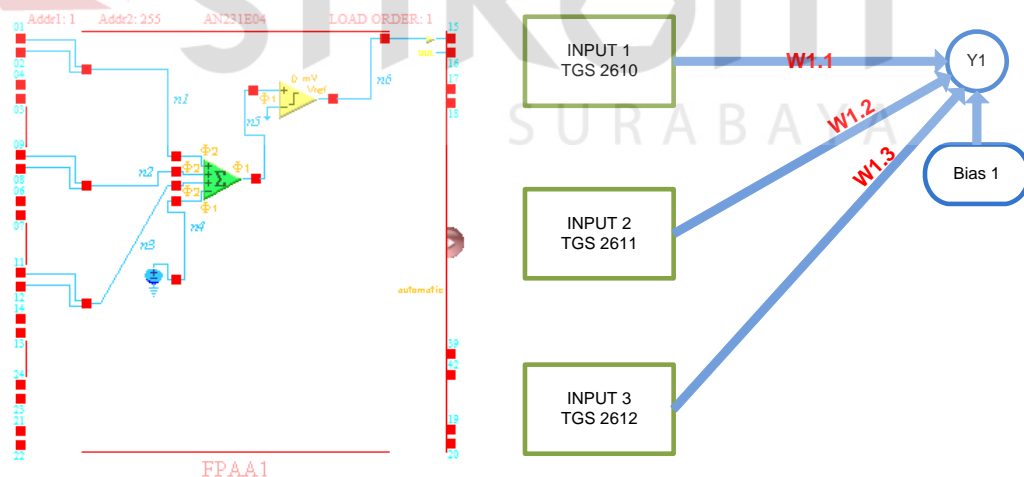
dengan besar tegangan 2 volt. *Voltage* diimplementasikan sebagai bias yang dikombinasikan dengan *sumdiff1*.

4. Pada *output cell* tipe *output* yang digunakan adalah *single-ended digital* dengan konfigurasi seperti pada Gambar 3.49.



Gambar 3.49. Konfigurasi *single-ended digital* pada *output cell*

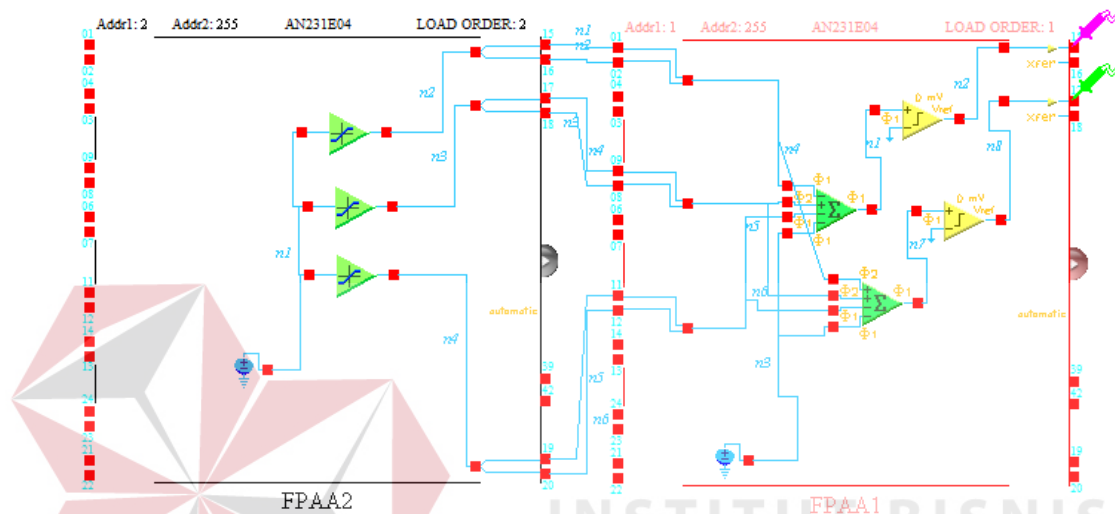
Untuk lebih jelasnya bagaimana *ilustrasi* dari rancangan jaringan saraf tiruan dapat dilihat pada Gambar 3.50.



Gambar 3.50. *Ilustrasi* jaringan saraf tiruan

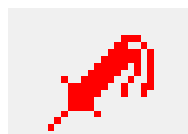
Setelah membahas tentang perancangan jaringan saraf tiruan pada FPAA, maka tahap selanjutnya yaitu simulasi. Simulasi pada AnadigmDesigner[®]2 untuk

mengetahui perancangan yang dilakukan benar atau tidak. Setelah simulasi berhasil maka program di *download* kedalam *Hardware*. Hal ini dapat dilakukan agar mengetahui apakah rangkaian dapat beroperasi sebagaimana mestinya. Pada Gambar 3.50 merupakan contoh rancangan yang digunakan dalam simulasi.



Gambar 3.51. Contoh rangkaian JST untuk disimulasikan

Pada *software* AnadigmDesigner[®]2, simulasi dapat dilakukan dengan menggunakan *oscilloscope probe* yang diletakkan pada titik tertentu, kemudian akan mengukur serta menampilkan hasil dari suatu sinyal seperti pada *oscilloscope*. Tetapi dalam *software* AnadigmDesigner[®]2 ini, *probe* yang bisa digunakan maksimal ada empat buah saja. Cara mengambilnya adalah dengan pilih kanan => kemudian pilih *create oscilloscope probe* atau pilih *icon* seperti pada Gambar 3.52 letakkan pada titik yang akan diukur.



Gambar 3.52. *Oscilloscope probe*

Kemudian pilih *icon begin simulation* seperti pada Gambar 3.53 dan akan muncul hasil pengukuran pada Gambar 3.54.

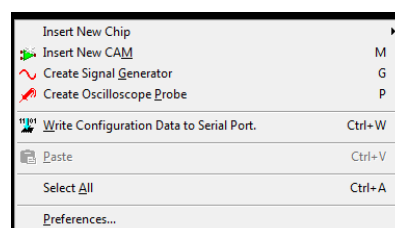


Gambar 3.53. *Begin simulation*



Gambar 3.54. *Oscilloscope*

Setelah semua rancangan jaringan saraf tiruan sudah sesuai dengan hasil simulasi yang diharapkan, lakukan *download* program dengan cara pilih kanan => *write Configurations data to serial port* atau dengan *Ctrl + W*, seperti yang terlihat pada Gambar 3.55.

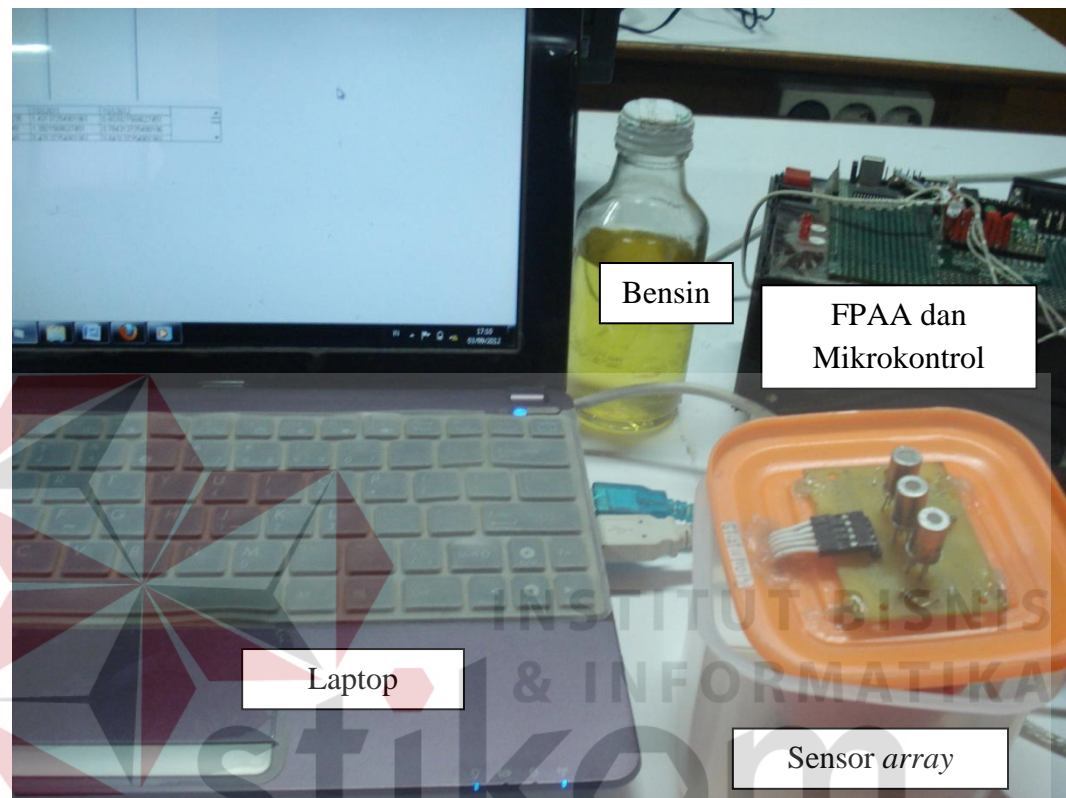


Gambar 3.55. *Download* program dari AnadigmDesigner[®]2 ke FPAA

Saat proses *download* akan terdengar suara “ting-ting” dua kali yang menandakan *download* program berhasil dan siap digunakan.

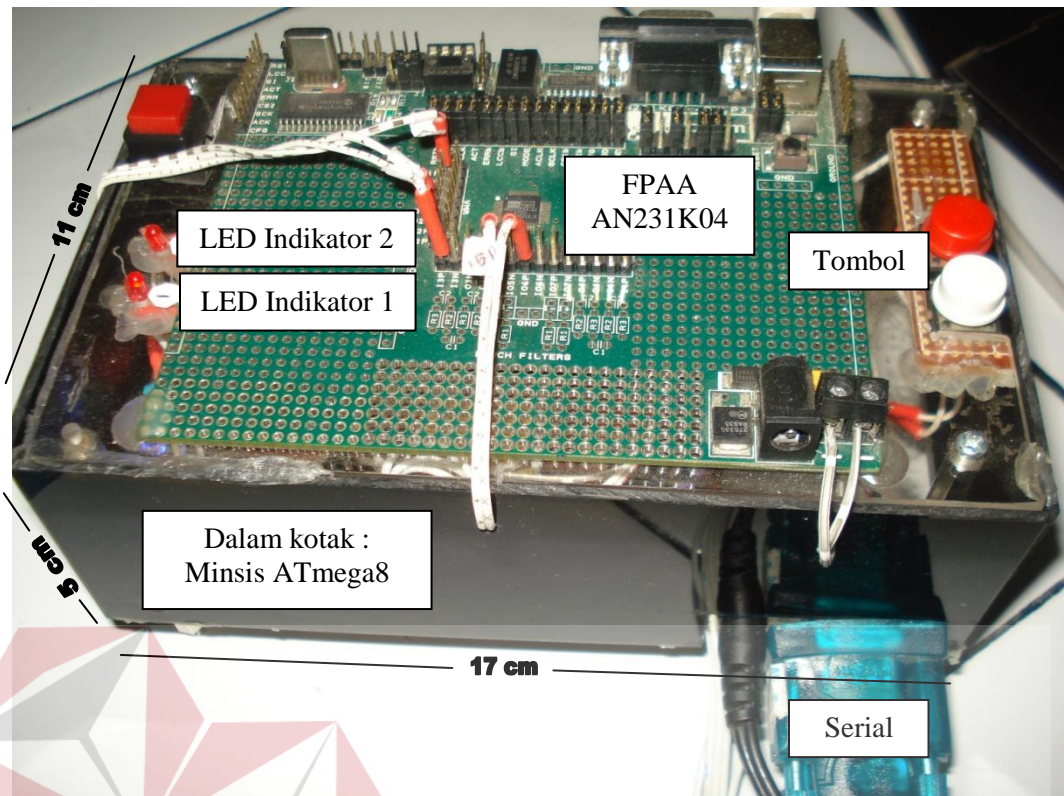
3.4. Perancangan model sistem

Rancangan model sistem yang dibuat dapat dilihat pada Gambar 3.55 berupa foto sensor *array* beserta sistem secara keseluruhan.



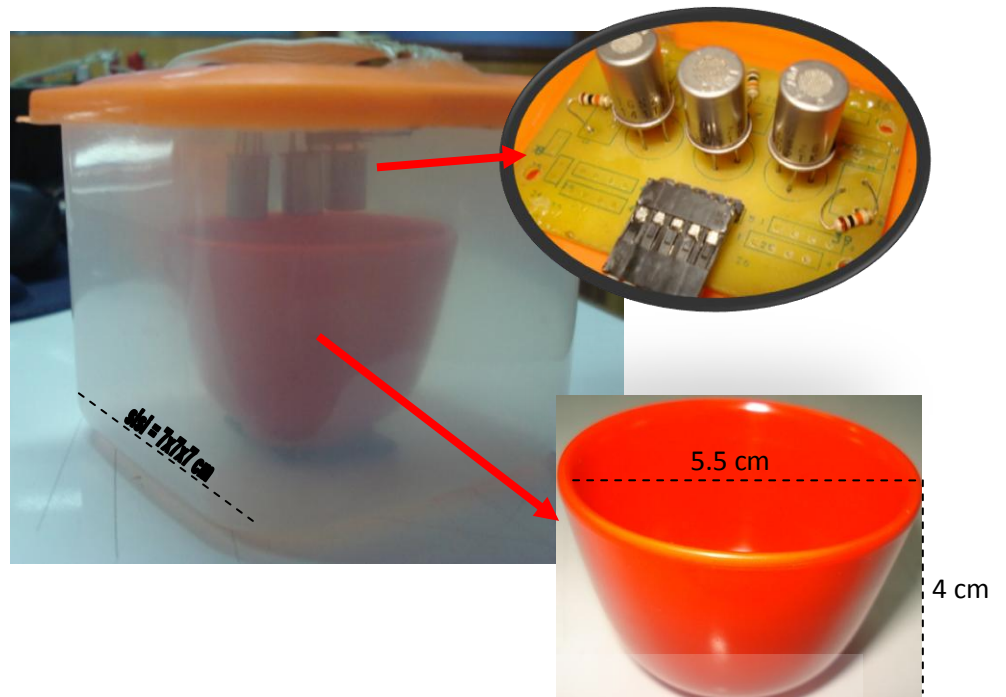
Gambar 3.56. Rancangan arsitektur sistem

Rancangan elektronika dari sistem yang dibuat dapat dilihat pada Gambar 3.56. Rancangan elektronika tersebut tersusun dalam bentuk *box* dengan ukuran 17x5x11cm, terdiri dari FPAA AN231K04, 2 tombol, 2 LED indikator dan didalam kotak terdapat rangkaian *minimum system*, *relay* serta beberapa potensio.



Gambar 3.57. Rancangan elektronika

Modul sensor *array* TGS yang telah dirancang dapat terlihat pada Gambar 3.58. Modul tersebut terdiri dari sensor TGS 2610, 2611, dan 2612. Kotak sensor berukuran 7x7x7cm, muk/mangkok kecil, dan suntik 3 cc/mL digunakan untuk memasukkan cairan, agar diperoleh takaran yang sesuai.



Gambar 3.58. Rancangan modul sensor *array* TGS

Injeksi jenis gas pada modul sensor *array* TGS saat dilakukan sampling dan pengujian dapat dilihat pada Gambar 3.59.



Gambar 3.59. Injeksi jenis gas modul sensor *array* TGS.