

**RANCANG BANGUN SISTEM PEREKAMAN SINYAL EMG UNTUK
MONITORING PERKEMBANGAN PASIEN PASCA STROKE**

(BAGIAN II)

TUGAS AKHIR



Oleh:

AISHATUL MARIFAH

NIM. 081310213032

PROGRAM STUDI D3 OTOMASI SISTEM INSTRUMENTASI

DEPARTEMEN TEKNIK

FAKULTAS VOKASI

UNIVERSITAS AIRLANGGA

2016

LEMBAR PERSETUJUAN TUGAS AKHIR

**RANCANG BANGUN SISTEM PEREKAMAN SINYAL EMG UNTUK
MONITORING PERKEMBANGAN PASIEN PASCA STROKE
(BAGIAN II)**

TUGAS AKHIR

Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Ahli Madya (A.Md)

Bidang Otomasi Sistem Instrumentasi
Pada Departemen Teknik Fakultas Vokasi
Universitas Airlangga

Oleh :

Aishatul Marifah
NIM. 081310213032

Disetujui Oleh :

Dosen Pembimbing,

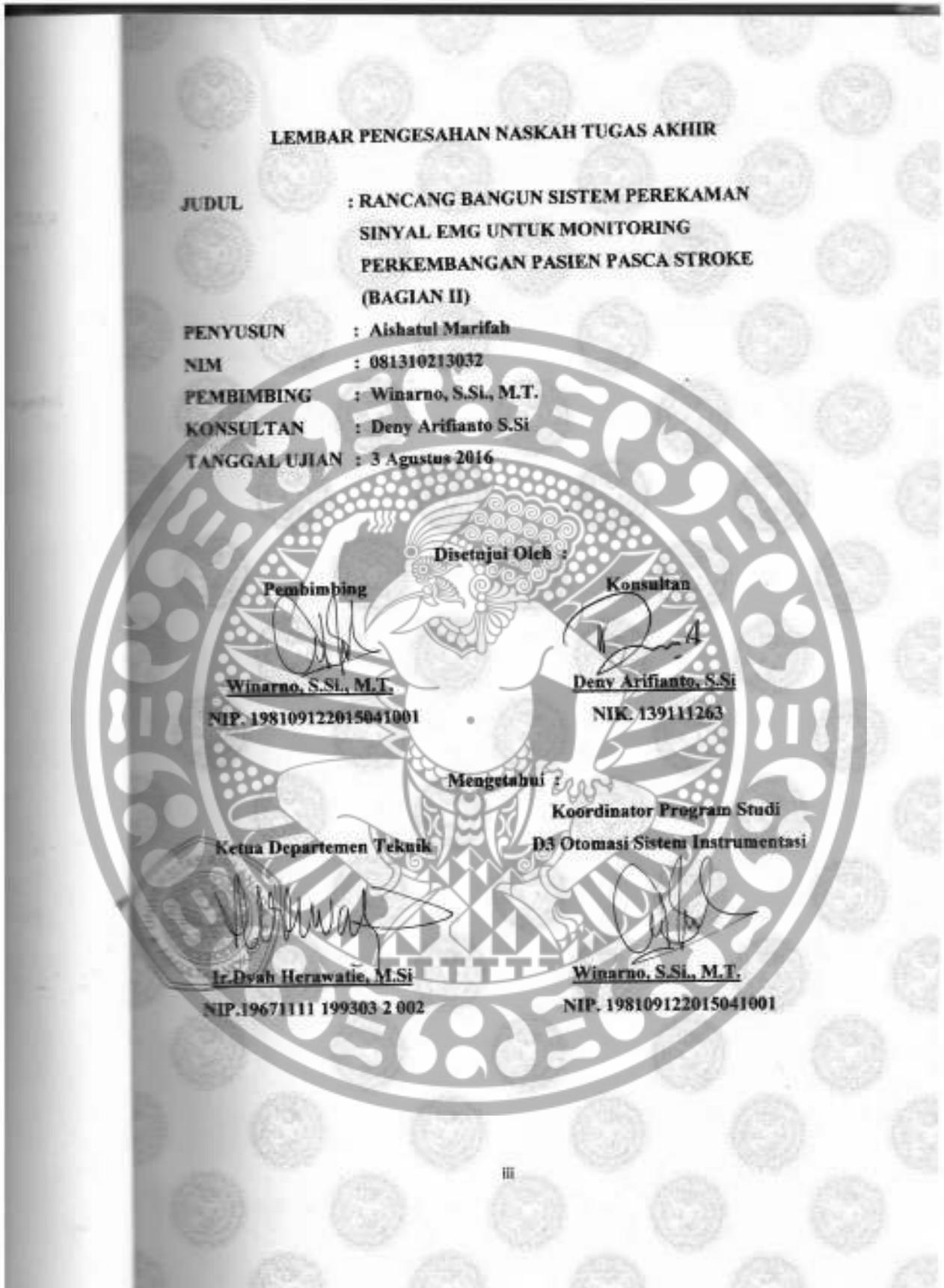
Winarno, S.Si., M.T.

NIP. 198109122015041001

Dosen Konsultan,

Denny Arifianto, S.Si

NIK. 139111263



LEMBAR PENGESAHAN NASKAH TUGAS AKHIR

**JUDUL : RANCANG BANGUN SISTEM PEREKAMAN
SINYAL EMG UNTUK MONITORING
PERKEMBANGAN PASIEN PASCA STROKE
(BAGIAN II)**

PENYUSUN : Aishatul Marifah

NIM : 081310213032

PEMBIMBING : Winarno, S.Si., M.T.

KONSULTAN : Deny Arifianto S.Si

TANGGAL UJIAN : 3 Agustus 2016

Disetujui Oleh :

Pembimbing

Winarno, S.Si., M.T.

NIP. 198109122015041001

Konsultan

Deny Arifianto, S.Si

NIK. 139111263

Mengetahui :

Ketua Departemen Teknik

Ir. Dyah Herawati, M.Si

NIP. 19671111 199303 2 002

**Koordinator Program Studi
D3 Otomasi Sistem Instrumentasi**

Winarno, S.Si., M.T.

NIP. 198109122015041001

PEDOMAN PENGGUNAAN TUGAS AKHIR

Tugas Akhir ini tidak dipublikasikan, namun tersedia di perpustakaan dalam lingkungan Universitas Airlangga. Diperkenankan untuk dipakai sebagai referensi kepustakaan, tetapi pengutipan seijin penulis dan harus menyebutkan sumbernya sesuai kebinasaan ilmiah.

Dokumen Tugas Akhir ini merupakan hak milik Universitas Airlangga.



KATA PENGANTAR

Segala puji syukur kami panjatkan kehadirat Allah SWT atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul **“Rancang Bangun Sistem Perekaman Sinyal EMG untuk Monitoring Perkembangan Pasien Pasca Stroke”**.

Dalam pembuatan laporan ini kami tidak lepas bantuan dari berbagai pihak. Untuk itu dengan segala kerendahan hati, saya menyampaikan terimakasih kepada:

1. Allah SWT yang telah memberikan rahmat serta hidayah-Nya.
2. Mami & Dad serta keluarga besar yang selalu memberikan support dan menguatkan setiap saat.
3. Bapak Dr. H. Widi Hidayat, SE., M.Si Ak., CMA, CA. Selaku Dekan Fakultas Vokasi Universitas Airlangga.
4. Ibu Ir. Dyah Herawati, M.Si selaku Ketua Departemen Teknik Fakultas Vokasi universitas Airlangga.
5. Bapak Winarno S.Si., M.T. selaku Koordinator Program Studi D3 Otomasi Sistem Instrumentasi Fakultas Vokasi Universitas Airlangga.
6. Bapak Winarno S.Si., M.T. selaku Dosen Pembimbing yang telah membimbing saat melaksanakan Tugas Akhir.
7. Bapak Deny Arifianto S.Si. selaku Dosen Konsultan yang telah sabar dan banyak meluangkan waktu untuk mengarahkan serta memberi banyak pembelajaran mengenai Tugas Akhir.
8. Mba Tya selaku Administrasi Prodi Otomasi Sistem Instrumentasi yang selalu menginformasikan, membantu serta mengarahkan untuk segera menyelesaikan Tugas Akhir.

9. Partner TA, Rizky Altryara untuk dukungan, semangat serta bantuan dalam proses penyelesaian Tugas Akhir.
10. Member Of Rumpita (MOR) yang terdiri dari sofi, mimink, dita, dewi, wanda, aliyah, meme, oneng serta dms dan acl yang selalu banyak memberikan bantuan berupa support mental dan juga bahu untuk bersandar ketika menangis.
11. Seluruh teman-teman prodi OSI angkatan 2013 ayahbagir, zu, php, jojo, gukjunpyo, ilham, arab, abik, kakak ud, realman, andin, nispul, affan, kakak lim, kakakwjo, duddin, isadayana, cececela, bosamir, rifand, difa, kembar, oscar, fauji, fida dan cincong. Terimakasih atas kesolidaritan dan kekompakkan kalian selama 3 tahun.
12. Guk & Yuk Sidoarjo, yuk windy, guk najibs, yuk dela, yuk adel, yuk Lanny, Yuk Widya dan Duta Fakultas Vokasi yang selalu mendoakan dan menyemangati supaya Tugas akhir segera terselesaikan serta menjadi penghibur dikala butuh hiburan saat menyelesaikan Tugas Akhir.
13. Miftha, Natya, Arini, Ella dan juga teman teman lain yang tidak bisa saya sebutkan satupersatu namanya disini terimakasih sekali atas segala yang telah kalian berikan demi mendukung dan mensukseskan Tugas Akhir saya.

Akhir kata dengan segala keterbatasan, saya berharap Tuga Akhir ini dapat bermanfaat bagi semua terutama di Program Studi D3 Otomasi Sistem Instrumentasi Fakultas Vokasi Universitas Airlangga.

Surabaya, 19 Juli 2016

Aishatul Marifah, 2016, *Rancang Bangun Sistem Perekaman Sinyal EMG untuk Monitoring Perkembangan Pasien Pasca Stroke (Bagian II)*. Tugas Akhir ini di bawah bimbingan Winarno, S.Si., M.T. dan Deny Arifianto S.Si. Prodi D3 Otomasi Sistem Instrumentasi Departemen Teknik Fakultas Vokasi Universitas Airlangga.

ABSTRAK

Dalam ilmu kesehatan, EMG banyak digunakan untuk membantu tim medis dalam memantau perkembangan kondisi otot pasien pasca stroke ataupun untuk mengetahui adanya gangguan pada otot. Pada Tugas Akhir ini penulis merancang EMG yang mampu merekam serta menyimpan hasil sinyal EMG ke dalam bentuk file untuk membantu tim medis dalam proses pemulihan pasien pasca stroke. File tersebut dapat memudahkan tim medis untuk memantau atau membandingkan kondisi otot pasien yang sekarang dengan yang sebelumnya, serta tidak menutup kemungkinan untuk menjadikan informasi sinyal EMG yang telah tersimpan sebagai referensi proses pemulihan pasien lain. Sistem perekaman sinyal EMG ini menggunakan rangkaian *differensial amplifier* dengan *gain* sebesar 100x, rangkaian *non-inverting* dengan penguatan sebesar 10x, rangkaian *low pass filter* dengan frekuensi *cut off* 500 Hz, rangkaian *notch filter* dengan frekuensi *cut off* 50 Hz dan juga rangkaian *clamper* yang berguna untuk menggeser seluruh nilai agar menjadi positif. Alat yang penyusun rancang menggunakan modul Arduino Uno sebagai pembacaan data dari modul EMG serta mengirimkan data pada LabVIEW. *Software* yang penyusun gunakan dalam menunjang alat yaitu IDE Arduino untuk program pada *board* arduino dan LabVIEW sebagai *user interface* untuk menampilkan serta menyimpan hasil dari perekaman.

Kata kunci: Pasca Stroke, Sistem Perekaman Sinyal EMG, File.

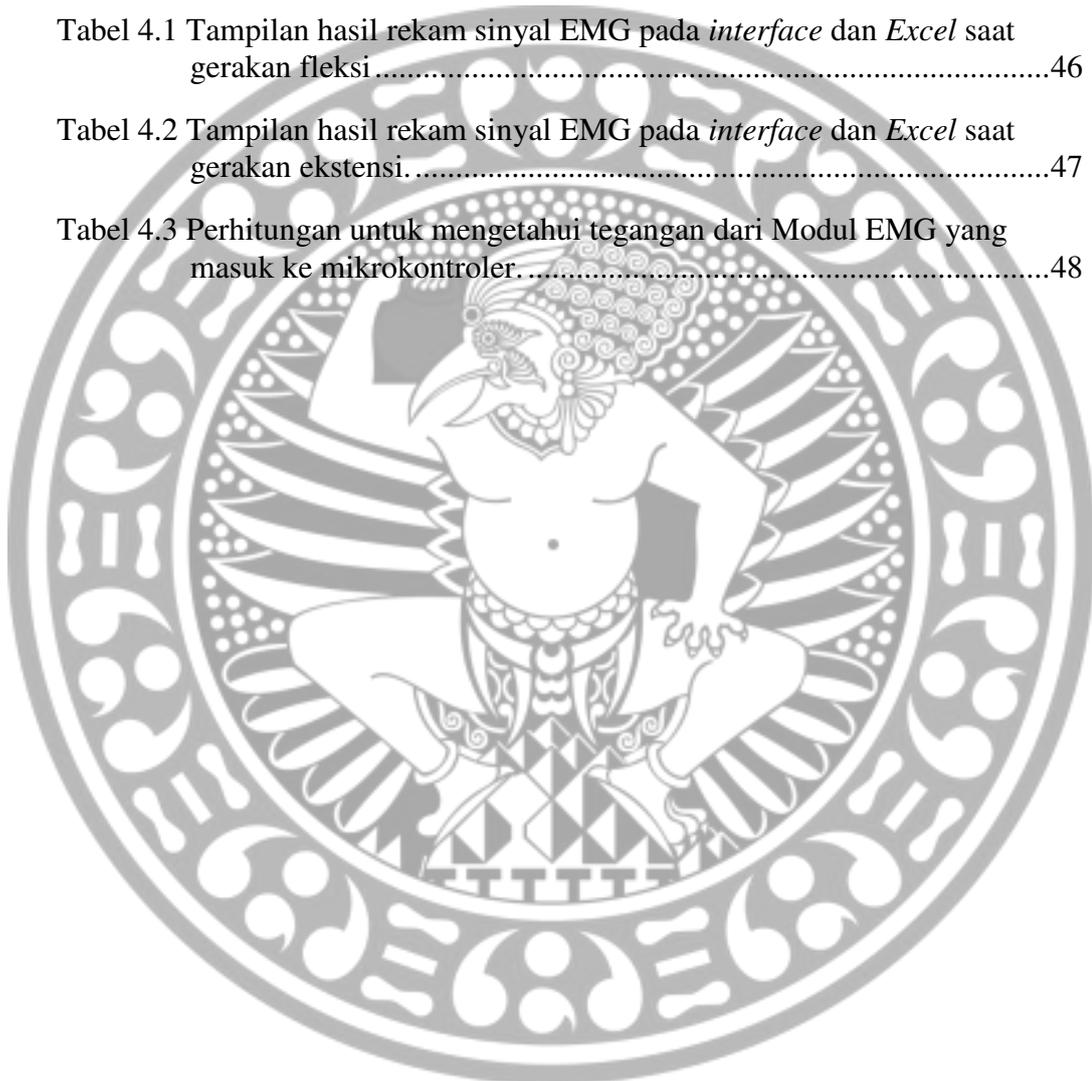
DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
PEDOMAN PENGGUNAAN TUGAS AKHIR.	iv
KATA PENGANTAR.....	v
ABSTRAK.	vii
DAFTAR ISI.....	viii
DAFTAR TABEL	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN.	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Otot.....	4
2.1.1 Prinsip Kerja Otot.....	4
2.1.2 Otot Bisep.....	5
2.2 <i>Electromyograph (EMG)</i>	6
2.3 Elektroda tempel	7
2.4 IDE Arduino	8
2.10 LabVIEW	11

BAB III METODE PENELITIAN	15
3.1 Tempat dan Waktu Penelitian	15
3.2 Bahan dan Alat	15
3.2.1 Bahan-Bahan Penelitian	15
3.2.2 Alat-Alat Penelitian	16
3.3 Prosedur Penelitian.....	16
3.3.1 Tahap Persiapan.....	18
3.3.2 Tahap Perancangan Alat.....	18
3.3.3 Tahap Perwujudan Alat.....	23
3.3.4 Tahap Pengujian Sistem.....	25
3.4 Analisis Data.....	30
BAB IV HASIL DAN PEMBAHASAN.....	31
4.1 Hasil Pembuatan <i>Software</i>	31
4.1.1 Konfigurasi <i>Port</i> Mikrokontroler Arduino Uno.....	31
4.1.2 Program <i>Interface</i> menggunakan LabView.....	34
4.1.3 Pengujian <i>Output</i> Modul EMG.....	44
4.2 Bentuk Sinyal.....	46
BAB V KESIMPULAN DAN SARAN.....	49
5.1 Kesimpulan.....	49
5.2 Saran.....	50
DAFTAR PUSTAKA.....	51
LAMPIRAN.....	52

DAFTAR TABEL

Tabel 2.1 Bentuk Sinyal EMG	7
Tabel 3.1 Pengujian data ADC.....	25
Tabel 3.2 Tampilan sinyal dari osiloskop dan hasil <i>Excel</i>	26
Tabel 4.1 Tampilan hasil rekam sinyal EMG pada <i>interface</i> dan <i>Excel</i> saat gerakan fleksi	46
Tabel 4.2 Tampilan hasil rekam sinyal EMG pada <i>interface</i> dan <i>Excel</i> saat gerakan ekstensi.	47
Tabel 4.3 Perhitungan untuk mengetahui tegangan dari Modul EMG yang masuk ke mikrokontroler.....	48



DAFTAR GAMBAR

Gambar 2.1 Gerak Fleksi dan Ekstensi.	6
Gambar 2.2 <i>Software</i> IDE Arduino	8
Gambar 2.3 Tampilan <i>sketch</i> IDE Arduino	9
Gambar 2.4 <i>Software</i> LabVIEW	11
Gambar 2.5 Bagian <i>Front Panel</i> pada LabVIEW	12
Gambar 2.6 Bagian <i>block diagram</i> pada LabVIEW.	13
Gambar 2.7 Contoh <i>user inrterface</i> pada LabVIEW..	14
Gambar 3.1 Diagram blok prosedur penelitian	17
Gambar 3.2 <i>Flowchart</i> pemrograman IDE Arduino	19
Gambar 3.3 <i>Flowchart</i> pemrograman LabVIEW.....	20
Gambar 3.4 Alat sistem perekaman sinyal EMG	24
Gambar 4.1 Pemrograman IDE Arduino.....	31
Gambar 4.2 Program LabVIEW dengan <i>case structure</i> kondisi <i>true</i>	34
Gambar 4.3 Program LabVIEW dengan <i>case structure</i> kondisi <i>false</i>	34
Gambar 4.4 Komponen <i>VISA Configure Serial</i>	36
Gambar 4.5 Tampilan <i>VISA resource name</i> dan <i>baudrate</i> pada <i>front panel</i>	37
Gambar 4.6 Komponen <i>button start/pause</i>	37
Gambar 4.7 Komponen <i>VISA read</i> dan <i>wire read buffer</i>	38
Gambar 4.8 Program bagian parsing data.	39
Gambar 4.9 Proses pengolahan data.....	40
Gambar 4.10 Program <i>case structure</i> kondisi <i>false</i>	41
Gambar 4.11 Tampilan pada <i>user interface</i> LabVIEW.....	41
Gambar 4.12 Program penyimpanan data ke <i>Excel</i>	42

Gambar 4.13 Tampilan *user interface* pada LabVIEW bagian *front panel*.43
Gambar 4.14 Tampilan perekaman sinyal EMG pada LabVIEW.....44
Gambar 4.15 Tampilan isi penyimpanan *file*.45



DAFTAR LAMPIRAN

Lampiran 1 Rangkaian Sistem Perekaman sinyal EMG

Lampiran 2 Program IDE Arduino

Lampiran 3 Program *Interface* LabVIEW

Lampiran 4 Tampilan / *user interface* pada LabVIEW

Lampiran 5 Penyimpanan berupa *file* di *Excel*



BAB I

PENDAHULUAN

1.1 Latar Belakang

Stroke merupakan penyakit yang menyebabkan penurunan fungsi organ tubuh untuk digerakkan sehingga akan menyebabkan kelumpuhan pada penderitanya. Penyakit stroke ada dua jenis yaitu ringan dan berat. Stroke ringan biasanya hanya akan membuat beberapa bagian tubuh mengalami lumpuh, namun apabila penyakit stroke yang parah maka kelumpuhan bisa dibanyak tempat bahkan seluruh tubuh pun juga bisa. Stroke memerlukan tindakan darurat medis pada masa emasnya (*Golden period*) beberapa jam setelah serangan stroke untuk mencegah supaya kerusakan yang terjadi tidak lebih parah, hal tersebut sangat berarti bagi kondisi kesehatan pasien pasca stroke. Selain itu pasien pasca stroke ini juga menerima pengobatan serta rehabilitasi untuk proses pemulihannya.

Rehabilitasi pasca stroke bertujuan agar penderita dapat hidup mandiri dan produktif kembali. Dalam masa rehabilitasi pasien pasca stroke, otot yang dimiliki belum bisa berfungsi seperti sedia kala, kondisi otot masih lemah sehingga menyebabkan pasien tidak mampu untuk menggerakkan bagian dari anggota tubuhnya. Oleh karena itu diperlukan teknik untuk mengetahui aktivitas sinyal serta kondisi otot menggunakan EMG (*Electromyograph*) agar diketahui perkembangan otot dari pasien pasca stroke sehingga memudahkan tim medis dalam memantau dan mengontrol proses rehabilitasi. Karena seorang pasien pasca stroke hanya dapat merasakan keluhan mampu atau tidaknya menggerakkan

anggota bagian tubuhnya tanpa mengetahui spesifik dari keadaan atau kondisi otot.

Hasil keluaran dari EMG ini berupa informasi sinyal kondisi otot pasien pasca stroke, informasi tersebut penting kerana berpengaruh terhadap proses rehabilitasi. Dari informasi sinyal EMG dapat diketahui bagaimana perkembangan otot dalam proses pemulihan pasien pasca stroke. Sehingga tim medis dapat mengontrol kondisi pasien, mampu menentukan langkah selanjutnya untuk mempercepat proses penyembuhan. Oleh karena itu kami merancang untuk membuat tugas akhir yang berjudul “Rancang Bangun Sistem Perekaman Sinyal EMG Untuk Monitoring Perkembangan Pasien Pasca Stroke”. Dimana alat ini nantinya digunakan untuk merekam serta menyimpan sinyal EMG untuk dapat dijadikan data rekam medis pasien pasca stroke.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka disusun rumusan masalah yang mencakup:

1. Bagaimana proses akuisisi data *signal* EMG dari pasien?
2. Bagaimana desain antarmuka sistem penyimpanan *signal* EMG?

1.3 Batasan Masalah

Dalam pembuatan tugas akhir ini, agar permasalahan tidak meluas maka penulis membuat batasan masalah antara lain:

1. Elektroda yang digunakan adalah jenis kulit atau permukaan (elektroda tempel).
2. Alat ini digunakan pada otot bicep dengan gerakan fleksi dan ekstensi.
3. Penelitian ini dilakukan pada orang normal.

1.4 Tujuan

Tujuan pembuatan tugas akhir ini adalah:

1. Mengetahui proses akuisisi data *signal* EMG dari pasien.
2. Membuat desain antarmuka sistem penyimpanan *signal* EMG

1.5 Manfaat Tugas Akhir

Manfaat tugas akhir ini adalah untuk memudahkan tim medis dalam memantau perkembangan kondisi otot pasien dan menentukan langkah apa yang harus diambil tim medis dengan menyimpan hasil rekaman sinyal EMG dalam bentuk *file*. Sehingga *file* tersebut sewaktu-waktu dapat ditampilkan informasinya, dengan kata lain *file* tersebut bisa menjadi referensi untuk proses pemulihan pasien lain sehingga bisa meminimalisir kesalahan tim medis dalam menentukan langkah selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Otot

Otot adalah alat gerak aktif karena otot memiliki kemampuan untuk berkontraksi. Keadaan otot akan memendek jika melakukan kontraksi dan keadaannya akan memanjang ketika relaksasi. Kontraksi dari otot tersebut dapat menyebabkan tulang yang ada di dekatnya bergerak. Otot memiliki *range* tegangan sebesar 0,4mV sampai 5mV pada frekuensi 20Hz sampai dengan 500 Hz.

Otot sendiri memiliki 3 karakteristik, yaitu:

1. Kontraksibilitas, yaitu kemampuan otot untuk memendek, lebih pendek daripada ukuran semula. Hal ini terjadi ketika otot sedang melakukan kegiatan.
2. Ektensibilitas, yaitu kemampuan otot untuk memanjang, lebih panjang daripada ukuran semula.
3. Elastisitas, yaitu kemampuan otot untuk kembali ke ukuran semula.

2.1.1 Prinsip Kerja Otot

Tulang dapat bergerak karena adanya otot yang bekerja secara kontraksi (berkerut). Otot tersebut akan berkontraksi apabila mendapat rangsangan dari saraf. Pada saat kontraksi otot akan memendek, mengeras serta membesar. Setelah

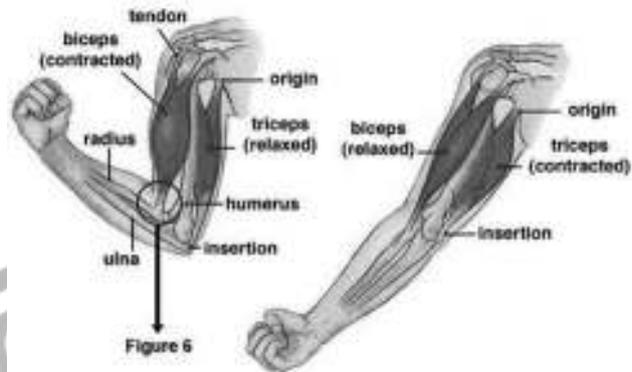
kontraksi, otot akan relaksasi (istirahat) dengan keadaan kebalikan dari kontraksi. Minimal terdapat dua otot untuk dapat menggerakkan tulang.

Menurut cara kerjanya otot dibedakan menjadi dua yaitu otot sinergis dan otot antagonis. Otot sinergis adalah dua otot atau lebih yang tujuan kerjanya sama (kontraksi atau relaksasi bekerja pada waktu yang bersamaan). Contohnya adalah otot-otot yang terdapat di tulang rusuk sama-sama bekerja searah saat pengambilan dan penghembusan nafas. Sedangkan otot antagonis adalah dua otot atau lebih yang tujuan kerjanya berlawanan (kontraksi dan relaksasi bekerja pada waktu yang berlawanan). Salah satu contohnya adalah otot bicep trisep yang menimbulkan gerakan fleksi dan ekstensi.

2.1.2 Otot Bicep

Otot bicep adalah otot yang memiliki dua ujung (dua tendon) yang melekat pada tulang dan terletak di lengan atas bagian depan. Pada otot bicep gerakan fleksi dan ekstensi. Fleksi adalah gerak membengkokkan sedangkan ekstensi adalah gerak meluruskan.

Otot bicep ini termasuk pada golongan otot antagonis karena kerjanya berlawanan. Ketika gerak fleksi otot bicep akan berkontraksi. Begitu pula sebaliknya yang terjadi ketika gerak ekstensi, otot bicep akan relaksasi.



Gambar 2.1 Gerak fleksi dan ekstensi (sumber: <https://learningjust4u.wordpress.com>, 2011)

2.2 *Electromyograph* (EMG)

Elektromiografi adalah teknik untuk mengukur serta merekam aktivitas kelistrikan yang dihasilkan oleh otot. Alat yang digunakan disebut *Electromyograph* dan hasil dari rekaman diagnosis ialah Elektromiogram. EMG membantu untuk membedakan antara kondisi otot di mana masalah di mulai pada otot dan kelemahan otot akibat gangguan saraf, selain itu EMG ini juga dapat digunakan untuk merehabilitasi pasien paska stroke. *Electromyograph* mendeteksi potensial listrik yang dihasilkan oleh sel-sel otot ketika sel-sel ini elektrik atau neurologis diaktifkan, sinyal dapat dianalisis untuk mendeteksi kelainan medis.

Ada dua jenis EMG digunakan secara luas yaitu permukaan EMG dan intramuskular (jarum dan kawat halus) EMG. Pada EMG yang menggunakan

elektrode permukaan cara kerjanya adalah ke dua elektrode permukaan (sebagai sensor) ditempelkan pada bagian otot yang akan di monitor kemudian hasilnya akan ditampilkan di monitor PC. Begitu pula dengan EMG intramuskular, cara kerjanya hampir sama dengan EMG permukaan hanya saja EMG intramuskular ini menggunakan jarum sebagai elektroda atau sensornya.

Tabel 2.1 Bentuk Sinyal EMG (sumber: Nomiyasari, 2011)

Gerak Fleksi / Otot Kontraksi	
Gerak Ekstensi / Otot Relaksasi	

2.3 Elektroda Tempel

Elektroda tempel adalah sensor/transduser yang digunakan untuk mendeteksi keadaan otot pada saat proses pemeriksaan EMG. Elektroda ini ditempelkan pada permukaan kulit pasien pada bagian yang telah ditentukan sehingga mampu menyadap sinyal listrik pada permukaan kulit. Elektroda yang digunakan adalah jenis tempel dengan bahan dari perak klorida (AgCl).

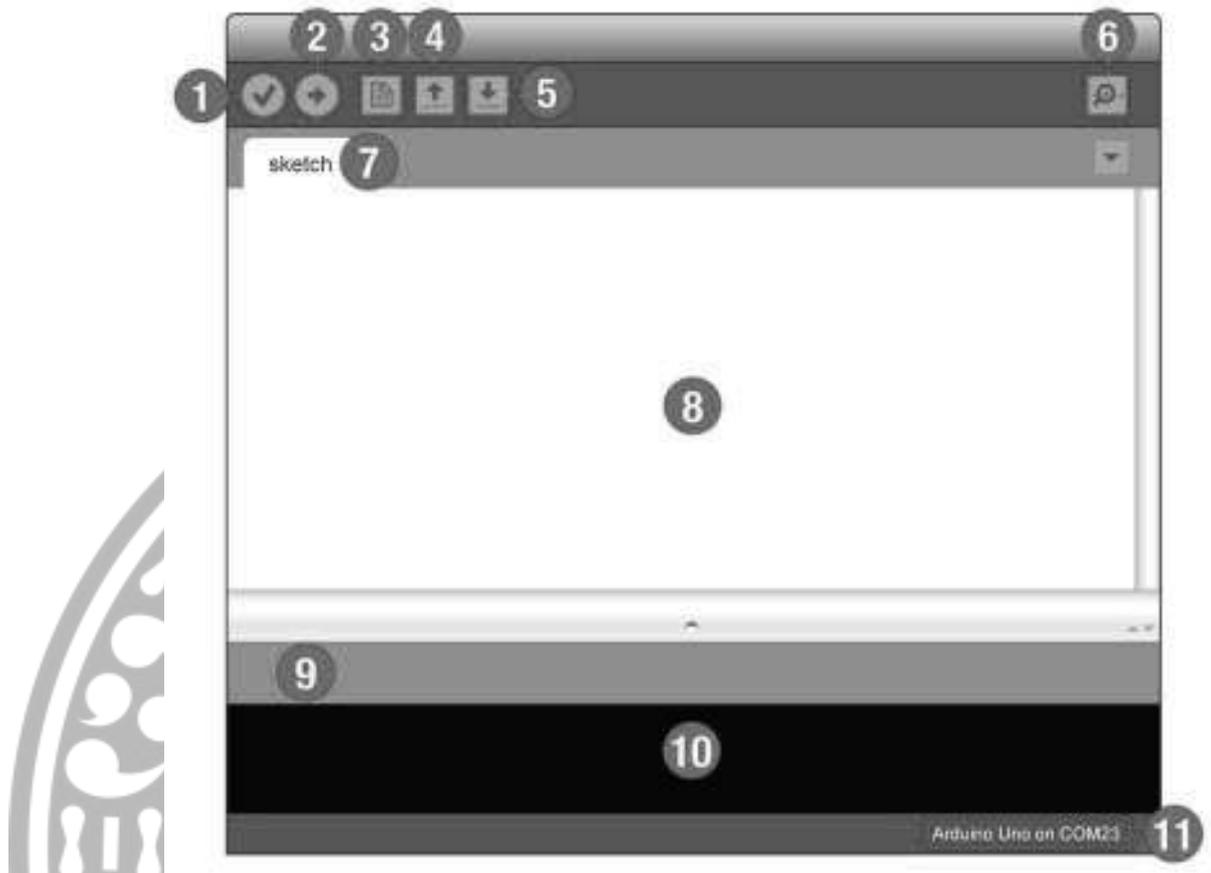
2.4 IDE Arduino

IDE Arduino adalah *software* yang digunakan untuk memprogram mikrokontroler arduino. *Code* program arduino biasa disebut *sketch* dan dibuat menggunakan bahasa pemrograman C. IDE Arduino terdiri dari:

1. Editor program adalah sebuah window yang memungkinkan pengguna menulis dan edit program dalam bahasa *Processing*.
2. *Compiler* adalah sebuah modul yang mengubah kode program (bahasa *Processing*) menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *Processing*. Yang bisa dipahami oleh mikrokontroler adalah kode biner. Itulah sebabnya *compiler* diperlukan dalam hal ini.
3. *Uploader* adalah sebuah modul yang memuat kode biner dari komputer ke dalam memory didalam papan Arduino.



Gambar 2.2 Software IDE Arduino (sumber: <https://blog.arduino.cc/category/software>, 2012)

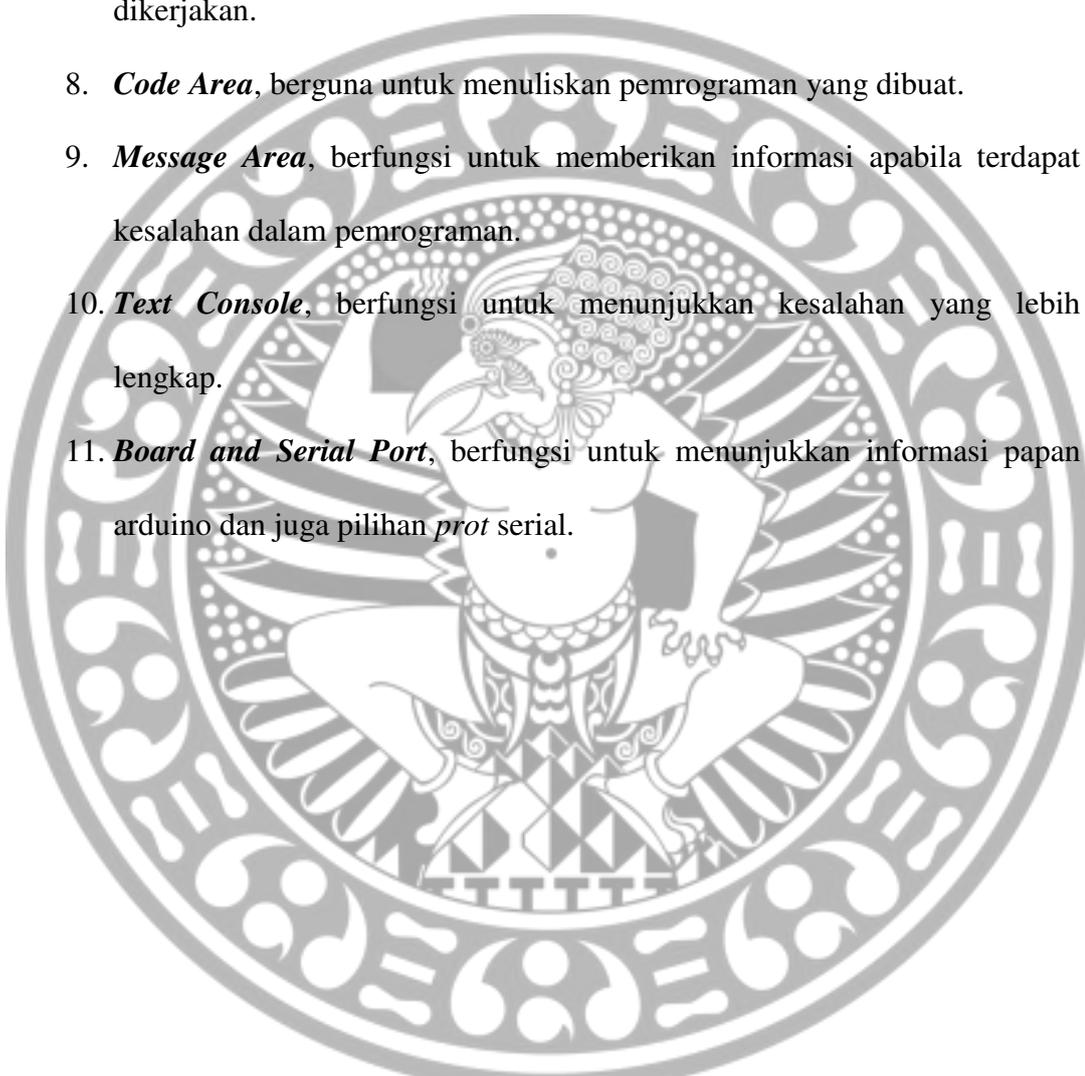


Gambar 2.3 Tampilan *Sketch* IDE Arduino (sumber: <https://learn.sparkfun.com/tutorials>, 2012)

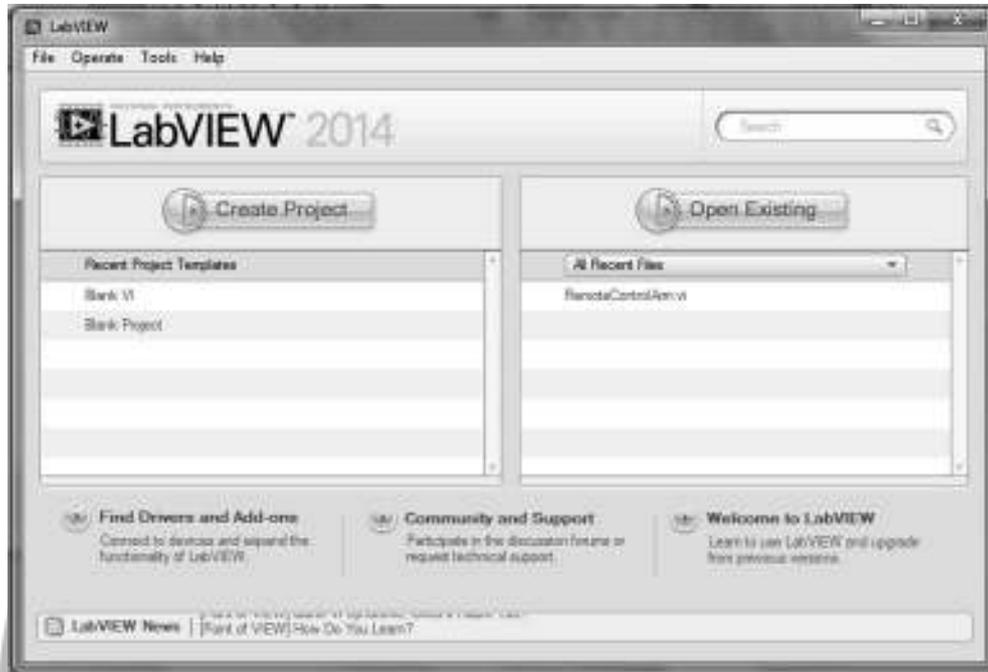
Berikut adalah keterangan dari Gambar 2.3 :

1. **Verify**, berfungsi untuk *Compiles* dan menyetujui kode yang telah dibuat. Serta menangkap apabila terdapat kesalahan pada program yang telah dibuat seperti hilangnya titik koma atau tanda kurung.
2. **Upload**, berfungsi untuk mengirim program ke papan/*board* arduino.
3. **New**, berfungsi untuk membuka lembar pemrograman baru.
4. **Open**, berfungsi untuk membuka program yang sebelumnya telah disimpan.

5. **Save**, berfungsi untuk menyimpan program yang telah dibuat.
6. **Serial Monitor**, berfungsi untuk membuka jendela untuk menampilkan informasi berantai.
7. **Sketch Name**, berfungsi untuk menunjukkan nama sketsa yang lagi dikerjakan.
8. **Code Area**, berguna untuk menuliskan pemrograman yang dibuat.
9. **Message Area**, berfungsi untuk memberikan informasi apabila terdapat kesalahan dalam pemrograman.
10. **Text Console**, berfungsi untuk menunjukkan kesalahan yang lebih lengkap.
11. **Board and Serial Port**, berfungsi untuk menunjukkan informasi papan arduino dan juga pilihan *prot* serial.



2.5 LabVIEW



Gambar 2.4 *Software LabVIEW* (sumber: <http://batechcreativity.blogspot.co.id/2016/04>, 2016)

LabVIEW singkatan dari *Laboratory Virtual Instrumentation Engineering Workbench*. LabVIEW adalah *software* yang khusus digunakan untuk pemrosesan dan visualisasi data dalam bidang akuisisi data, sistem kontrol, sistem instrumentasi serta otomasi industri..

Program LabVIEW disebut dengan *Virtual Instrumen* (VI) karena beberapa tampilan dan operasi pada program LabVIEW menyerupai suatu instrumen seperti *oscilloscope* dan *multimeter*. Setiap VI menggunakan fungsi-fungsi yang memanipulasi *input* dari *user interface* atau sumber lain dan menampilkan informasi tersebut atau memindahkan informasi tersebut ke *file/* komputer lain.

LabVIEW memiliki beberapa kelebihan jika dibandingkan dengan bahasa pemrograman lain seperti:

- Bahasa pemrograman LabVIEW jelas dan mudah dipahami karena berbentuk grafis. Terdiri dari ikon-ikon yang dapat dihubungkan dengan menggunakan *wire*.
- Memiliki ribuan pustaka dan perintah yang dapat digunakan dengan mudah.
- Bersifat modular, dimana sebuah program yang rumit dapat dibuat dengan beberapa program sederhana.
- Dikenal adanya VI dan sub VI, di mana setiap program yang telah dibuat dapat digunakan oleh program lainnya.

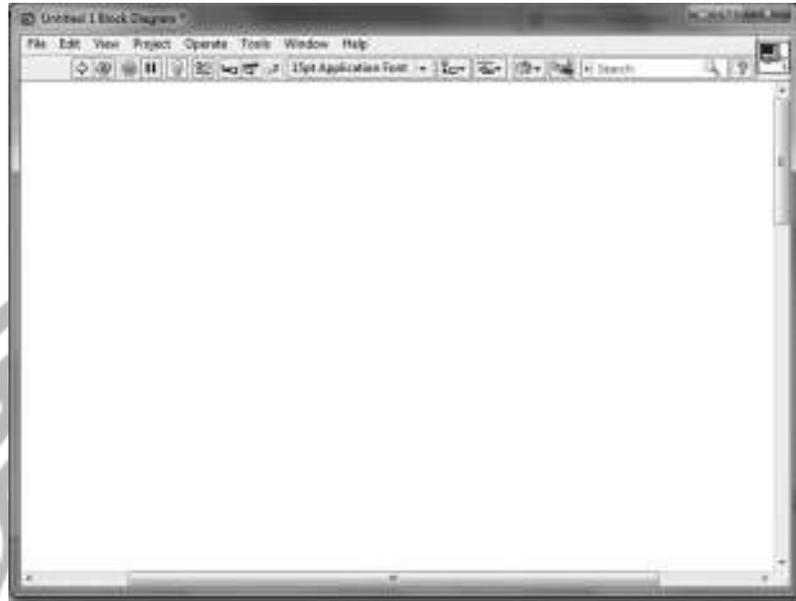
Program LabVIEW terdiri dari dua bagian, yaitu:

1. *Front Panel*, merupakan area pemrograman visual, dimana visualisasi program LabVIEW dapat didesain terlebih dahulu sesuai dengan keinginan pada *user*.



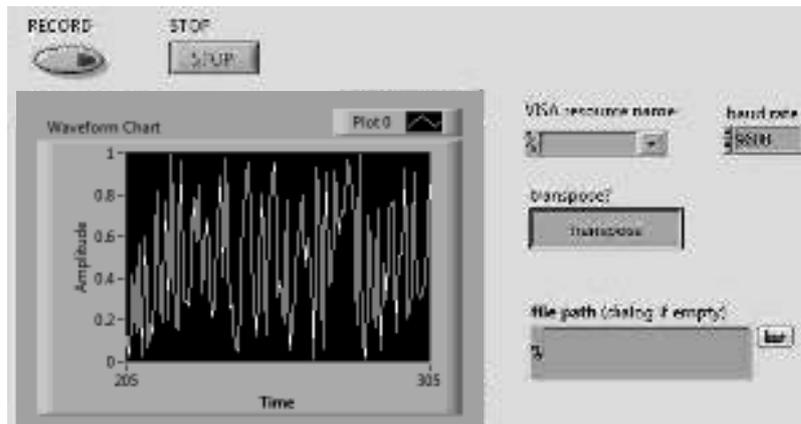
Gambar 2.5 Bagian *Front Panel* pada LabVIEW

2. *Block Diagram*, merupakan pemrograman fungsi, LabVIEW menggunakan ikon-ikon sebagai bahasa pemrogramannya.



Gambar 2.6 Bagian *Block Diagram* pada LabVIEW

Kegunaan aplikasi LabVIEW pada rancang bangun alat kami yaitu untuk memproses data sinyal EMG yang didapat dari Arduino Uno sehingga bisa ditampilkan pada PC dan juga bisa disimpan sebagai data rekam medis pasien. Dengan kata lain, aplikasi LabVIEW adalah aplikasi yang penyusun manfaatkan sebagai *user interface*. Komponen-komponen LabVIEW yang digunakan kurang lebih terdapat pada Gambar 2.7.

Gambar 2.7 Contoh *user interface* LabVIEW

Record Button : *Button* yang digunakan untuk memulai/berhenti sejenak pada saat proses perekaman sinyal EMG.

Waveform Chart : Komponen yang digunakan sebagai *display* sinyal EMG.

VISA Resource Name : Komponen yang digunakan untuk inisialisasi komunikasi serial dengan mikrokontroler.

File Path : Komponen yang digunakan untuk menentukan tempat penyimpanan *file*.

Stop : *Button* yang digunakan untuk mengakhiri proses perekaman.

BAB III

METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian

Perancangan dan pembuatan alat ini dilakukan di Laboratorium Elektronika, Program Studi D3 Otomasi Sistem Instrumentasi, Departemen Fisika, Fakultas Sains dan Teknologi Universitas Airlangga selama kurang lebih 4 bulan dimulai dari bulan April 2016 sampai Juli 2016.

3.2 Bahan dan Alat

3.2.1 Bahan-Bahan Penelitian

Bahan-bahan yang diperlukan dalam penelitian ini adalah sebagai berikut:

1. PCB
2. *Breadboard*
3. Kapasitor
4. Resistor
5. *Power Supply*
6. IC AD620
7. IC CA3140EZ
8. IC LF356
9. Elektroda tempel
10. Baterai Li-Po (*lithium-polimer*)

3.2.2 Alat-Alat Penelitian

Alat yang diperlukan dalam penelitian ini adalah sebagai berikut:

Perangkat keras (*Hardware*):

1. Modul Mikrokontroler Arduino Uno
2. *Oscilloscope*
3. *Signal Generator*
4. *Digital Multimeter*
5. *PC (Personal Computer) / Laptop*

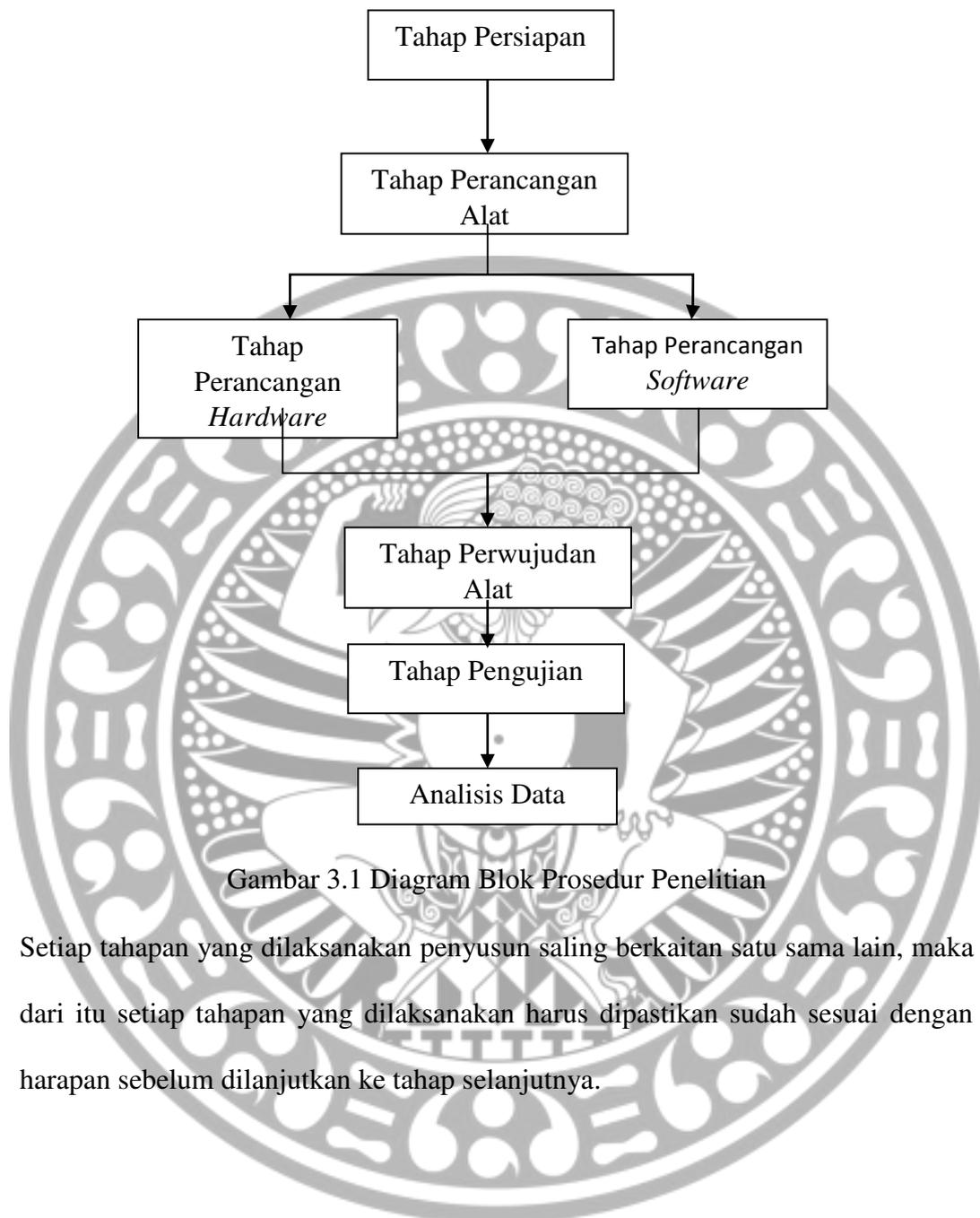
Perangkat Lunak (*Software*):

1. *Eagle*
2. IDE Arduino
3. LabVIEW
4. Windows 7

3.3 Prosedur Penelitian

Prosedur penelitian yang dilakukan terdiri dari lima tahapan, yaitu:

1. Tahap Persiapan
2. Tahap Perancangan Alat
3. Tahap Perwujudan Alat
4. Tahap Pengujian Sistem
5. Analisis Data.



Gambar 3.1 Diagram Blok Prosedur Penelitian

Setiap tahapan yang dilaksanakan penyusun saling berkaitan satu sama lain, maka dari itu setiap tahapan yang dilaksanakan harus dipastikan sudah sesuai dengan harapan sebelum dilanjutkan ke tahap selanjutnya.

3.3.1 Tahap Persiapan

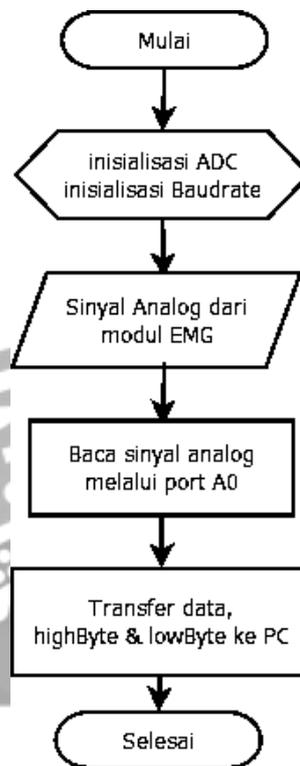
Tahap persiapan merupakan tahapan awal dalam melakukan penelitian, pada tahap ini penulis melakukan studi literatur dengan mencari berbagai acuan baik melalui artikel, buku, jurnal, skripsi maupun tugas akhir dengan narasumber yang jelas dan terpercaya dengan tujuan untuk melengkapi literatur mengenai penelitian ini.

3.3.2 Tahap Perancangan Alat

Tahap perancangan alat dibagi menjadi dua, yakni tahap perancangan *hardware* dan tahap pemrograman *software*. Tahap Perancangan *hardware* meliputi pembuatan modul EMG yang terdiri dari rangkaian, *differensial amplifier*, *non inverting*, *low pass filter*, *notch filter*, dan *clamper*. Tahap perancangan *hardware* lebih detail dijelaskan pada laporan Bagian I. Sedangkan tahap pemrograman *software* menggunakan IDE Arduino dan LabVIEW.

a. Tahap Pemrograman *Software*

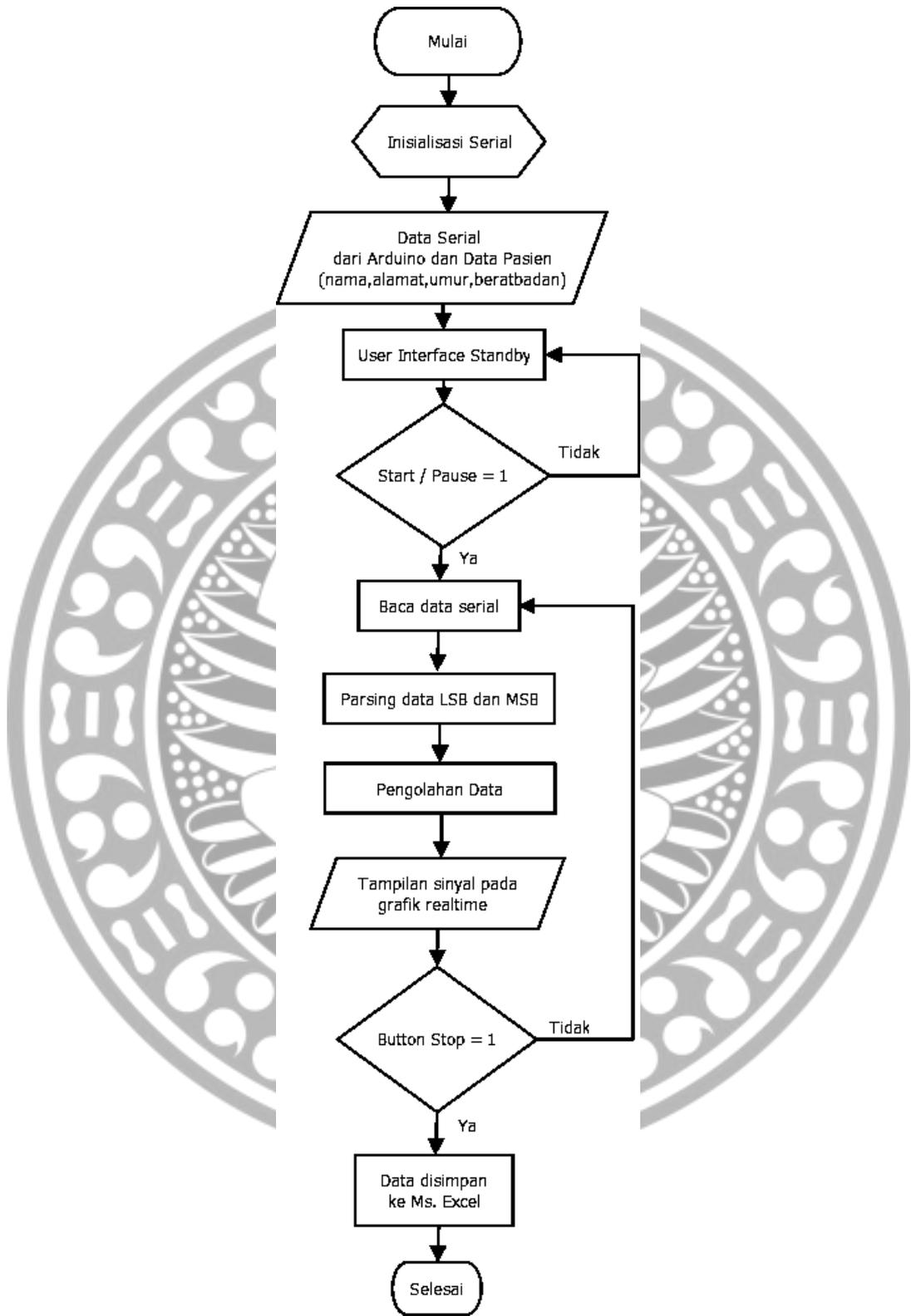
Tahap pemrograman *software* meliputi pembuatan program untuk *hardware* yang sebelumnya telah dibuat. Aplikasi yang digunakan yaitu LabVIEW dan IDE Arduino. IDE Arduino adalah aplikasi yang digunakan untuk mentransfer data sinyal EMG ke PC untuk diproses oleh LabVIEW. LabVIEW pada PC mengolah sinyal EMG sedemikian rupa untuk bisa ditampilkan pada monitor dan juga disimpan ke dalam bentuk *file*. Dengan kata lain, LabVIEW pada rancang bangun alat berperan sebagai aplikasi *interface*



Gambar 3.2 *Flowchart* Pemrograman IDE Arduino

Gambar 3.2 merupakan *flowchart* dari pemrograman IDE Arduino. Dari gambar tersebut dapat diuraikan langkah kerja sebagai berikut:

- a. Mulai merupakan langkah awal untuk menjalankan program IDE Arduino
- b. Inisialisasi ADC dan *Baudrate* merupakan proses dimana program melakukan inisialisasi *port* analog yang digunakan beserta *baudrate*.
- c. Sinyal Analog dari modul EMG merupakan *input* bagi program IDE Arduino. *Input* berupa sinyal analog.
- d. Baca sinyal analog melalui *port* A0 merupakan proses pembacaan data yang telah diterima oleh arduino dari *input*.
- e. Transfer data *highByte* & *lowByte* ke PC merupakan proses untuk mengirimkan data dari arduino ke PC (khususnya program LabVIEW).
- f. Selesai, merupakan langkah berakhirnya program dari IDE Arduino.



Gambar 3.3 Flowchart Pemrograman LabVIEW

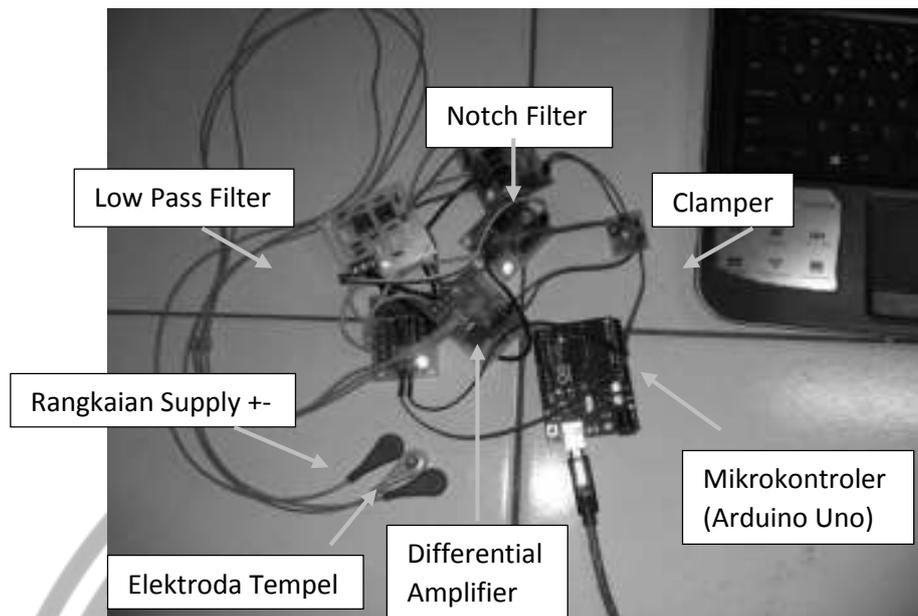
Gambar 3.3 merupakan *flowchart* dari pemrograman LabVIEW. Dari gambar tersebut dapat diuraikan langkah kerja sebagai berikut:

- a. Mulai adalah langkah awal program LabVIEW dimulai..
- b. Inisialisasi serial berguna untuk menginisialisasi serial pada LabVIEW dengan menggunakan komponen *VISA Configure Serial Port*. Memilih *COM Port* serta *baudrate* sesuai dengan mikrokontroler yang digunakan.
- c. Data serial *Input* dari program LabVIEW ini ialah data serial yang telah diperoleh dari arduino. Selain itu *input* program ini ialah data diri pasien meliputi nama, alamat, umur serta berat badan.
- d. *User interface standby* adalah proses dimana *interface* siap digunakan tetapi tidak ada data yang terekam.
- e. *Start/Pause = 1* adalah kondisi untuk *button start/pause*. Apabila *button start/pause* tidak di-klik maka program akan kembali ke proses *user interface standby*. Tetapi apabila *button start/pause* di-klik maka program akan mengeksekusi lanjutan untuk proses pembacaan data.
- f. Baca data serial adalah proses LabVIEW membaca data serial menggunakan komponen *VISA Read* dan disimpan pada *readbuffer*.
- g. Parsing data LSB dan MSB adalah proses memecah data yang masuk menjadi 2 bagian yaitu *highByte* (MSB) dan *lowByte* (LSB) agar data dari ADC arduino dapat diolah pada program LabVIEW.

- h. Pengolahan data adalah proses dimana data diolah untuk ditampilkan pada *waveformchart*. Pada proses ini terdapat 2 kemungkinan proses, pertama adalah menggunakan program yang ada di *case true* dan kedua adalah menggunakan program yang ada di *case false*.
- i. Tampilan sinyal pada grafik *realtime*. *Output* dari program ini adalah menampilkan data yang telah diproses pada *display waveformchart* berupa bentuk sinyal yang *continue (realtime)*
- j. *Button Stop = 1* adalah kondisi apabila *button stop* tidak di-klik maka program akan terus mengeksekusi mulai dari pembacaan data hingga menampilkan data. Tetapi jika *button stop* di-klik maka program pembacaan data hingga penampilannya akan berhenti dan berlanjut ke proses penyimpanan.
- k. Data disimpan ke *Ms.Excel* adalah proses dimana data yang telah terkumpul berisikan identitas pasien, waktu dan tanggal perekaman serta data ADC sinyal otot pasien semua tersimpan dalam *Ms.Excel* dengan menggunakan komponen *Write to Spreadsheet* pada LabVIEW.
- l. Selesai adalah tanda berakhirnya program LabVIEW ini.

3.3.3 Tahap Perwujudan alat

Pada tahap perwujudan sistem perekaman sinyal EMG digunakan elektroda tempel sebagai sensor pendeteksi sinyal listrik pada otot. Karena sinyal listrik yang dihasilkan aktivitas otot sangat kecil maka dibutuhkan rangkaian penguat instrumentasi yang terdiri dari rangkaian *differential amplifier* dan *non inverting* dengan penguatan masing-masing sebesar 100x dan 10x sehingga total penguatan adalah 1000x . Setelah sinyal dikuatkan maka sinyal diteruskan ke beberapa rangkaian filter yaitu rangkaian *low pass filter* yang berguna untuk meloloskan frekuensi hanya sampai 500 Hz, lebih dari frekuensi tersebut maka sinyal akan mulai melemah. *notch filter* berguna untuk menghilangkan frekuensi 50 Hz yang berasal dari jala-jala PLN maupun radiasi yang ada dalam tubuh manusia dan juga rangkaian *clamper* yang berguna untuk menggeser nilai negatif menjadi positif agar terbaca oleh ADC pada mikrokontroler. Selanjutnya data sinyal EMG dikonversi ke bentuk sinyal digital oleh ADC pada Arduino Uno untuk bisa diproses di LabVIEW pada PC. LabVIEW berperan sebagai aplikasi *interface* sehingga proses perekaman dan penyimpanan sinyal EMG dapat dikontrol melalui LabVIEW. *File* data sinyal EMG yang disimpan berformat Ms. Exel (.xls).



Gambar 3.4 Alat Sistem Perekaman Sinyal EMG

Dari gambar 3.4 dapat diketahui bahwa alat (*hardware*) yang telah dibuat oleh penyusun memiliki 3 elektroda dan juga terdapat sebuah kotak, dimana di dalam kotak ini berisikan Arduino Uno, rangkaian *differensial amplifier*, rangkaian *low pass filter*, *notch filter*, *clamper*.

3.3.4 Tahap Pengujian Sistem

Tahap pengujian bagian *software* ini adalah terdiri dari:

a. Pengujian ADC

Pengujian ADC dilakukan untuk mengetahui kesempurnaan kinerja ADC yang ada pada mikrokontroler. Pengujian ini dilakukan dengan menggunakan resistor variabel dengan *input* tegangan DC antara 0V - 5V.

Tabel 3.1 Pengujian Data ADC

Tegangan (V)	Nilai ADC (bit)
0	0
0,5	101
1	204
1,5	307
2	410
2,5	513
3	616
3,5	719
4	818
4,5	922
5	1023

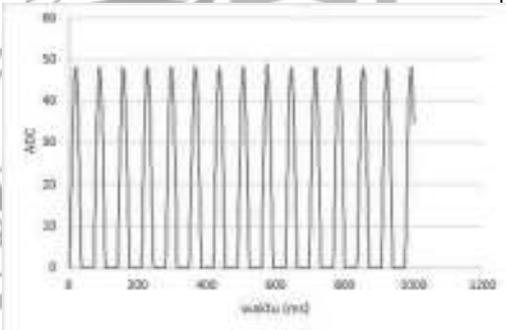
Dari tabel dapat diketahui bahwa ADC yang digunakan ialah 10 bit. Karena ketika diberi tegangan maksimal yaitu 5 volt nilai ADC ialah 1023. Dari tabel juga dapat diambil kesimpulan bahwa ADC pada mikrokontroler bekerja sesuai. Dengan *input* 0,5 volt nilai ADC yang terbaca ialah 101. Jika 101 dimasukkan kedalam rumus konversi bit ke tegangan hasilnya 0,49. Dari hasil masing-masing nilai ADC

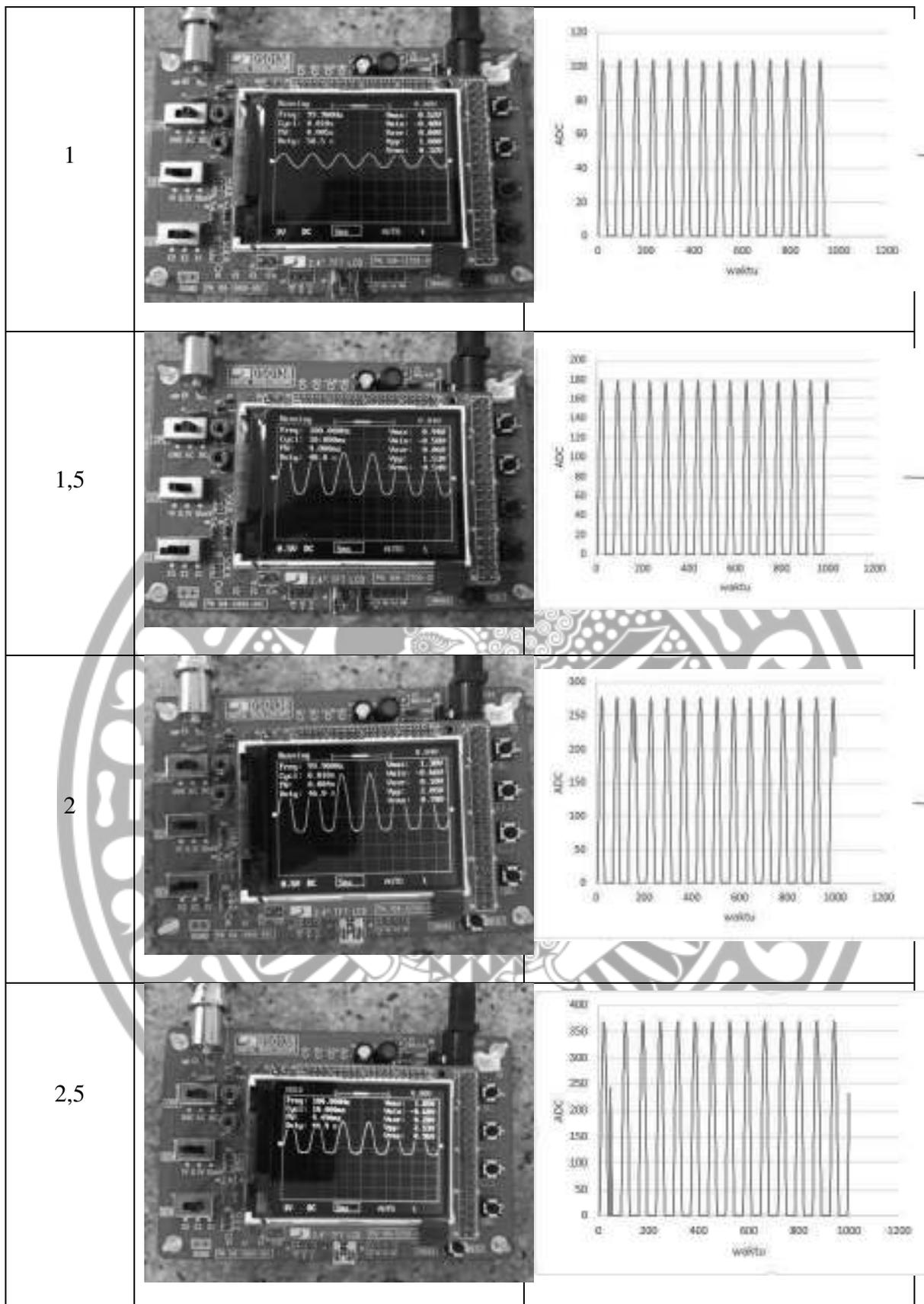
nilainya mendekati dengan tegangan *input* apabila dimasukkan dalam rumus konversi.

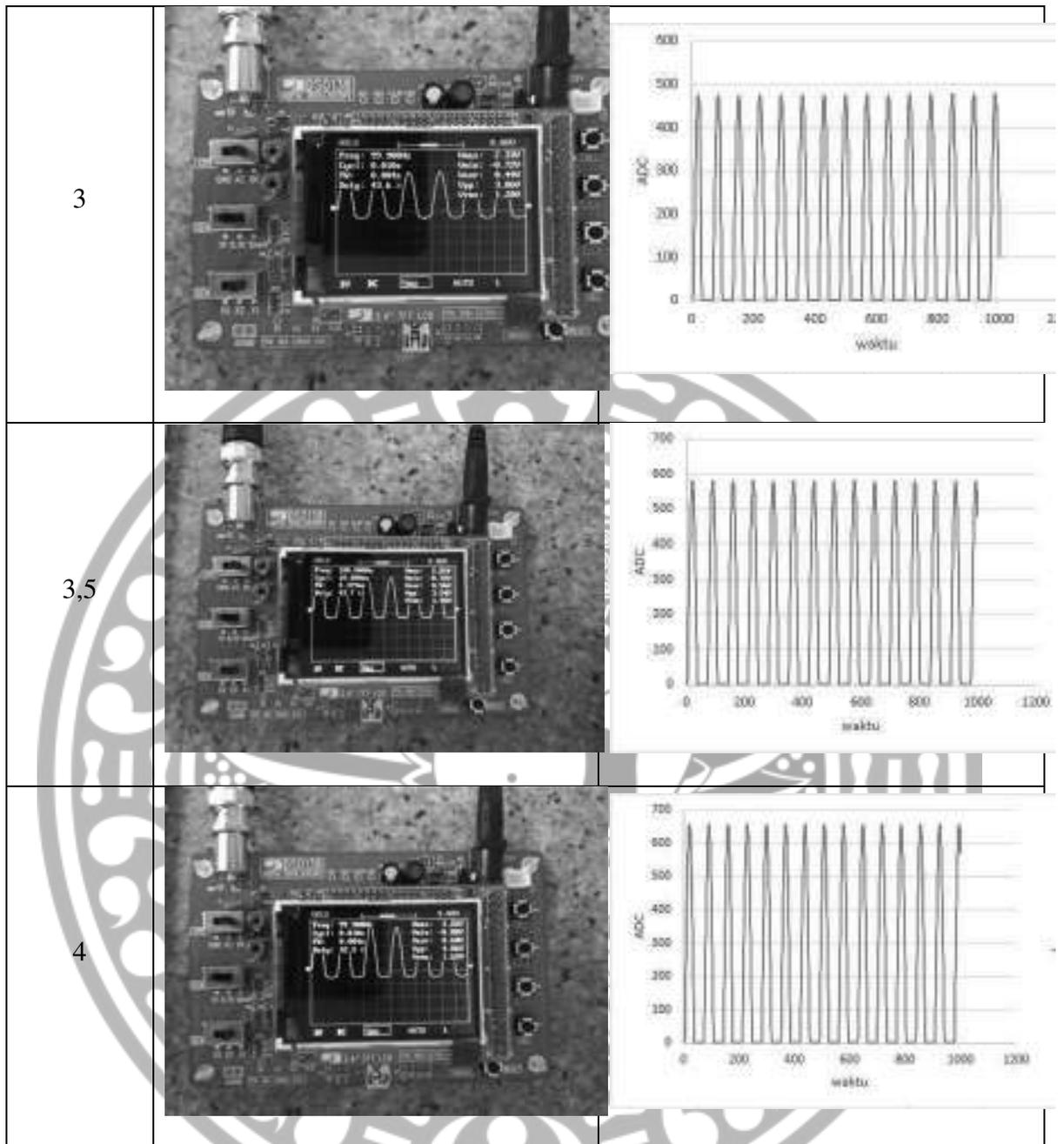
b. Pengujian Sinyal dengan sinyal yang telah teridentifikasi

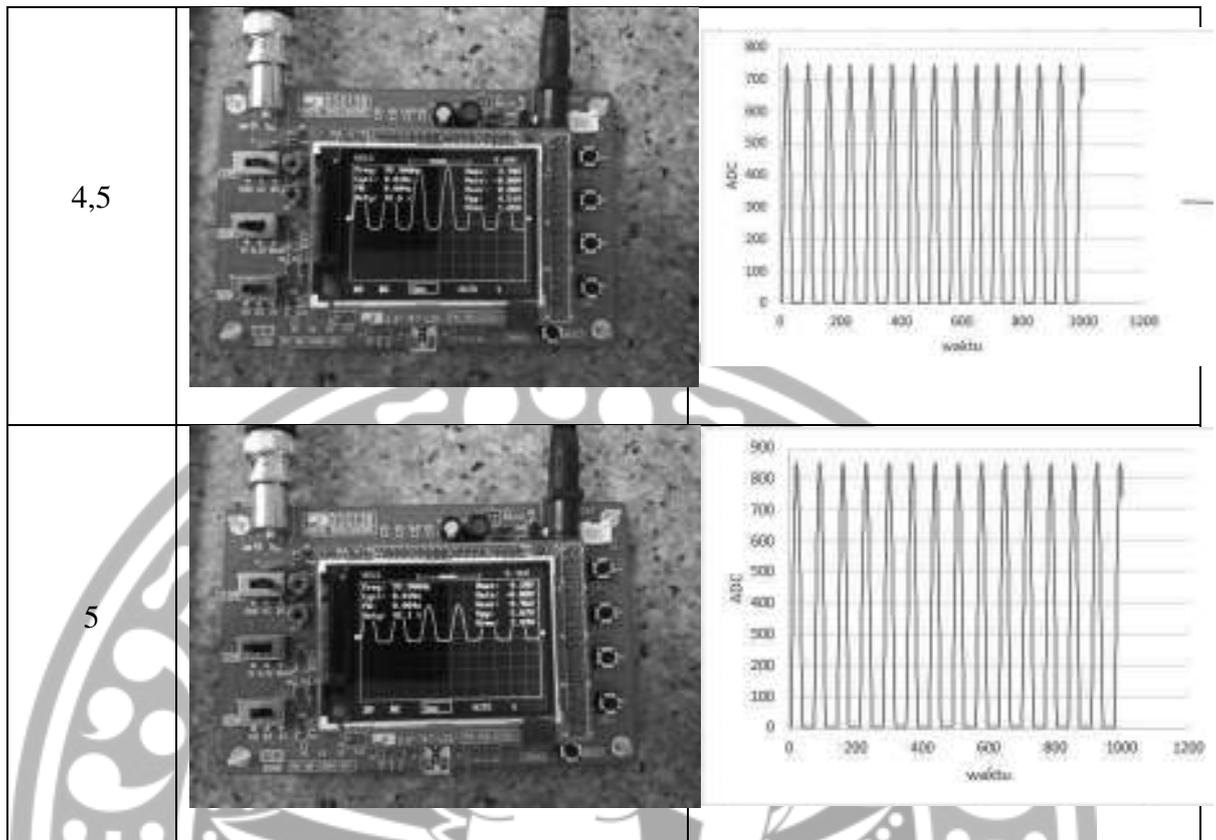
Pada pengujian ini hampir mirip dengan pengujian ADC yang sebelumnya telah dilakukan, hanya saja penelitian ini menggunakan frekuensi konstan sebesar 100Hz dari *signal generator* dan *input* tegangan dari resistor variabel dengan mengubah-ubah *range* tegangan dari 0V – 5V menggunakan tegangan DC.

Tabel 3.2 Tampilan sinyal dari Osiloskop dan Hasil *Excel*

Input Tegangan (Volt)	Sinyal pada Osiloskop	Sinyal pada penyimpanan di Excel
0,5		







Dari hasil tersebut diketahui bahwa bentuk sinyal yang terlihat pada *Ms.Excel* terpotong karena pada saat pengambilan data tegangan yang diberikan pada ADC berupa tegangan DC. Sedangkan pada osiloskop tegangan yang didapat berupa tegangan AC dari signal generator yang terhubung pada catu daya 220 volt.

c. Analisis *software*

Analisis *software* ini meliputi pengujian respon *hardware* dari program yang telah dikirim oleh mikrokontroler serta program untuk menampilkan pada LabVIEW. Pengujian ini dilakukan untuk mengetahui kesesuaian antara program dengan hasil yang diinginkan yaitu dapat membaca serta menampilkan hasil sinyal otot.

d. Pengujian *output* modul EMG

Pada pengujian *output* modul EMG ini adalah menganalisis apakah sinyal yang dihasilkan masih dalam *range* sinyal pada literatur, serta menganalisis data apa saja yang tersimpan dalam *file*.

3.4 Analisis Data

Analisis data dalam penelitian “Rancang Bangun Sistem Perekaman Sinyal EMG untuk Monitoring Perkembangan Pasien Pasca Stroke” BAGIAN II ini adalah :

1. Analisis program IDE Arduino
2. Analisis program LabVIEW
3. Pengujian *Output* Modul EMG

Pengambilan data tersebut dilaksanakan untuk mengetahui apakah *hardware* dapat beroperasi sesuai dengan yang diinginkan yaitu mampu merekam serta menyimpan hasil keluaran sinyal dari modul EMG.

BAB IV**HASIL DAN PEMBAHASAN****4.1 Hasil Pembuatan Software**

Agar *hardware* yang dibuat mampu beroperasi sesuai dengan yang dituju maka dibuatlah *software* (perangkat lunak) untuk akuisisi data *output* sinyal EMG yang mampu merekam dan menyimpan.

4.1.1 Konfigurasi Port Mikrokontroler Arduino Uno

Mikrokontroler Arduino Uno memiliki *port* analog dan *port* digital, dalam penelitian ini *port* dalam arduino yang digunakan hanya satu yaitu *port* analog A0. Dimana *port* ini berguna untuk mendapat *input* yaitu dari modul EMG berupa data sinyal analog.



```
arduino | Arduino 1.0.5
File Edit Sketch Tools Help
arduino$
void setup() {
  Serial.begin(9600);
}

void loop() {
  int adc = analogRead(A0);
  Serial.write(byte(adc));
  Serial.write(lineByte(adc));
  delay(10);
}
```

Gambar 4.1 Pemrograman IDE arduino

```
Void setup ( ) {  
  Serial.begin (9600);  
}
```

Program tersebut digunakan untuk menginisialisasi *baudrate*. *Baudrate* yang digunakan pada penelitian ini ialah 9600. Kegunaan *baudrate* adalah untuk menentukan kecepatan data yang dikirim. *Input* dari program ini ialah sinyal analog dari modul EMG yang dihubungkan dengan *port A0*.

```
Void loop ( ) {  
  int adc = analogRead (A0); // variabel adc merupakan data ADC  
  dari port A0
```

Program tersebut digunakan untuk membaca data analog yang berasal dari modul EMG berupa sinyal.

```
Serial.write (highByte(adc));  
Serial.write (lowByte(adc));  
delay (10);
```

Program tersebut digunakan untuk pengiriman data ADC ke *PC*. Terjadi dua kali pengiriman data yaitu *highByte* (MSB) dan *lowByte* (LSB)

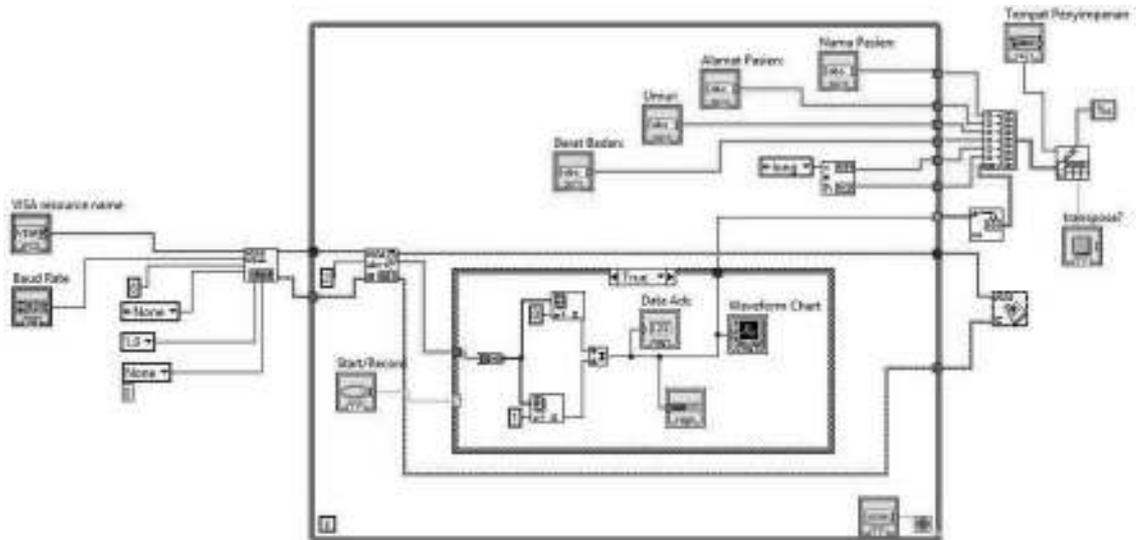
Perintah *Serial.write (highByte(adc))* ialah memerintahkan untuk mengirim data MSB ke LabVIEW.

Perintah *Serial.write (lowByte(adc))* ialah memerintahkan untuk mengirim data LSB ke LabVIEW.

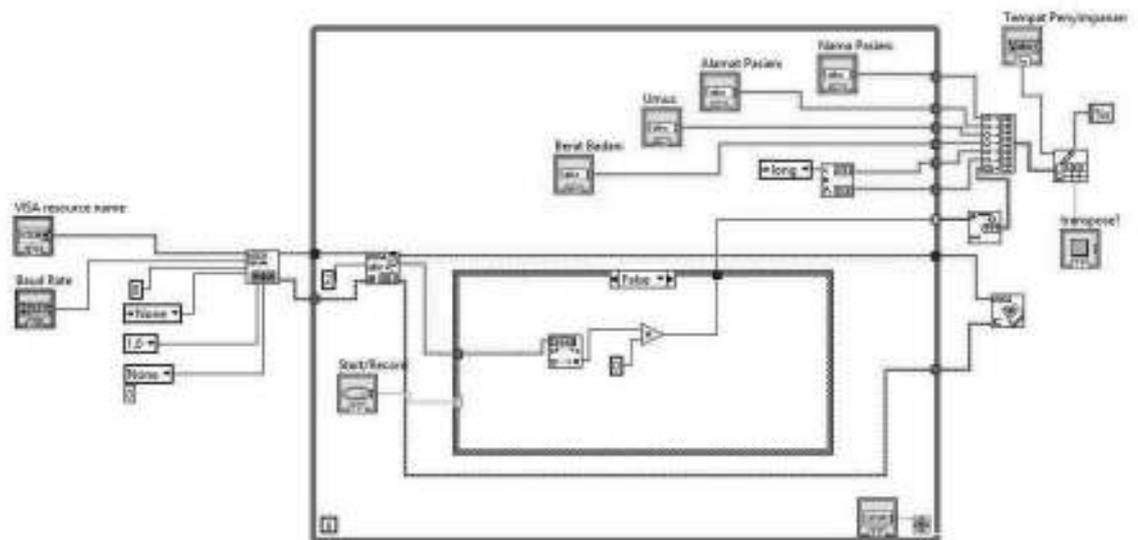
Program dari *void loop* hingga *delay (10)* adalah blok dimana program dieksekusi secara terus menerus. Apabila program sudah sampai akhir blok maka akan dilanjutkan dengan mengulang eksekusi dari awal.

Dapat diketahui dari program tersebut digunakan untuk proses pengambilan data sinyal dari modul EMG, 2 elektroda yang telah terpasang pada bagian lengan (otot bicep) dan 1 elektroda yang terpasang pada bagian dekat siku (digunakan untuk referensi) mendeteksi sinyal yang terdapat pada otot kemudian sinyal tersebut diproses oleh modul EMG melalui beberapa rangkaian yang telah dibuat yaitu *differnsial amplifier, non inverting, low pass filter, notch filter* dan *clamper*. Setelah melalui beberapa rangkaian, *output* dari sistem inilah yang menjadi *input / masukan* bagi arduino untuk kemudian diproses sehingga mampu memberikan keluaran yang dapat diproses lagi oleh LabVIEW guna menyempurnakan akuisisi data.

4.1.2 Program *Interface* menggunakan LabVIEW



Gambar 4.2 Program LabVIEW dengan *case structure* kondisi *True*



Gambar 4.3 Program LabVIEW dengan *case structure* kondisi *False*

Berikut adalah penjelasan dari program LabVIEW sebagai *user interface* :

Pada program Lab View menggunakan 2 macam blok yaitu *while loop* dan *case structure*. *While loop* berguna untuk megulang/*looping* program-program yang ada didalamnya tanpa harus menentukan berapa kali proses pengulangan harus di lakukan. Dengan menekan *button stop* maka program dalam *while loop* akan berhenti dieksekusi. Komponen-komponen yang ada di dalam *while loop* adalah:

- *VISA Read*
- *String Control* (Nama Pasien)
- *String Control* (umur)
- *String Control* (berat badan)
- *Get Date/Time String*
- *String Control* (Alamat)
- *Button start/pause*
- *Case Structure*

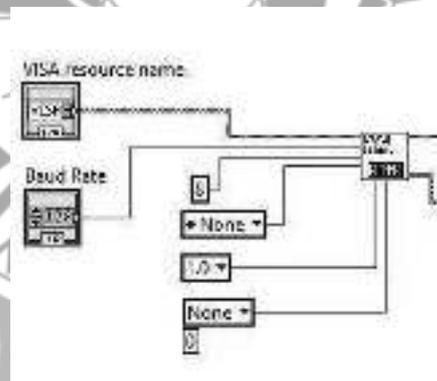
Sedangkan komponen yang berada di luar *while loop* ialah:

- *VISA Configure Serial Port*
- *Number to decimal string*
- *Write to spreadsheet*
- *VISA Close*
- *Build array*

Case structure berfungsi sebagai pemberian kondisi, jadi dalam *case structure* terdapat 2 kondisi yaitu *true* atau *false*. Dalam masing-masing kondisi isi programnya berbeda tetapi tidak terkait antara satu dengan yang lain. Karena *case structure* berada didalam *while loop* sehingga program-program yang ada di dalamnya otomatis juga akan mengulang sesuai dengan *input* kondisi yang diterima oleh *while loop*. Komponen yang berada dalam *case structure* ialah:

- *String To byte array*
- *Index array*
- *Join Number*
- *Waveform Chart*
- *Decimal string to number*
- *Multiply*
- *Horizontal Progress Bar*
- *Numeric Indicator*

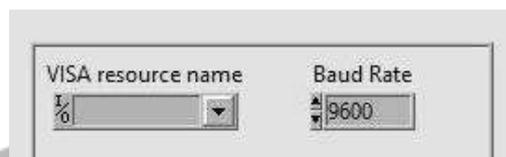
Pada saat *button start/pause* ditekan maka *interface* mulai mengeksekusi program dengan langkah awal melakukan inisialisasi.



Gambar 4.4 Komponen VISA *Configure Serial*

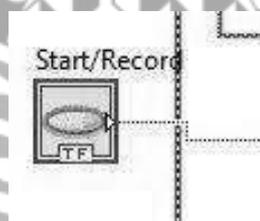
Pada gambar 4.4 adalah komponen VISA *Configure Serial*. Komponen ini berfungsi untuk inisialisasi komunikasi serial pada LabVIEW. Komponen ini terdiri dari VISA *Resource name* (untuk memilih COM *Port* yang digunakan), *baudrate* (untuk memilih *baudrate* yang digunakan) dan pemilihan *baudrate*

disesuaikan pada mikrokontroler yang digunakan yaitu sebesar 9600. Serta terdapat data bit sebesar 8 yang artinya pada saat sekali pengiriman data, data yang masuk ialah sebanyak 8 bit.



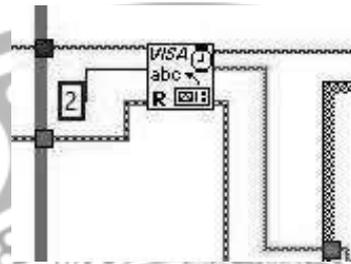
Gambar 4.5 Tampilan VISA *resource name* dan *Baudrate* pada *Front Panel*

Setelah proses inisialisasi maka proses selanjutnya ialah *interface standby* atau siap untuk digunakan. Saat proses *standby* ini *interface* tidak merekam apapun sehingga tidak ada perubahan tampilan pada *display waveform chart*. Apabila *button start/pause* = 0 (tidak di-klik) maka *interface* akan tetap *standby*, tetapi apabila *button start/pause* = 1 (di-klik) maka program akan mengeksekusi bagian proses pembacaan data serial. *Button start/pause* inilah yang memberikan kondisi pada *case structure* untuk memilih *case* bagian mana yang akan di eksekusi.



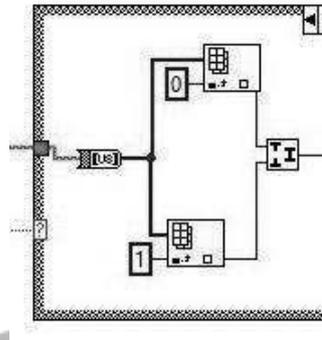
Gambar 4.6 Komponen *button start/pause*

Proses pembacaan data serial dilakukan oleh komponen *VISA Read*, dimana seluruh data yang dibaca kemudian disimpan pada *read buffer* untuk kemudian dihubungkan pada *case structure* guna untuk proses selanjutnya yaitu parsing data.



Gambar 4.7 Komponen *VISA Read* dan *wire Read Buffer*

Dari gambar 4.7 dapat diketahui bahwa *wire* berwarna *pink* ialah *wire* yang menghubungkan antara *VISA Read* dengan *Case Structure*, *wire* berwarna *pink* menunjukkan tipe data yang dibawa ialah tipe data *string*. Karena pada awal mula data yang masuk ialah jenis *string*. Terdapat konstanta sebesar 2 yang artinya sekali pengiriman data terjadi 2 gelombang pemecahan yaitu data LSB dan juga MSB.



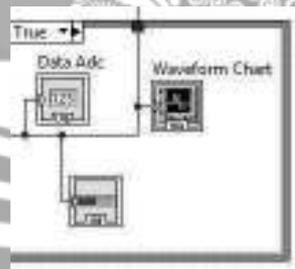
Gambar 4.8 Program bagian parsing data

Pada proses parsing data dilakukan di dalam blok *while loop* dan berada pada blok *case structure* bagian kondisi *true* atau *button stop/pause = 1*. Fungsi dari parsing data ini adalah untuk memecahkan data dan mengurutkan data supaya tidak acak. *Case structure* mendapat masukan data *string* dari *read buffer* yang kemudian dihubungkan dengan komponen *string to byte array*. Kegunaan komponen tersebut adalah untuk merubah tipe data *string* menjadi *array* supaya dapat dimasukkan dalam komponen *index array* dan berada sesuai pada posisinya masing-masing. Yaitu *index array* dengan konstanta 0 ialah *array* untuk data LSB sedangkan *index array* dengan konstanta 1 ialah *array* untuk data MSB.

Terdapat data MSB dan juga LSB karena ADC yang digunakan pada arduino ialah sebesar 10 bit, sedangkan komunikasi serial proses pengiriman data hanya bisa 8 bit. Sehingga dari 10 bit harus dipecah menjadi 2 bagian, bagian pertama pengiriman data utuh sebesar 8 bit kemudian pengiriman keduanya tetap 8 bit namun yang terisi bagiannya hanya 2 bit saja.

Wire pada bagian proses parsing data sudah tidak lagi berwarna pink melainkan biru karena dari proses tersebut datanya telah berubah menjadi *number*. Setelah data disendirikan bagian LSB dan juga MSB maka data harus dipersatukan kembali untuk diproses selanjutnya menggunakan komponen *join number*.

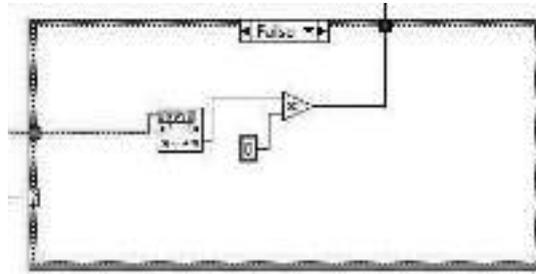
Proses selanjutnya ialah pengolahan data, data yang telah tergabung melalui komponen *join number* akan diolah untuk dapat ditampilkan pada *waveform chart* serta *Horizontal Progress Bar* dan *numeric indicator*.



Gambar 4.9 Proses pengolahan data

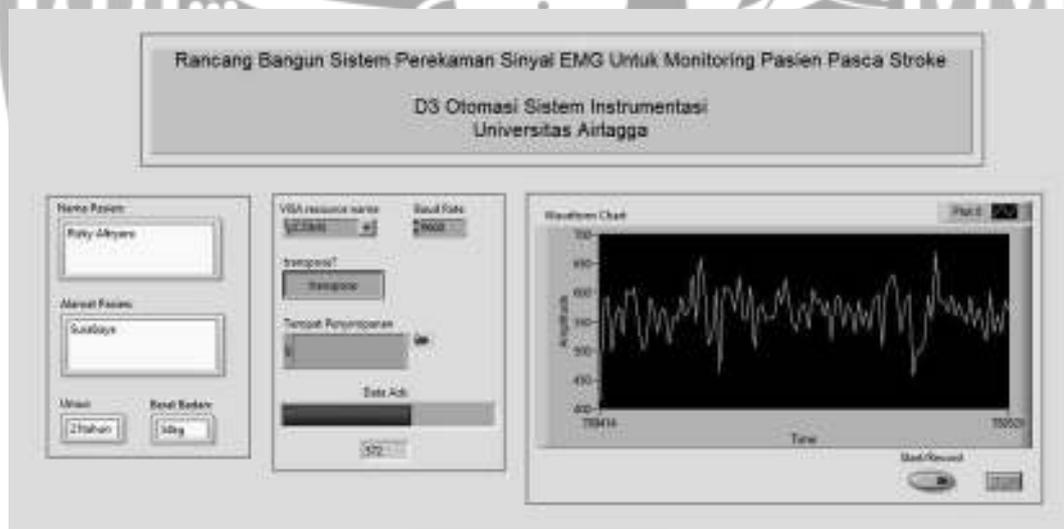
Proses tersebut hanya akan terjadi apabila pada kondisi *true*. Tetapi jika kondisi *false* maka proses pengolahan data menggunakan komponen *decimal string to number* untuk merubah *input* data *string* menjadi *number* yang kemudian diberi komponen *multiply* dengan konstanta 0 guna untuk tidak menampilkan nilai data.

Berikut adalah gambar program apabila kondisi *case structure* ialah *false*,



Gambar 4.10 program *case structure* kondisi *false*

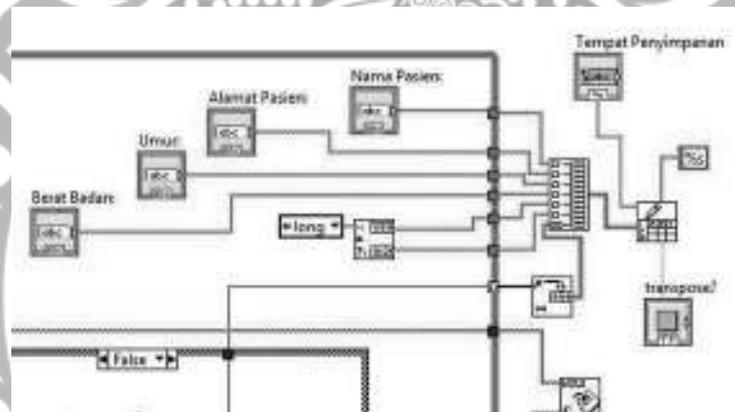
Output atau keluaran dari program ini ialah tampil dan merekamnya sinyal dari modul EMG. Data yang telah diolah akan ditampilkan pada *waveformchart* berupa grafik berjalan secara *realtime* juga *number indicator* yang akan menampilkan nilai ADC.



Gambar 4.11 Tampilan pada *user interface* LabVIEW

Data sinyal EMG akan terus tampil apabila *button stop* tidak ditekan. Apabila *button stop* = 1 (di-klik) maka akan berlanjut ke proses selanjutnya yaitu penyimpanan data ke *Ms. Excel*.

Pada proses penyimpanan data di *Excel*, data yang disimpan antara lain nama pasien, alamat rumah pasien, umur pasien, berat badan pasien, tanggal dan waktu pada saat proses perekaman serta hasil rekam sinyal otot pasien, data tersebut berupa data *string*. Data sinyal yang disimpan berupa data ADC.

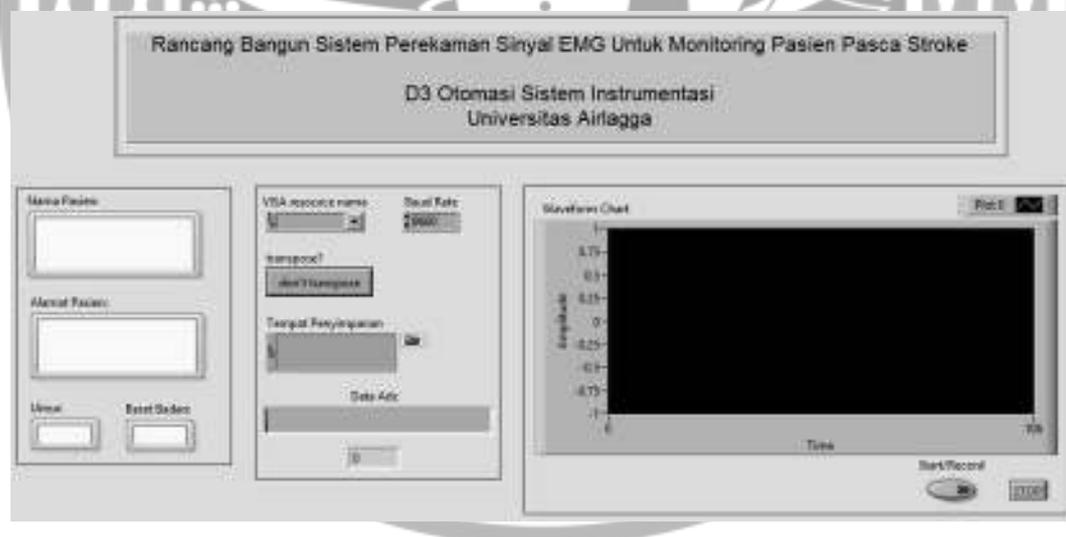


Gambar 4.12 Program penyimpanan data ke *Ms. Excel*

Komponen yang digunakan untuk menyimpan data ke *Excel* ialah *Write to Spreadsheet* yang berada diluar *while loop*. Komponen ini terdiri dari *file path* dan juga *transpose*. *File path* berguna untuk menentukan letak penyimpanan *file* sedangkan *transpose* digunakan untuk menjadikan data yang disimpan bentuknya vertikal (kebawah). Terdapat juga *%s* yang artinya tipe data yang digunakan pada saat proses penyimpanan ialah tipe data *string*.

Sebelum disimpan ke *excel* melalui komponen *write to spreadsheet*. Data yang telah tampil di *waveform chart* akan dirubah menjadi bentuk *string* kembali guna untuk bisa disimpan menggunakan komponen *number to decimal string*. Dari banyaknya data yang akan disimpan semuanya memiliki tipe data *string*. Dari semua tipe data *string* dimasukkan ke dalam komponen *build array* untuk dijadikan satu dan dikelompokkan agar tidak acak ketika disimpan pada *excel*.

Dan pada bagian *loop tunnel* data dari *waveform chat* dipilih *tunnel mode* nya ialah *indexing* supaya data yang diterima oleh *build array* adalah data keseluruhan. Sedangkan untuk nama pasien, alamat, berat badan, umur, tanggal dan juga waktu *loop tunnel* dipilih pada bagian *last value* atau data yang akan disimpan hanyalah data terakhirnya saja.



Gambar 4.13 Tamplan *User Interface* pada LabVIEW bagian *front panel*

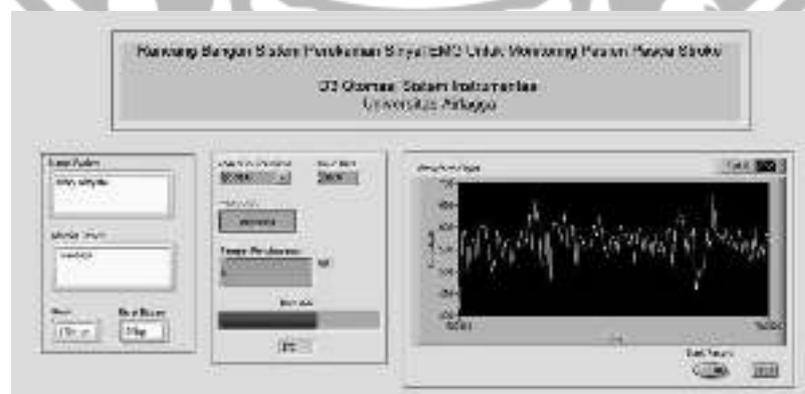
4.1.3 Pengujian *Output* Modul EMG

Pengujian ini dilakukan setelah seluruh program telah selesai dibuat dan disimulasikan bersama dengan *hardware*. Penyusun melakukan pengambilan data sinyal otot bisep beberapa orang untuk diuji output dari modul EMG.

Langkah pertama ialah menempelkan 2 elektroda pada bagian otot bisep dengan posisi vertikal (atas bawah) untuk mengetahui kondisi otot bisep dan 1 elektroda pada bagian bawah lengan dekat dengan siku, hal ini dilakukan untuk mendapatkan sinyal referensi. Setelah semua elektroda tertempel dan dipastikan sudah tertempel dengan erat maka langkah selanjutnya ialah memilih *button* “*start/pause*” yang ada pada tampilan LabVIEW.

Ketika tombol *start/pause* di-klik maka perekaman sinyal otot bisep seseorang terekam dan hasilnya tampil pada *waveform chart* yang ada di LabVIEW, tampilan grafik sinyal ini *real time* atau tampil terus sesuai dengan lamanya waktu perekaman.

Berikut adalah gambar perekaman sinyal EMG pada display LabVIEW:



Gambar 4.14 Tampilan Perekaman Sinyal EMG pada Lab View

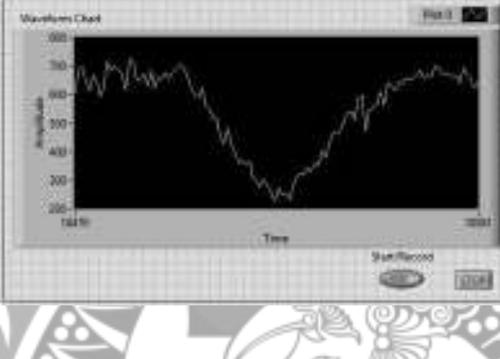
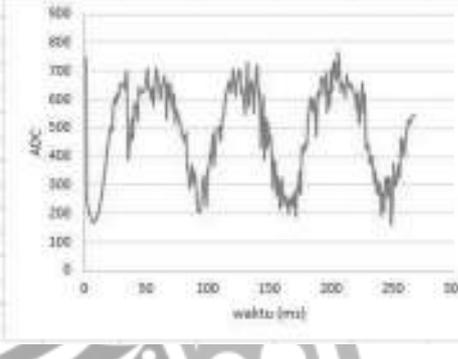
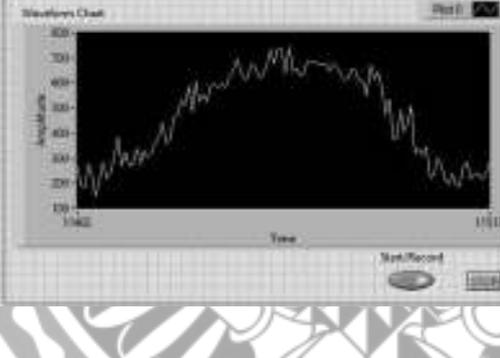
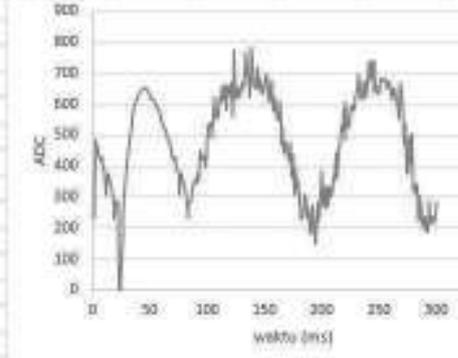
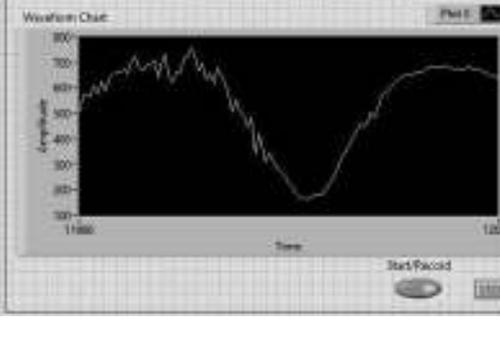
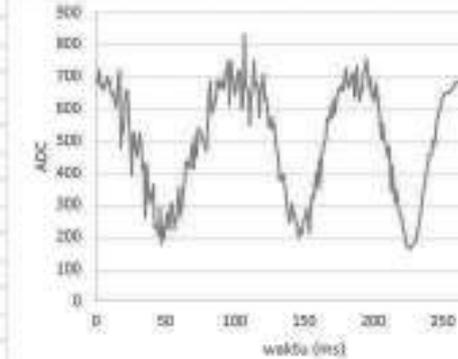
Setelah dirasa cukup dalam proses pengambilan data perekaman sinyal otot maka dengan memilih *button stop*, *interface* otomatis membuka halaman untuk tempat penyimpanan *file* perekaman sinyal otot tersebut. Isi *file* dari yang tersimpan ialah nama, alamat, berat badan, umur pasien, tanggal perekaman, waktu pelaksanaan perekaman serta hasil atau tampilan *waveform chart* tapi dalam bentuk data ADC.

	A	B	C	D	E
1	Rizky Altryara				
2	Surabaya				
3	21tahun				
4	54kg				
5	Tuesday, August 2, 2016				
6	15:20				
7	567				
8	568				
9	538				
10	559				
11	583				
12	575				
13	568				
14	566				
15	564				
16	566				
17	565				
18	565				
19	565				
20	535				
21	566				
22	564				
23	537				
24	567				
25	550				
26	556				
27	560				
28	559				

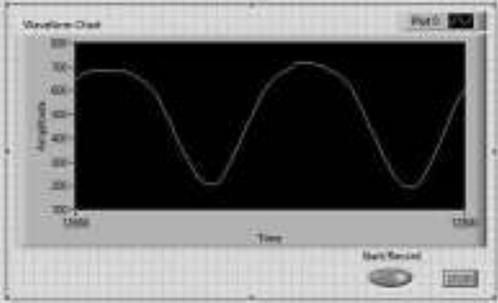
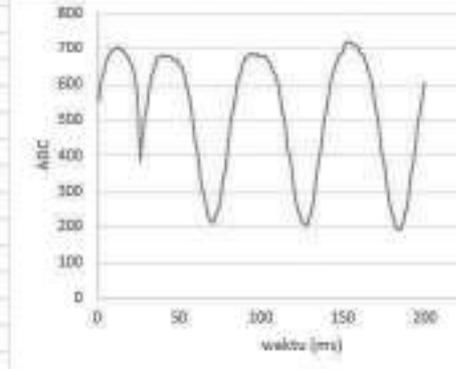
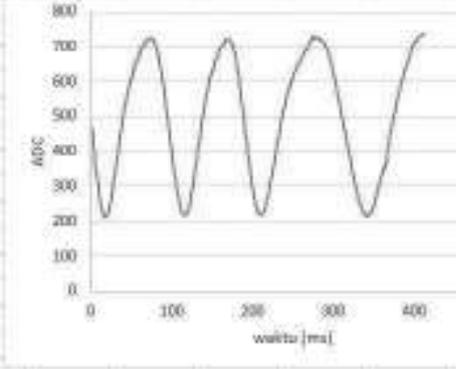
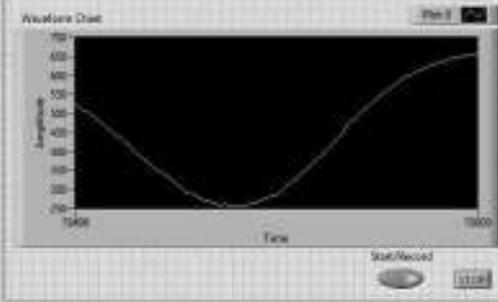
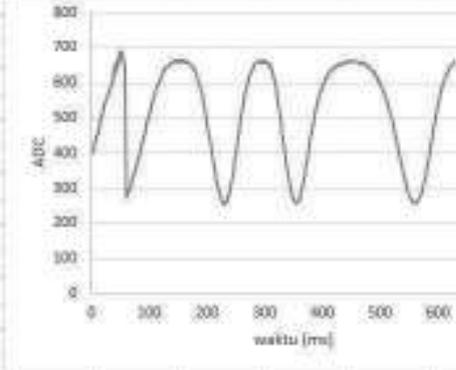
Gambar 4.15 Tampilan isi penyimpanan *file*

4.2 Bentuk sinyal EMG

Tabel 4.1 Tampilan hasil rekam sinyal EMG pada *interface* dan *Excel* pada gerakan fleksi

Percobaan ke	Tampilan LabVIEW	Tampilan Ms. Excel
1		
2		
3		

Tabel 4.2 Tampilan hasil Rekam Sinyal EMG pada *interface* dan *Excel* pada gerakan ekstensi

Percobaan Ke-	Tampilan LabVIEW	Tampilan <i>Ms.Excel</i>
1		
2		
3		

Dari Tabel 4.1 pada saat keadaan otot relaksasi, dapat diketahui bahwa data adc terbesar yang terekam adalah 473. Sedangkan data adc terbesar pada saat kontraksi yang terekam adalah 680. Dibutuhkan perhitungan untuk mengetahui tegangan yang dihasilkan oleh otot saat relaksasi dan kontraksi, berikut adalah perhitungannya:

Tabel 4.3 Perhitungan untuk mengetahui tegangan dari modul EMG yang masuk ke mikrokontroler

Relaksasi / Ekstensi	Kontraksi / Fleksi
Bit = $\frac{\text{tegangan input}}{5V} \times 1024$	Bit = $\frac{\text{tegangan input}}{5V} \times 1024$
Tegangan input = $\frac{473}{1024} \times 5V$	Tegangan input = $\frac{680}{1024} \times 5V$
Tegangan input = 2,3V	Tegangan input = 3,32V

Dari perhitungan pada Tabel 4.3, dapat diketahui bahwa tegangan *output* dari modul EMG yang dibuat masih dalam *range* sinyal yang tercantum dalam jurnal yaitu 0,4V sampai 5V.

BAB V

KESIMPULAN DAN SARAN

5.1 KESIMPULAN

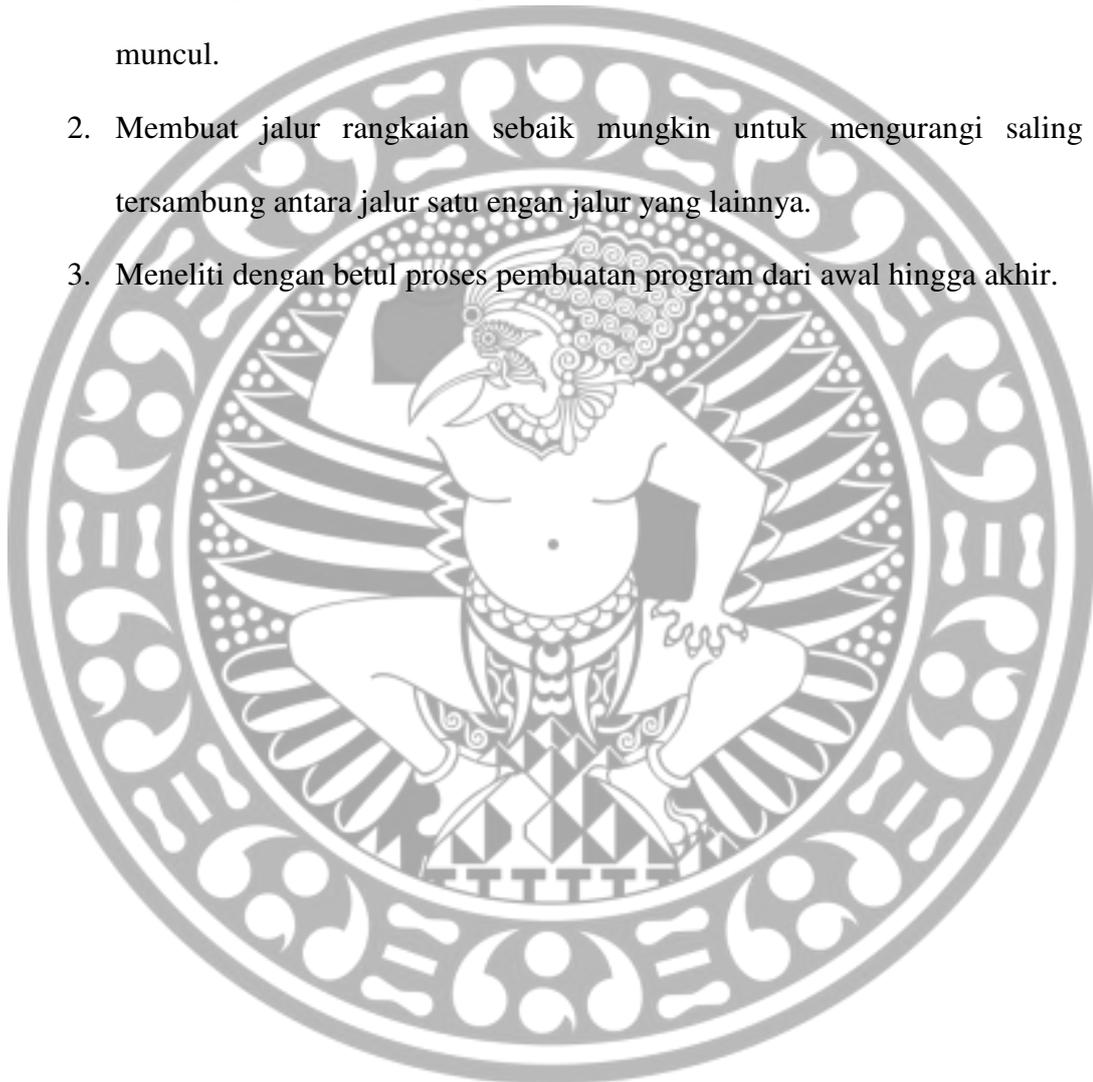
Dari hasil perancangan Sitem Perekaman Sinyal EMG maka dapat ditarik kesimpulan:

1. Perwujudan sistem perekaman sinyal EMG ini terdiri dari 2 yaitu perwujudan *hardware* dan juga perwujudan *software*. Dari perwujudan sistem tersebut akuisisi data dapat dilakukan mulai dari pembuatan *hardware* yang mencakup 5 macam rangkaian yaitu *Differensial Amplifier, Non Inverting, Low Pas Filter, Notch Filter* dan *Clamper* yang berfungsi untuk menguatkan dan *me-filter* sinyal otot yang di tangkap/diterima oleh elektroda. Kemudian dilanjut dengan pemrograman di IDE Arduino yang berguna untuk mengeksekusi, mengkonversi data analog (sinyal otot) menjadi data digital kemudian data yang telah di proses akan berlanjut ke PC dengan LabView sebagai *interface*
2. Sistem antarmuka yang digunakan ialah pemrograman dari LabView. Dengan tampilan *interface* yang mampu menyimpan secara otomatis data hasil perekaman sinyal EMG akan memudahkan untuk memantau sinyal otot pasien pasca stroke.

5.2 SARAN

Dari hasil perancangan Sitem Perekaman Sinyal EMG maka dapat ditingkatkan lagi kinerjanya untuk menghasilkan hasil yang lebih baik, antara lain:

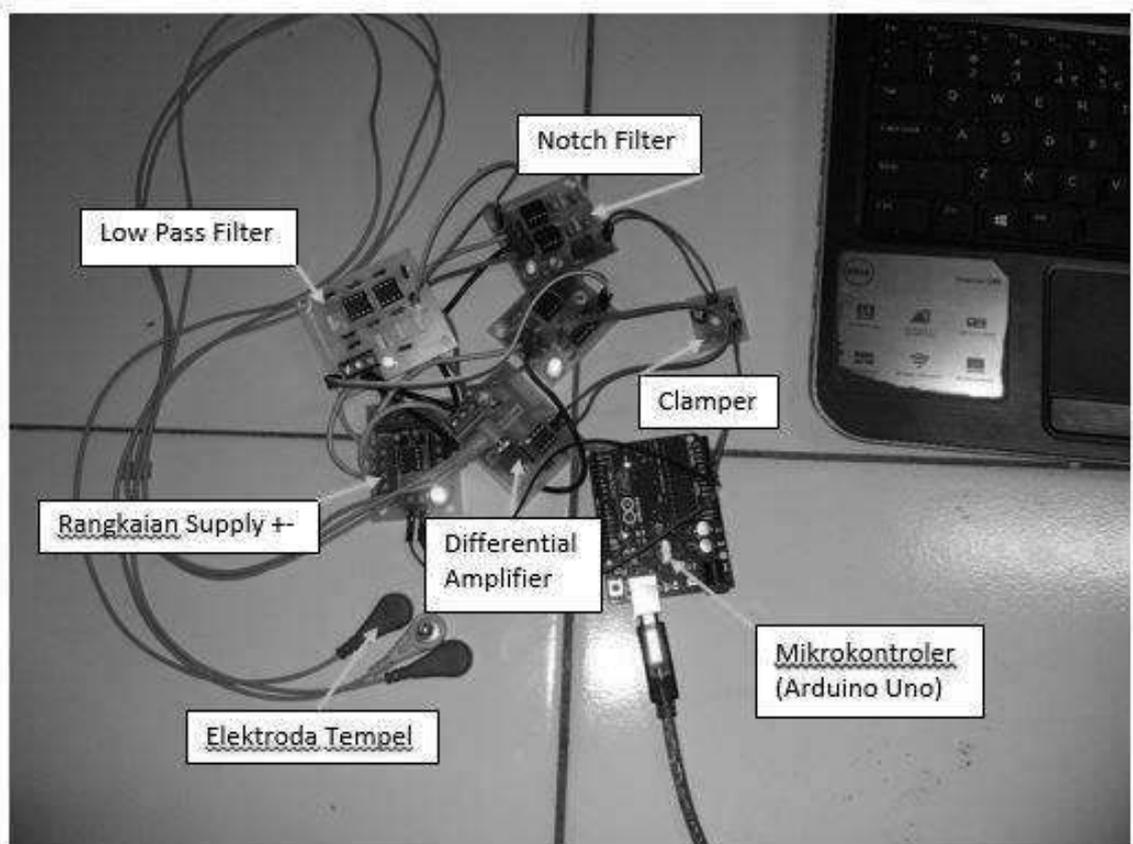
1. Menyempurnakan teknik penyolderan untuk mengurangi noise yang muncul.
2. Membuat jalur rangkaian sebaik mungkin untuk mengurangi saling tersambung antara jalur satu engan jalur yang lainnya.
3. Meneliti dengan betul proses pembuatan program dari awal hingga akhir.



DAFTAR PUSTAKA

- Artanto, Dian. 2012. **Interaksi Arduino dan LabView**. Jakarta: PT Elex Media Komputindo.
- Arisma, Prisa. 2015. Rancang Bangun Sistem Monitoring ECG Berbasis *Wireless* Bagian II. Surabaya:Universitas Airlangga.
- Baker, L.L., “Clinical uses of neuromuscular electrical stimulation,” in R. P. Nelson, and D. P. Currier, eds., *Clinical electromyography*, 2nd ed., Appleton and Lange, Norwalk, CT, 1991.
- Basmajian, J. V., ed, *Biofeedback: Practice and principles*, 3rd ed., Williams and Wilkins, Baltimore, 1989.
- Cram, J.R., and D. Engstrom “Patterns of neuromuscular activity in pain and non-pain Patients,” *Clin BiofeedbackmHealth* 9, 106-116 (1986).
- <https://www.arduino.cc/> diakses pada 17 Desember 2015
- <http://www.elektronika-dasar.web.id/> diakses pada 22 Desember 2015
- <https://www.ni.com/labview/> diakses pada 17 Desember 2015
- Muhammad Rivai. 2000. **Sistem Pemrosesan Sinyal**. Institut Sepuluh Nopember Surabaya.
- Muzakki, Ahmad.2015.Rancang Bangun Sistem Monitoring ECG Berbasis *Wireless* Bagian I.Surabaya:Universitas Airlangga.
- Rachnad Setiawan, 1999. **Sistem Pengukuran Sinyal Biomedika**. Institut Sepuluh Nopember Surabaya.
- Robert F Coughlin, Frederick F Driscoll, 1992. **Penguat Operasional Dan Rangkaian Terpadu Linear**. Edisi kedua, diterjemahkan oleh Herman Widodo Soemitro, Jakarta, Erlangga.

LAMPIRAN



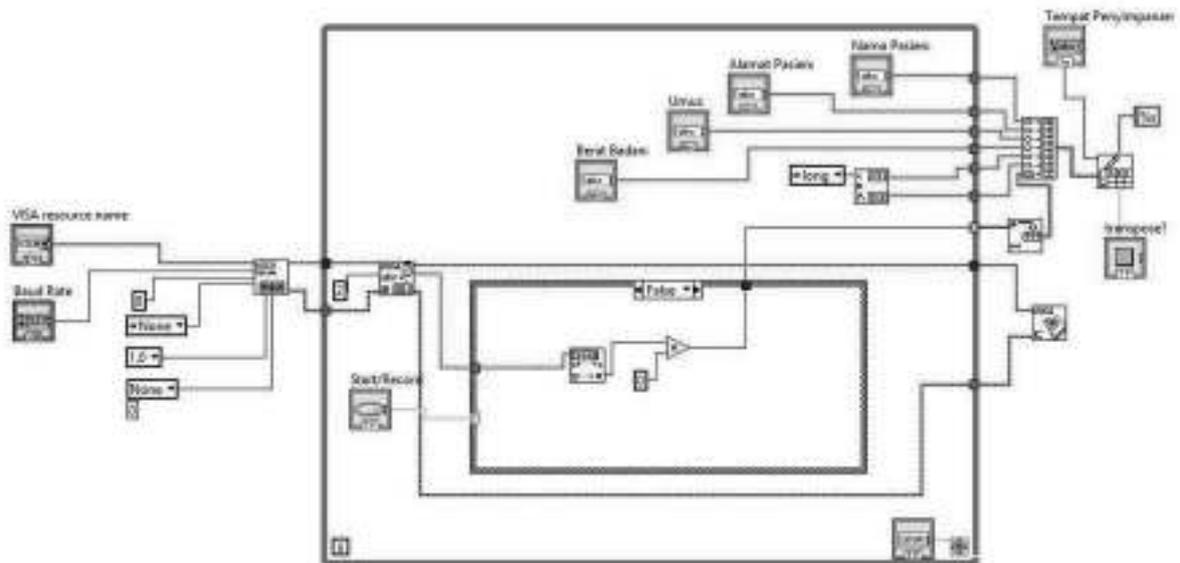
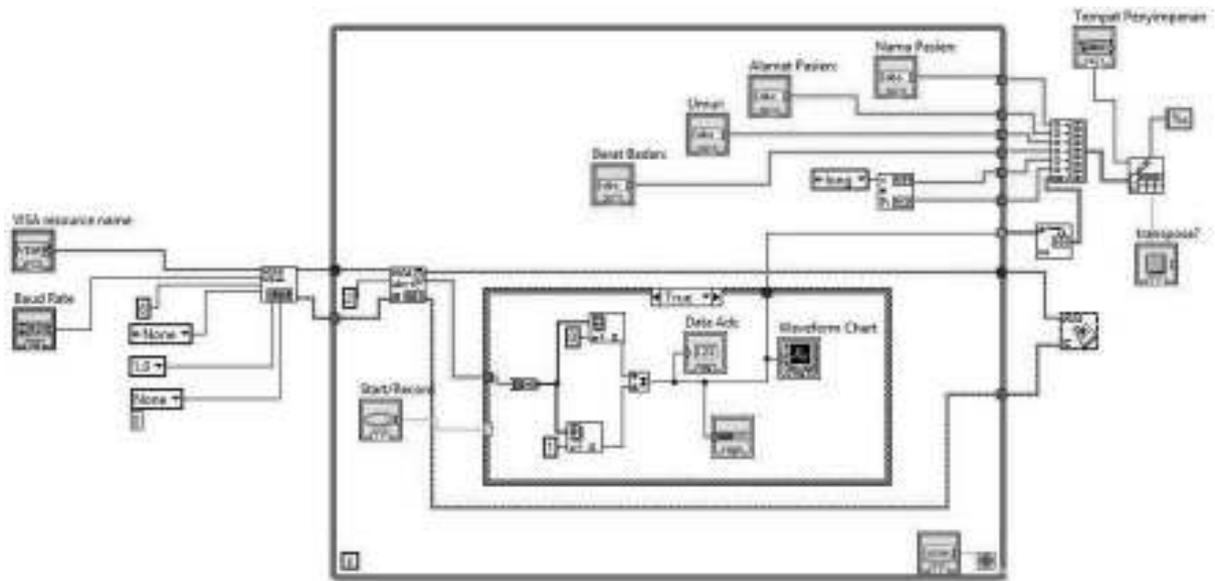
Rangkaian Sistem Perekaman sinyal EMG

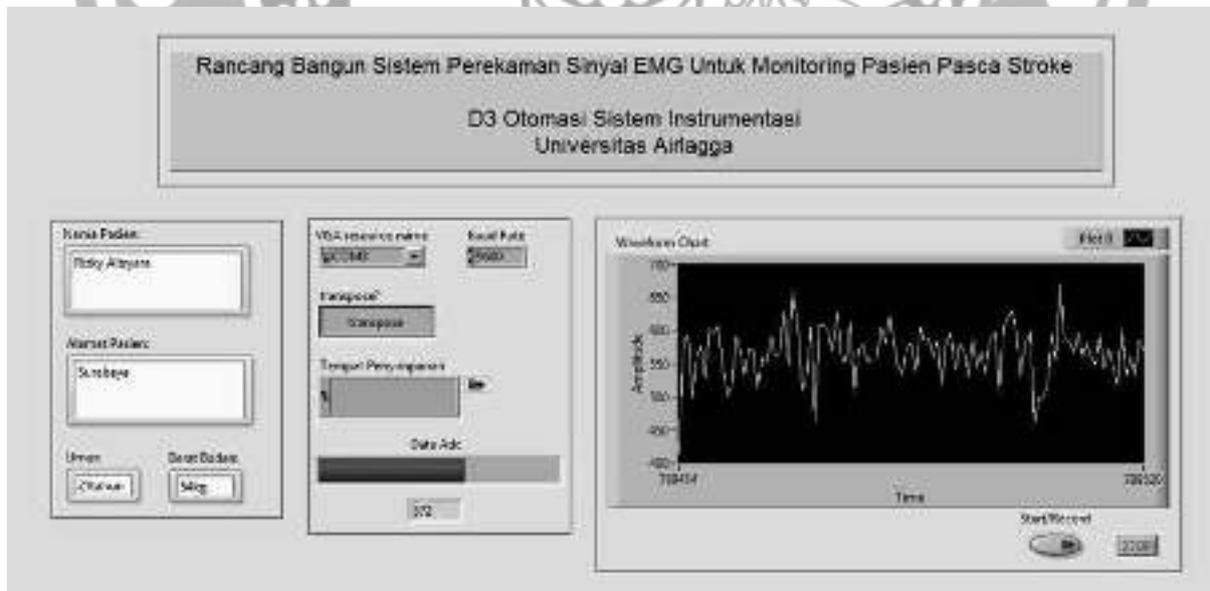
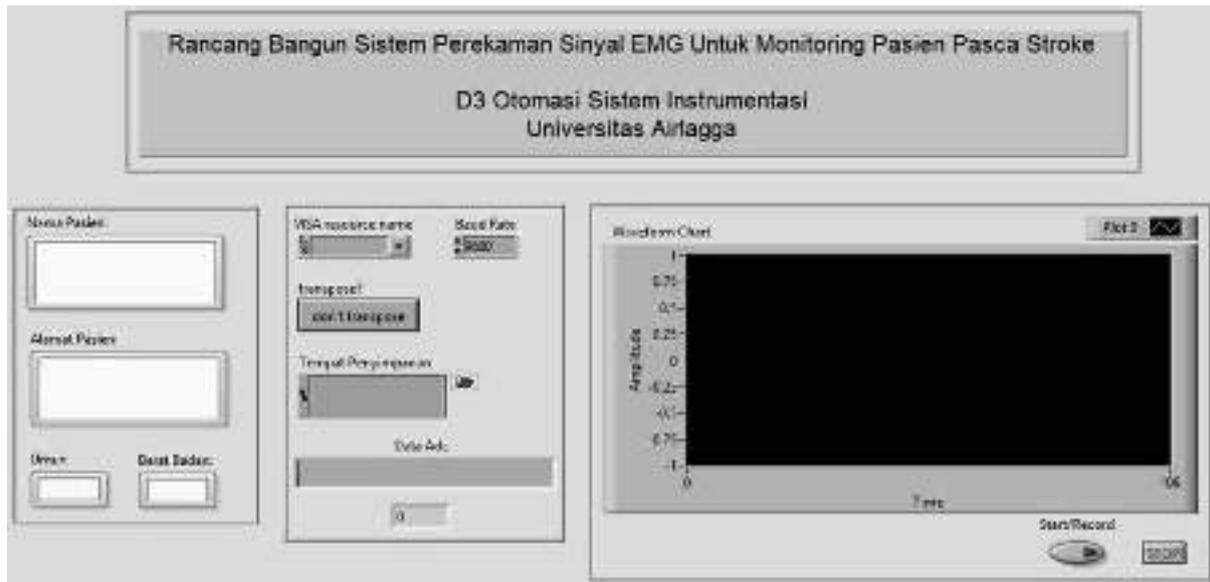


The image shows a screenshot of the Arduino IDE 1.0.5 software. The main window displays a C++ sketch for reading an analog sensor and sending data to the serial monitor. The sketch is as follows:

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  int adc = analogRead(A0);  
  Serial.write(highByte(adc));  
  Serial.write(lowByte(adc));  
  delay(10);  
}
```

Below the sketch editor, the serial monitor shows the message "Done uploading." and "Binary sketch size: 2,054 bytes (of a 32,256 byte maximum)". The status bar at the bottom indicates "10" and "Arduino Uno on COM3".





FILE HOME INSERT PAGE LAYOUT FORMULAS

A7 : X ✓ fx 567

	A	B	C	D	E
1	Rizky Altryara				
2	Surabaya				
3	21tahun				
4	54kg				
5	Tuesday, August 2, 2016				
6	15:20				
7	567				
8	568				
9	538				
10	559				
11	583				
12	575				
13	568				
14	566				
15	564				
16	566				
17	565				
18	565				
19	565				
20	535				
21	566				
22	564				
23	537				
24	567				
25	550				
26	556				
27	560				
28	559				

Data sinyal baru