

PENGOLAHAN SINYAL RESPIRASI DENGAN FIR UNTUK ANALISA VOLUME DAN KAPASITAS PULMONARY

Moh. Fat'ak Diya'ul Haq, Kemalasari, Ardik wijayanto

Jurusan Teknik Elektronika, Politeknik Elektronika Negeri Surabaya (PENS)

Institut Teknologi Sepuluh Nopember (ITS), Surabaya

ABSTRAK

Beberapa penyakit paru-paru dapat mempengaruhi volume dan kapasitas paru-paru, misalnya penyakit asma atau emfisema yang terjadi akibat berkurangnya aliran udara yang keluar masuk paru-paru. Sinyal respirasi dapat digunakan untuk mengetahui volume dan kapasitas paru-paru. Oleh karena itu, pada proyek akhir ini dirancang suatu alat untuk mengetahui kapasitas dan volume paru-paru dengan menggunakan alat tiup yang dilengkapi dengan sensor *strain gauge* MPX5100. *Strain gauge* dengan tipe MPX5100 merupakan sensor tekanan yang peka terhadap tekanan rendah, hanya dengan tiupan saja dapat mempengaruhi tegangan output yang dihasilkan. Perubahan tegangan yang dihasilkan akan dikuatkan dengan *Op-Amp non-inverting* yang selanjutnya diintegrasikan dengan PC memanfaatkan mikrokontroler ATmega16 sebagai pengolah data analog ke digital. Input data dari mikrokontroler dimasukkan dalam aplikasi program *Visual Basic* untuk diubah menjadi bentuk sinyal/grafik. Sinyal ini diperhalus dengan menggunakan *filter FIR* untuk mengurangi *noise-noise* yang ada dan untuk memperjelas sinyal. Hasilnya berupa sinyal yang dapat digunakan untuk menentukan volume dan kapasitas paru-paru seseorang dengan prosentase error antara 10.953% - 37.81% untuk kondisi normal/rileks.

Kata kunci : Sinyal respirasi, MPX5100, *Op-Amp non-inverting* ATmega16, *Visual Basic*, *filter FIR*.

I. PENDAHULUAN

Beberapa penyakit paru-paru dapat mempengaruhi volume dan kapasitas paru-paru, seperti penyakit asma atau emfisema yang terjadi akibat berkurangnya aliran udara yang keluar masuk paru-paru yang menyebabkan menyempitnya diameter saluran udara ke paru-paru sehingga terjadi hambatan pada saluran tersebut.

Sinyal respirasi dapat digunakan untuk mengetahui volume dan kapasitas paru-paru sehingga bisa mendeteksi ada tidaknya penyakit pada paru-paru.

Strain gauge dengan tipe MPX5100 merupakan sensor tekanan yang peka terhadap tekanan rendah, hanya dengan tiupan saja dapat mempengaruhi tegangan output yang dihasilkan. Oleh karena itu, pada proyek akhir ini dirancang suatu alat untuk mengetahui kapasitas dan volume paru-paru dengan menggunakan alat tiup

yang dilengkapi dengan sensor *strain gauge* MPX5100. Tegangan output dari sensor pada alat tiup dikuatkan dengan penguat *Op-Amp non-inverting* agar bisa dibaca mikrokontroler ATmega16.

Pembacaan tegangan pada mikrokontroler akan diubah menjadi grafik atau sinyal dengan memanfaatkan program aplikasi *Visual Basic*. Sinyal yang didapatkan dikurangi *noisenya* dan diperhalus dengan menggunakan *filter FIR* sehingga menghasilkan sinyal yang dapat digunakan untuk menentukan volume dan kapasitas paru-paru seseorang.

II. TEORI PENUNJANG

2.1 Mekanisme Pernapasan pada Manusia

Ada dua macam mekanisme pernapasan pada manusia, yaitu:

2.1.1 Pernapasan Dada

Pernapasan dada terjadi karena gerakan tulang-tulang rusuk oleh otot-otot antar tulang rusuk (*interkostal*). Mekanismenya dapat dibedakan sebagai berikut:

a. Fase Inspirasi

Terjadi jika otot-otot tulang rusuk berkontraksi sehingga tulang-tulang rusuk terangkat keatas, demikian pula tulang dada ikut terangkat keatas. Akibatnya rongga dada membesar dan paru-paru juga ikut membesar, sehingga tekanan udara dalam paru-paru berkurang dan udara luar masuk ke paru-paru.

b. Fase Ekspirasi

Terjadi jika otot-otot antar tulang rusuk relaksasi, yaitu tulang rusuk dan tulang dada turun kembali pada kedudukan semula sehingga rongga dada mengecil. Akibatnya volume paru-paru berkurang maka tekanan udara dalam paru-paru bertambah, sehingga udara keluar.

2.1.2 Pernapasan Perut

Pernapasan perut terjadi karena gerakan otot diafragma (sekat rongga badan yang membatasi rongga dada dan rongga perut). Mekanismenya adalah sebagai berikut:

a. Fase Inspirasi

Terjadi jika otot diafragma berkontraksi sehingga letaknya agak mendatar, yang menyebabkan rongga dada membesar dan paru-paru ikut membesar. Akibatnya tekanan udara dalam paru-paru berkurang sehingga udara luar masuk ke paru-paru.

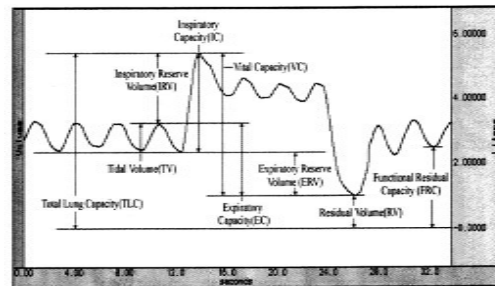
b. Fase Ekspirasi

Terjadi jika otot diafragma mengendur kembali pada kedudukan semula, sehingga rongga dada mengecil dan paru-paru pun ikut mengecil. Oleh karena itu volume paru-paru berkurang dan tekanan udara dalam paru-paru bertambah akibatnya udara keluar dari paru-paru.

2.2 Volume dan Kapasitas Udara Pernapasan

Macam-macam pengukuran volume paru-paru adalah sebagai berikut:

1. *Tidal volume (TV)* : Volume udara hisap dan hembus dari sekali napas (normal 500 ml, saat olah raga lebih dari 3 liter).
2. *Inspiratory reserve volume (IRV)* : Volume udara saat hisap napas maksimal sampai akhir TV (normal/santai mencapai 3.300 ml pada usia muda laki-laki dan 1.900 ml pada usia muda perempuan).
3. *Expiratory reserve volume* : Volume udara saat hembus napas maksimal sampai akhir TV (normal/santai mencapai 1.000 ml pada usia muda laki-laki dan 700 ml pada usia muda perempuan).
4. *Residual volume (RV)* : Volume dari sisa gas dalam paru-paru saat hembus maksimal (*expiratory*). RV tidak akan berubah saat santai atau setelah olah raga (rata-rata RV 1.200 ml untuk orang laki-laki dan 1.100 ml untuk orang perempuan).



Gambar 2.1 Kapasitas dan Volume Pulmonary

Kapasitas paru-paru adalah jumlah dari dua atau beberapa volume utama. Ada 5 macam kapasitas paru-paru, yaitu:

1. *Inspiratory Capacity (IC)*
 $IC = TV + IRV$
2. *Expiratory Capacity (EC)*
 $EC = TV + ERV$
3. *Functional Residual Capacity (FRC)*
 $FRC = ERV + RV$
4. *Vital Capacity (VC)*
 $VC = IRV + TV + ERV$

5. Total Lung Capacity (TLC)

$$TLC = IRV + TV + ERV + RV$$

2.3 Pengukuran Kapasitas Paru-paru

Rumus untuk mengukur kapasitas vital paru-paru

a. Untuk laki-laki:

$$VC = 0.052T - 0.022U - 3.00$$

b. Untuk perempuan:

$$VC = 0.041T - 0.018U - 2.69$$

Keterangan:

U = Umur

T = Tinggi badan

80 % dari nilai VC sudah dianggap sehat (normal)

2.4 MPX5100

MPX5100 adalah *Strain Gauge* jenis *piezoresistif transducer* berbahan silikon yang terintegrasi dalam sebuah *chip*, bekerja pada tekanan 0 kPa sampai 100 kPa (0 psi sampai 14.5 psi) atau 15 kPa sampai 115 kPa (2.18 psi sampai 16.68 psi) dengan tegangan output 0.2 volt sampai 4.7 volt.



Gambar 2.2 Komponen MPX5100

Sensor tekanan ini didesain untuk aplikasi *range* yang lebar, terutama bekerja pada mikrokontroler atau mikroprosesor dengan analog/digital input, terbuat dari elemen *transducer* tunggal yang dikombinasikan menggunakan teknik *micromachining* dengan logam film tipis dan diproses secara bipolar untuk menghasilkan output sinyal analog level tinggi yang akurat dan proporsional untuk aplikasi tekanan.

Adapun konfigurasi *pin-pin* pada kaki komponen ini adalah sebagai berikut:

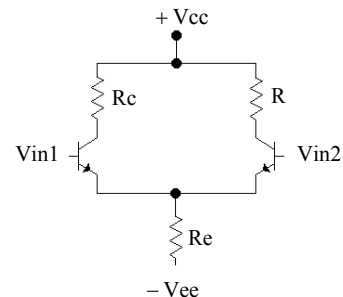
Tabel 2.1 Pin Number dari MPX5100

PIN NUMBER	
1	V _{Out}
2	Gnd
3	V _s
4	N/C
5	N/C
6	N/C

Catatan: Pin 4,5,6 adalah sambungan internal *device*, jangan disambungkan dengan eksternal *circuit* atau ground.

2.5 Operational Amplifier

Operasional Amplifier disingkat *Op-Amp* adalah *Interated Circuit (IC)* untuk rangkaian linier. *Op-Amp* ini bisa digunakan untuk membuat rangkaian elektronika analog. *Operasional Amplifier* berasal dari rangkaian differensial transistor yaitu dari rangkaian dengan 2 input dan 2 output dengan supply ganda seperti pada gambar 2.6. Rangkaian ini adalah sebagai pembeda atau pembanding dari referensi input yang lainnya.



Gambar 2.3 Dasar *Op-Amp* Sederhana

Operasional Amplifier (Op-Amp) mempunyai 2 macam input (*inverting* dan *non-inverting*) dan sebuah output, sedangkan yang lainnya adalah seperti setting tegangan, kaki tegangan supply, dan lain lain. *Op-Amp* yang ideal mempunyai sifat :

1. Penguatnya sangat besar
2. Tahanan inputnya sangat kecil, sehingga tegangan input sangat kecil

- Tahanan outputnya sangat besar, sehingga tegangan outputnya sangat besar

2.5.1 Non-Inverting

Rangkaian *non-inverting* hampir sama dengan rangkaian *inverting* hanya perbedaannya terletak pada tegangan input dari masukan *non-inverting*.

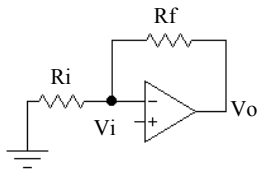
Rumusnya seperti berikut :

$$V_o = \frac{R_f + R_i}{R_i} V_i$$

sehingga persamaannya menjadi

$$V_o = \left(\frac{R_f}{R_i} + 1 \right) V_i$$

Hasil tegangan output *non-inverting* ini akan lebih dari satu dan selalu positif.



Gambar 2.4 Rangkaian *Non-Inverting Amplifier*

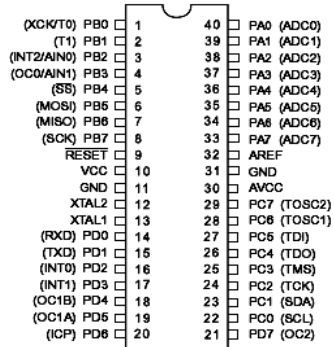
2.6 Mikrokontroler AVR ATmega16

AVR merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur *RISC (Reduced Instruction Set Computer)*. Hampir semua instruksi dieksekusi dalam satu siklus *clock*. AVR mempunyai 32 *register general-purpose, timer/counter* fleksibel dengan *mode compare, interrupt* internal dan eksternal, serial UART, *programmable Watchdog Timer*, dan *mode power saving*. Mempunyai ADC dan PWM internal. AVR juga mempunyai *In-System Programmable Flash on-chip* yang memungkinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI.

Atmega16 adalah mikrokontroler CMOS 8-bit daya-rendah berbasis arsitektur *RISC* yang ditingkatkan. Kebanyakan instruksi dikerjakan pada satu

siklus *clock*, Atmega16 mempunyai *throughput* mendekati 1 MIPS per MHz membuat disainer sistem untuk mengoptimasi konsumsi daya versus kecepatan proses.

Konfigurasi pin ditunjukkan pada gambar 2.7 dibawah ini:

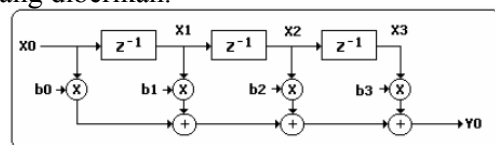


Gambar 2.5 Pin-pin ATmega16 kemasan 40-pin

Guna memaksimalkan performa dan paralelisme, AVR menggunakan arsitektur Harvard (dengan memori dan bus terpisah untuk program dan data). Instruksi pada memori program dieksekusi dengan *pipelining single level*. Selagi sebuah instruksi sedang dikerjakan, instruksi berikutnya diambil dari memori program.

2.7 Finite Impulse Response (FIR)

Filter FIR adalah salah satu tipe dari *filter* digital yang dipakai pada aplikasi *Digital Signal Processing (DSP)*. *FIR* kepanjangan dari *Finite Impulse Response*. Dalam *filter* ini tidak ada *feedback* sehingga responnya dikatakan berhingga/terbatas. Keuntungan *filter FIR* antara lain adalah stabil dan memiliki fasa yang linier. Sedangkan kerugiannya adalah *filter FIR* terkadang membutuhkan lebih banyak memory dan/atau perhitungan untuk mencapai karakteristik respon *filter* yang diberikan.



Gambar 2.6 Flowgraph *FIR* Orde 3

Bentuk respon impulse *filter FIR* secara umum dapat dinyatakan dalam persamaan:

$$y[n] = \sum_{i=0}^N b_i \cdot x[n-i] = \sum_{k=0}^M b_k \cdot x[n-k]$$

$$b_k = \frac{1}{M+1}$$

Dimana:

M = Orde *filter*

k = indek *filter*

b_k = koefisien *filter*

2.8 Visual Basic 6.0

Visual Basic adalah salah suatu *development tools* untuk membangun aplikasi dalam lingkungan *windows*. Dalam pengembangan aplikasi, *Visual Basic* menggunakan pendekatan *Visual* untuk merancang *user interface* dalam bentuk *form*, sedangkan untuk kodingnya menggunakan dialek bahasa *Basic* yang cenderung mudah dipelajari. *Visual Basic* telah menjadi *tools* yang terkenal bagi para pemula maupun para developer. Dalam lingkungan *window's user-interface* sangat memegang peranan penting, karena dalam pemakaian aplikasi yang dibuat, pemakai senantiasa berinteraksi dengan *user-interface* tanpa menyadari bahwa dibelakangnya berjalan instruksi-instruksi program yang mendukung tampilan dan proses yang dilakukan.

Pada pemrograman *Visual*, pengembangan aplikasi dimulai dengan pembentukan *user interface*, kemudian mengatur properti dari objek-objek yang digunakan dalam *user interface*, dan baru dilakukan penulisan kode program untuk menangani kejadian-kejadian (*event*).

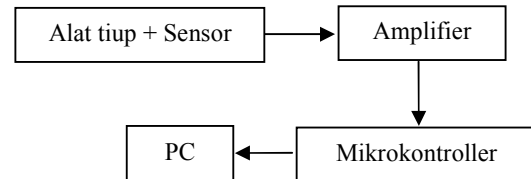
III. PERENCANAAN SISTEM

Pada bab ini akan dibahas mengenai perencanaan sistem secara keseluruhan, mulai dari blok diagram sistem, perencanaan *hardware*, perencanaan

software, serta pengambilan data dari biopac sebagai pembanding data yang dihasilkan oleh alat tiup.

3.1 Blok Diagram Sistem

Sebelum merancang dan membuat *hardware* dan *software* perlu dipahami terlebih dulu susunan atau blok diagram dari sistem itu sendiri.



Gambar 3.1 Blok Diagram Sistem

Sensor yang diintegrasikan pada alat tiup akan menghasilkan output berupa perubahan tegangan, tegangan output yang dihasilkan akan dikuatkan oleh penguat *Op-Amp non-inverting* kemudian dihubungkan dengan mikrokontroller ATmega16 untuk diubah datanya dari analog ke digital dengan memanfaatkan ADC internal mikro selanjutnya diintegrasikan ke PC menggunakan komunikasi serial Max 232 untuk diolah secara *software*.

3.2 Perencanaan dan Pembuatan Sistem

Perencanaan sistem pada proyek akhir ini secara keseluruhan dibagi menjadi dua bagian, yaitu: perencanaan *hardware* dan perencanaan *software*.

3.2.1 Perencanaan *Hardware*

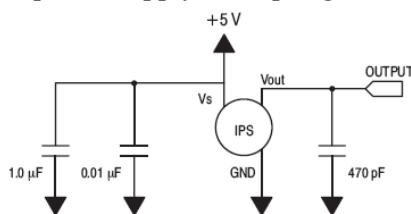
Hardware yang dipakai untuk mendukung terselesainya proyek akhir ini adalah:

3.2.1.1 Rangkaian Sensor

Sensor yang digunakan pada proyek akhir ini adalah MPX5100 untuk membangkitkan perubahan tegangan

ketika terjadi perubahan resistansi akibat tekanan berupa tarikan dan hembusan napas.

MPX5100 adalah *strain gauge* jenis *piezoresistif transducer* berbahan silikon yang terintegrasi dalam sebuah *chip* yang didesain untuk aplikasi *range* yang lebar, terutama bekerja pada mikrokontroler atau mikroprosesor dengan analog/digital input. Untuk *interfacing* output dari sensor ini ke analog/digital input dari mikroprosesor atau mikrokontroler direkomendasikan untuk menggunakan *circuit decoupling*, yaitu *power supply decoupling*.



Gambar 3.2 Rangkaian Sensor dengan Power Supply Decoupling

Tegangan output dari sensor ini berkisar antara 0.2 volt sampai 4.7 volt, untuk mengantisipasi adanya tegangan rendah perlu dikuatkan terlebih dahulu sebelum diintegrasikan kedalam mikrokontroler.

3.2.1.2 Rangkaian Penguat (*Amplifier*)

Penguat yang digunakan dalam proyek akhir ini adalah penguat *Op-Amp non-inverting* karena penguatan yang dibutuhkan bernilai positif, yaitu untuk menaikkan tegangan output dari sensor MPX5100 sehingga bisa dibaca mikrokontroler ATmega16. Tegangan Output yang dihasilkan oleh penguat ini sesuai dengan persamaan:

$$V_o = \left(\frac{R_f}{R_i} + 1\right)V_i$$

Sedangkan persamaan penguatannya adalah:

$$A_v = \frac{V_o}{V_i} = \frac{R_f}{R_i} + 1$$

Dengan:

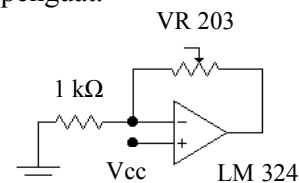
- V_o = Tegangan output
- R_f = Hambatan referensi
- R_i = Hambatan input
- V_i = Tegangan input
- A_v = Penguatan tegangan

Op-Amp yang digunakan untuk rangkaian penguat dalam proyek akhir ini adalah IC (*Integrated Circuit*) LM 324, R_i yang digunakan bernilai 1 k Ω dan R_f berupa Variabel Resistor 203 (20 k Ω). Dengan komponen-komponen tersebut nilai penguatan maksimal yang bisa dihasilkan adalah:

$$A_v = \frac{R_f}{R_i} + 1$$

$$= \frac{20}{1} + 1 = 21 \text{ kali}$$

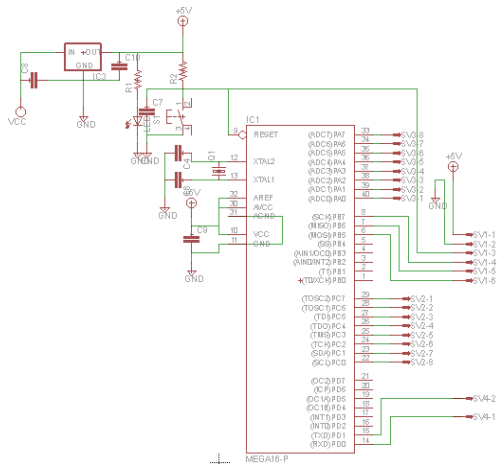
Dengan tegangan input antara 0.2 volt sampai 4.7 volt dan penguatan maksimal sebesar 21 kali bisa didapatkan tegangan output sesuai kebutuhan untuk bisa dibaca mikrokontroler dengan mengatur penguatan yang dihasilkan rangkaian penguat.



Gambar 3.3 Rangkaian *Op-Amp Non-Inverting*

3.2.1.3 Mikrokontroler ATmega16

Sistem pada proyek akhir ini menggunakan mikrokontroler ATmega16 sebagai pengonversi data dari analog ke digital memanfaatkan ADC internal pada mikrokontroler itu sendiri untuk bisa ditampilkan pada PC menggunakan komunikasi serial RS 232. Adapun rangkaian mikrokontroler itu sendiri seperti tampak pada gambar 3.4 berikut:



Gambar 3.4 Rangkaian Mikrokontroler ATmega16

3.2.2 Perencanaan *Software*

Software yang digunakan pada proyek akhi ini antara lain *Code Vision AVR* ATmega16, *Visual Basic*, serta *filter FIR*.

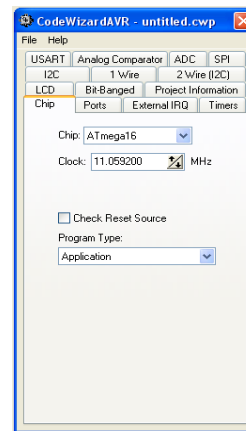
3.2.2.1 Program *CodeVision AVR* ATmega16

CodeVision AVR merupakan *software C-cross compiler*, dimana program dapat ditulis menggunakan bahasa-C. Dengan menggunakan pemrograman bahasa-C diharapkan waktu disain (*developing time*) akan menjadi lebih singkat. Setelah program dalam bahasa-C ditulis dan dilakukan kompilasi tidak terdapat kesalahan (*error*) maka proses download dapat dilakukan. Mikrokontroler *AVR* mendukung sistem download secara *ISP (In-System Programming)*.

Gambar di bawah tampilan *Code Vision AVR* dalam melakukan setting jenis IC yang digunakan, yaitu ATmega16 serta *clock* yang dipakai berdasarkan nilai komponen kristal pada *hardware* yaitu 11.0592 MHz.

Untuk aplikasi apa saja yang ingin dipakai dalam pemrograman *AVR* ini, perlu di set terlebih dahulu aplikasi-aplikasi tersebut, misalnya dalam hal ini,

penggunaan *ADC* dengan menggunakan 8 bit data serta transfer data menggunakan *USART* seperti tampak pada gambar 3.6 di bawah ini.



Gambar 3.5 *Lay out* Pemilihan jenis *Chip* dan *Clock* pada *AVR*

Karena pada proyek akhir ini mikrokontroler digunakan untuk konversi data analog ke digital serta transfer data dari mikro ke PC maka untuk pemilihan setting aplikasi program pada *AVR* disesuaikan dengan kebutuhan tersebut. *ADC* yang digunakan adalah 8 bit data, sedangkan pemakaian *USART* hanya pada *transmitter* saja, karena yang dibutuhkan hanya pengiriman data.

Adapun program yang digunakan untuk pembacaan *ADC* pada mikrokontroler adalah sebagai berikut:

```
// Declare your global variables here
int a;

while (1)
{
// Place your code here
a = read_ADC(1);
printf("%d ",a);
delay_ms(100);
};
```

Transfer data dilakukan dengan pembacaan *ADC* pada port A(1) dengan koefisien delay 100 ms. Artinya data akan dikirim ke PC setiap 100 mili sekon sekali.

3.2.2.2 Pemrograman *Visual Basic*

Pembangkitan grafik atau sinyal dari input tegangan digital dari mikrokontroler memanfaatkan pemrograman *Visual Basic*. Pembuatan program aplikasi menggunakan *Visual Basic* dilakukan dengan membuat tampilan aplikasi pada *form*, kemudian diberi script program di dalam komponen-komponen yang diperlukan. Form disusun oleh komponen-komponen yang berada di [*Toolbox*], dan setiap komponen yang dipakai harus diatur propertinya lewat jendela [*Property*]. Untuk memulai pembuatan program aplikasi di dalam *Visual Basic*, yang dilakukan adalah membuat *project* baru. *Project* adalah sekumpulan *form*, modul, fungsi, data dan laporan yang digunakan dalam suatu aplikasi. Membuat *project* baru dapat dilakukan dengan memilih menu [*File*] >> [*New Project*] atau dengan menekan ikon [*new project*] pada *Toolbar* yang terletak di pojok kiri atas.



Gambar 3.6 Layar Pemilihan Jenis *Project*

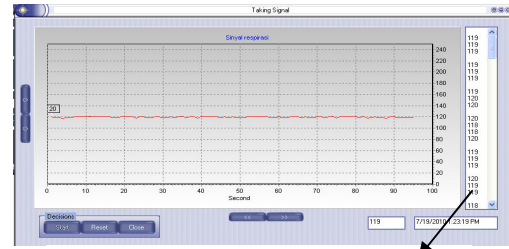
Selanjutnya pilih Standard EXE dan tekan [Ok]. Lalu muncul tampilan dari Standard Exe seperti pada gambar 3.9 Dengan demikian *project* sudah siap dibuat.

IV. PENGUJIAN DAN ANALISA

4.1 Pembacaan data

Data-data yang ada berupa tegangan output dari penguat yang dikirim oleh mikrokontroler dibaca oleh *Visual Basic*

dengan memanfaatkan komponen *MSComm* dengan *CommPort 1* (sesuai dengan *CommPort* di PC).



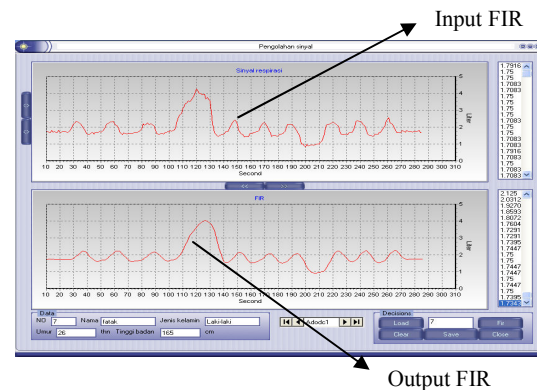
Gambar 4.1 Sinyal Awal Pembacaan *Visual Basic*

Nilai awal pembacaan *Visual Basic* sesuai dengan nilai output dari mikrokontroler yaitu sekitar 120 yang ditampilkan pada *form* dengan memanfaatkan komponen *TChart*.

Untuk menyesuaikan sinyal yang didapat dengan yang hasil dari biopac dilakukan pengkalibrasian dengan melakukan penapasan biasa dan penapasan dalam pada masing-masing alat.

4.2 Penggunaan FIR

Pembacaan nilai yang dilakukan oleh *Visual Basic* setelah dimodifikasi dengan nilai koefisien rumus hasil kalibrasi menunjukkan grafik/sinyal yang masih terdapat banyak *noise*, untuk itu diperlukan sebuah *filter* untuk mengurangi *noise* serta memperhalus sinyal. Dengan memanfaatkan *general FIR* orde 8 dapat diperoleh sinyal yang lebih halus.



Gambar 4.2 Sinyal Input dan Output *FIR*

Dari sinyal pada gambar 4.8 diatas menunjukkan adanya perbedaan antara input dan output *filter* . Sinyal output dari FIR lebih halus dan sedikit *noise*, sehingga bisa diketahui dengan jelas amplitudo dari masing-masing sinyal yang dihasilkan untuk digunakan menentukan volume dan kapasitas paru-paru.

Program *FIR* itu sendiri pada *Visual Basic* adalah:

```

For i = 0 To c - 9
ReDim Preserve Y(i)
Y(i) = (1 / 9) * (data(i) + data(i + 1)
+ data(i + 2) + data(i + 3) + data(i + 4)
+ data(i + 5) + data(i + 6) +
data(i + 7) + data(i + 8))
List2.AddItem Y(i)
TChart2.Series(0).asFastLine.AddRealTime i, Y(i)
Next i

```

Penggunaan orde 8 dari FIR tersebut disesuaikan dengan bentuk sinyal yang diharapkan. Bila orde terlalu rendah maka pengurangan *noise* terlalu sedikit, akibatnya *noise* yang tampak masih terlalu banyak, jika orde semakin besar *noise* akan semakin hilang, amplitudo semakin kecil, sehingga semakin tidak kelihatan.

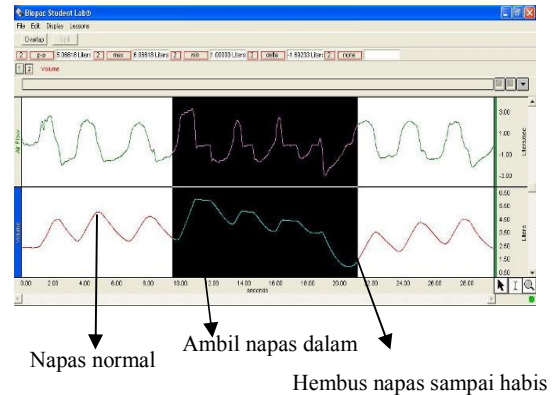
4.3 Perbandingan dengan Hasil Pengukuran Biopac

Untuk mengetahui ketepatan hasil yang didapatkan dari pemfilteran, perlu dilakukan perbandingan dengan hasil pengukuran biopac, sehingga bisa ditentukan tingkat kepresisian dari alat yang dibuat. Sinyal hasil pengukuran alat tiup yang telah di *filter* dengan FIR diharapkan menyerupai sinyal hasil pengukuran biopac, minimal untuk nilai amplitudo masing-masing sinyal yang didapat sehingga nilai dari volume dan kapasitas paru-paru hasil pengukuran alat tiup dan hasil pengukuran biopac menunjukkan selisih yang tidak terlalu besar.

Contoh perbandingan sinyalnya:

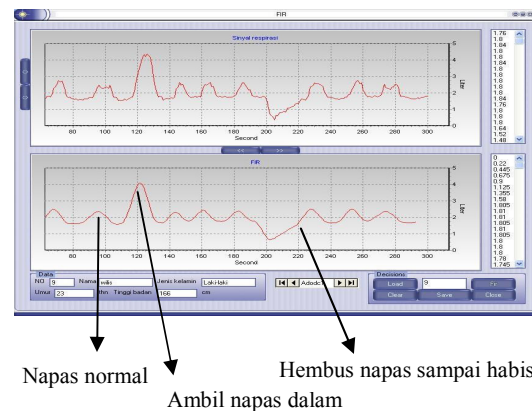
- Nama : Wilis aji Nu projo
- Jenis kelamin : Laki-laki
- Umur : 23 Tahun
- Tinggi badan : 165 Cm

Hasil pengukuran biopac:



Gambar 4.3 Hasil Pengukuran Biopac

Hasil pengukuran alat tiup:



Gambar 4.4 Hasil Pengukuran Alat Tiup

Berdasarkan perbandingan sinyal pada gambar 4.9 – gambar 4.12 menunjukkan bahwa sinyal/grafik hasil pengukuran dengan alat tiup menyerupai sinyal hasil pengukuran biopac meskipun masih terdapat perbedaan terutama pada prosedur pengukuran hembus napas biasa 3 kali setelah tarik napas dalam. Pada pengukuran alat tiup hembus napas biasa 3 kali setelah tarik napas dalam menyerupai napas biasa 3 kali diawal pengukuran, sedangkan pada pengukuran biopac hembus napas biasanya sedikit diatas

napas biasa, namun hal ini tidak mempengaruhi nilai volume dan kapasitas paru-paru yang dihasilkan masing-masing alat karena nilai yang diambil berdasarkan amplitudo napas biasa dan tarik napas dalam serta hembus napas sampai habis dari masing-masing pengukuran.

V. PENUTUP

5.1 KESIMPULAN

Setelah melakukan tahap perencanaan dan pembuatan sistem yang kemudian dilanjutkan dengan tahap pengujian dan analisa maka dapat diambil kesimpulan sebagai berikut :

- Sensor MPX5100 dapat digunakan untuk mengukur tekanan dalam bentuk tarikan dan hembusan napas, baik pernapasan biasa maupun pernapasan dalam.
- Penggunaan VR (Variabel Resistor) sebagai Rf pada rangkaian *Op-Amp Non-Inverting* memudahkan untuk mendapatkan penguatan sesuai dengan kebutuhan.
- *Filter FIR* dapat digunakan untuk menghaluskan sinyal sehingga dapat dimanfaatkan untuk menentukan volume dan kapasitas paru-paru.
- Berdasarkan hasil dari alat yang telah dibuat, data dari biopac serta perhitungan secara teori menunjukkan bahwa jenis kelamin, umur dan tinggi badan seseorang dapat mempengaruhi nilai volume dan kapasitas paru-parunya.
- Hasil pengukuran dari alat tiup bila dibandingkan dengan hasil dari pengukuran biopac untuk kondisi normal/rileks prosentase errornya berkisar antara:
 - VC : 0.555% - 34.97%
 - IRV : 7.004% - 40.32%
 - ERV : 3.123% - 50.073%
 - TV : 6.547% - 41.73%
- Prosentase error kapasitas vital paru-paru (VC) hasil pengukuran dengan alat tiup dengan perhitungan secara teori berkisar antara 10.953% - 37.81% untuk kondisi normal/rileks, 1.15% - 26.67% untuk kondisi setelah makan dan 1.99% - 41.17% untuk kondisi setelah olah raga/aktivitas.

5.2 SARAN

Dari hasil PA ini masih terdapat banyak kekurangan dan membutuhkan banyak penyempurnaan, untuk itu bisa dijadikan masukan saran-saran sebagai berikut:

- Pengolahan sinyal respirasi ini perlu dikembangkan untuk bisa mendapatkan bentuk sinyal yang sesuai dengan sinyal hasil

pengukuran biopac sehingga bisa didapatkan volume dan kapasitas paru-paru yang presisi.

- Perlu dilakukan banyak penelitian untuk bisa menganalisa sinyal respirasi sehingga bisa digunakan untuk mengidentifikasi ada tidaknya penyakit pada paru-paru.
- Tindak lanjut dari penelitian ini bisa dilakukan untuk mengetahui penyebab adanya perbedaan nilai volume dan kapasitas paru-paru untuk masing-masing kondisi obyek saat pengukuran.

DAFTAR PUSTAKA

- [1] Andrianto, Heri, *Pemrograman mikrokontroller AVR ATMEGA16*, Informatika, Bandung, 2008.
- [2] Arief, Zaenal, Adil, Ratna, Kemalasari, *Buku Praktikum Elektronika Medika*, PENS, Surabaya, 2005.
- [3] Happyanto, Dedid Cahya, *SENSOR*, PENS, Surabaya, 2005.
- [4] Oktavianto, Harry, *Praktikum Pengolahan Sinyal*, PENS, Surabaya, 2007.
- [5] Pitoyo, Edhy, *Respirasi Hardware*, proyek Akhir : T. Elektronika PENS-ITS, PENS, Surabaya, 2003.
- [6] Pratiwi, D.A., Maryati Sri, *Buku Penuntun Biologi untuk SMU kelas 2*, Erlangga, Jakarta, 2003.
- [7] Thabrani, Suryanto, *Mudah & Cepat Menguasai Visual Basic*, Mediakita, Jakarta, 2007.
- [8] www.freescale.com/files/sensors/doc/data_sheet/MPX5100.pdf diakses pada Oktober 2009.
- [9] www.en.wikipwdia.org/wiki/Fir_filter diakses pada Juli 2010.