



UNIVERSITAS INDONESIA

**SISTEM MONITORING DATA ELEKTROKARDIOGRAM
MENGUNAKAN TINI MIKROKONTROLLER**

SKRIPSI

Diajukan sebagai satu syarat untuk memperoleh gelar Sarjana Sains

**ABDUL RASYID ZAKARIA
0304027013**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN
ALAM
DEPARTEMEN FISIKA
DEPOK
JUNI 2009**

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Abdul Rasyid Zakaria
NPM : 0304027013
Program Studi : Fisika Reguler
Judul Skripsi : Sistem Monitoring Data Elektrokardiogram
Menggunakan TINI Mikrokontroller

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Sains pada Program Studi Fisika, Fakultas Matematika dan Ilmu pengetahuan Alam, Universitas Indonesia

DEWAN PENGUJI

Pembimbing 1 : Dr. Sastra Kusuma Wijaya (.....)
Pembimbing 2 : Surya Darma M, Si (.....)
Penguji 1 : Dr. Prawito (.....)
Penguji 2 : Prof. Dr. Djarwani S. Soejoko (.....)

Ditetapkan di : Depok
Tanggal : 19 Juni 2009

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : Abdul Rasyid Zakaria

NPM : 0304027013

Tanda Tangan :

Tanggal : 7 Mei 2009

KATA PENGANTAR/UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Sains Jurusan Fisika pada Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

- (1) Dr. Sastra Kusuma Wijaya, selaku dosen pembimbing 1 yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
- (2) Surya Darma M,Si, selaku dosen pembimbing 2 yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
- (3) Bpk. Sukatman dan Bpk. Budi, selaku laboran laboratorium yang saya gunakan yang telah banyak membantu dalam usaha menyediakan peralatan penelitian saya;
- (4) orang tua dan keluarga saya yang telah memberikan bantuan dukungan material dan moral;
- (5) Alm. Raditya Dimas dan Alm. Achmad Isradi Aditya selaku sahabat terbaik saya yang telah memberikan dukungan, menyediakan waktunya untuk mendengarkan segala keluhan saya selama penelitian. Kehadiran dan mengenal mereka merupakan berkah untuk saya;
- (6) Alm. M. Choliq, selaku sahabat yang telah memberikan waktunya untuk saling berbagi pengetahuan dan sebagian waktunya yang telah dia berikan untuk saya dalam prose penyelesaian skripsi ini;
- (7) Welly Anggoro, selaku sahabat yang telah menyediakan diri, waktu dan pikirannya untuk bertukar pikiran dalam penyusunan skripsi ini;
- (8) Budi Purnomojati, Zamroni, Sugiharto, dan semua teman mahasiswa angkatan 2004 yang telah mendukung dan memberikan bantuan secara moral sehingga penelitian ini dapat selesai;

(9) Franky Siadari telah banyak membantu saya dalam menyelesaikan skripsi ini;

(10) Helmi Indra, Idha Rachmawati, Dian Wulan, dan Akhmad Kholidin, selaku teman satu perjuangan yang telah memberikan semangat pada saya selama penelitian;

(11) Gemmy Adyeandra, Jimmy Daniel, Puspita Hudanyanti, Aditya Wira, Tajuddin Noer, Riki Fahlevi Zain, M. Nanda, M. Joel B, Yan S, dan beberapa sahabat saya dari semua angkatan di jurusan Fisika yang telah memberikan berjuta semangat, menyediakan tempat naungan selama penelitian, memperbolehkan saya untuk menggunakan laptopnya untuk tujuan penelitian serta memberikan sebagian materinya untuk proses penyusunan skripsi ini;

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 20 Juni 2009

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Abdul Rasyid Zakaria
NPM : 0304027013
Program Studi : Fisika S1 Reguler
Peminatan : Instrumentasi
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul :

Sistem Monitoring Data Elektrokardiogram Menggunakan TINI Mikrokontroller

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 9 Mei 2009

Yang menyatakan


(Abdul Rasyid Zakaria)

ABSTRAK

Nama : Abdul Rasyid Zakaria
Program studi : Fisika S1 Reguler
Judul : Sistem Informasi Monitoring Data Elektrokardiogram
Menggunakan TINI Mikrokontroller

Elektrokardiogram (EKG) merupakan alat untuk mendeteksi aktifitas listrik yang terjadi di jantung. Aktifitas listrik tersebut menimbulkan potensial aksi. TINITM Mikrokontroller, sebuah mikrokontroller buatan Dallas Semikonduktor memiliki beberapa layanan untuk I/O (*i-Wire, i-Button, Serial, Ethernet, dll*). Penelitian ini memfokuskan pada fasilitas *Ethernet* (LAN) dengan menkonfigurasi TCP/IP dan port TINI untuk memonitor aktifitas listrik tersebut. Pengolahan data dilakukan dengan Java. Untuk proses konversi data analog dari EKG digunakan sebuah Mikrokontroller AVR berbasis 10-bit. Hasil dari aktifitas listrik tersebut dikuatkan menggunakan instrumentasi *amplifier* AD620AN dan di saring oleh *notch filter* 50 Hz menggunakan OP 07 dan TL 081.

Kata kunci:

Elektrokardiogram, TINI Mikrokontroller, Java, *Ethernet*, NetBeans

ABSTRACT

Electrocardiogram is a device for detecting some electric activity which conduct in heart. There activity will producing some action potential. TINI Microcontroller, a device belong to Dallas Semiconductor have many facilities or I/O activity (*1-Wire, i-Buttonm Serial, Ethernet, etc*). This research will focusing to *Ethernet* device with configure TCP/IP and port in TINI for electric aktivty monitoring system. Data proessing using AVR Microcontroller 10-bit. Product from electric activity will gained using AD620AN instrumentation amplifier and will filtered with 50 Hz notch filter using OP 07 and TL081.

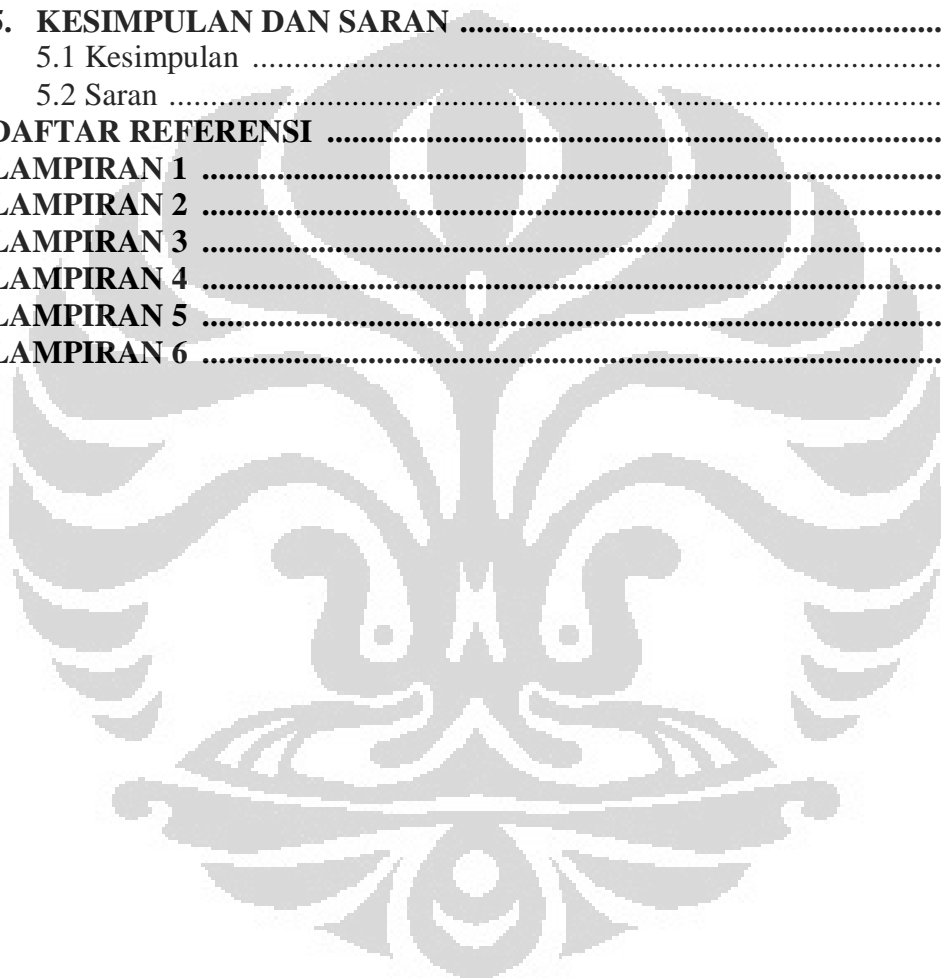
Key words :

Electrocardiogram, TINI Microcontroller, Java, *Ethernet*, NetBeans

DAFTAR ISI

| | |
|---|-------------|
| HALAMAN JUDUL | i |
| LEMBAR PENGESAHAN | ii |
| LEMBAR PERNYATAAN ORISINALITAS | iii |
| KATA PENGANTAR..... | iv |
| LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH | vi |
| ABSTRAK | vii |
| DAFTAR ISI | viii |
| DAFTAR GAMBAR..... | x |
| 1. PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Tujuan Penelitian | 2 |
| 1.3 Pembatasan Masalah | 2 |
| 1.4 Metodologi Penelitian | 3 |
| 1.5 Sistematika Penulisan | 4 |
| 2. TEORI DASAR | 6 |
| 2.1 Elektrokardiogram | 6 |
| 2.1.1 Pengertian Elektrokardiogram..... | 6 |
| 2.1.2 Bioelektrik | 6 |
| 2.1.3 Sistim Konduktivitas Jantung | 8 |
| 2.1.4 Force Vektor dan Depolarisasi | 9 |
| 2.1.5 Lead | 11 |
| 2.1.6 EKG Waveform | 13 |
| 2.2 Sistim Elektronika | 15 |
| 2.2.1 Bioelektroda | 15 |
| 2.2.2 Sistim Penggerak Kaki Kanan | 17 |
| 2.2.3 Instrumentasi Amplifier | 18 |
| 2.2.4 Filter Aktif | 20 |
| 2.2.5 ATmega 324 p..... | 21 |
| 2.3 TINI Mikrokontroller | 22 |
| 2.3.1 Hardware | 23 |
| 2.3.2.Ethernet | 25 |
| 2.3.3 Komunikasi Serial | 27 |
| 2.3.4 TINI Kernel | 30 |
| 2.3.5 Inisialisasi Serial..... | 34 |
| 2.4.6 Inisialisasi TCP/IP | 36 |
| 2.4.7 Netbeans 6.5 | 36 |
| 3. PERANCANGAN SISTEM..... | 38 |
| 3.1 Alur Penelitian | 38 |
| 3.2 Peralatan | 39 |
| 3.3 Sistim Hardware | 40 |
| 3.3.1 Instrumentasi Amplifier | 40 |
| 3.3.2 Perancangan Filter Aktif | 41 |
| 3.3.3 Perancangan Direct Adder dan Penguatan 100 kali | 42 |
| 3.4 Perancangan ADC | 45 |
| 3.3 Alur Software | 40 |

| | |
|--|-----------|
| 4. HASIL DAN PEMBAHASAN | 47 |
| 4.1 Hasil Penelitian | 47 |
| 4.1.1 Uji Penguatan | 47 |
| 4.1.2 Data Low-Pass Filter 10 Hz | 49 |
| 4.1.3 Data Elektrokardiogram | 50 |
| 4.1.4 Data ADC | 52 |
| 4.1.5 Layout PC | 54 |
| 4.2 Analisa | 55 |
| 4.2.1 Uji Linieritas | 55 |
| 4.2.2 Analisa Data | 56 |
| 4.2.3 Pembahasan Software | 57 |
| 5. KESIMPULAN DAN SARAN | 58 |
| 5.1 Kesimpulan | 58 |
| 5.2 Saran | 58 |
| DAFTAR REFERENSI | 60 |
| LAMPIRAN 1 | 61 |
| LAMPIRAN 2 | 64 |
| LAMPIRAN 3 | 66 |
| LAMPIRAN 4 | 68 |
| LAMPIRAN 5 | 70 |
| LAMPIRAN 6 | 71 |



DAFTAR GAMBAR

| | | |
|-------------|---|----|
| Gambar 1.1 | Diagram langkah-langkah penelitian | 3 |
| Gambar 2.1 | Pergerakan ion Na dan K dalam sebuah sel | 7 |
| Gambar 2.2 | Resting Potential yang dihasilkan oleh sel..... | 8 |
| Gambar 2.3 | Force Vector | 10 |
| Gambar 2.4 | Penjumlahan vector dalam depolarisasi jantung | 10 |
| Gambar 2.5 | Cardiac Axis dalam beberapa vector | 11 |
| Gambar 2.6 | Lead yang berada di dada | 13 |
| Gambar 2.7 | Waveform EKG | 13 |
| Gambar 2.8 | Beberapa bentuk gelombang kompleks QRS | 14 |
| Gambar 2.9 | Antarmuka Elektroda-elektrolit | 15 |
| Gambar 2.10 | Model rangkaian untuk elektroda medis | 17 |
| Gambar 2.11 | Skematik penggerak kaki kanan | 18 |
| Gambar 2.12 | Skematik AD620AN pada penguatan 100 | 19 |
| Gambar 2.13 | Instrumentasi Amplifier | 19 |
| Gambar 2.14 | Skematik filter aktif beserta direct adder yang digunakan | 20 |
| Gambar 2.15 | TINI Board | 22 |
| Gambar 2.16 | 72-pin SIMM Konektor | 24 |
| Gambar 2.17 | TINI runtime environment | 26 |
| Gambar 2.18 | Konfigurasi serial null-modem | 28 |
| Gambar 2.19 | Hubungan server (TINI) dengan host | 28 |
| Gambar 2.20 | Aplikasi Javakit | 31 |
| Gambar 3.1 | Diagram alur sistim penelitia | 38 |
| Gambar 3.2 | Skematik perancangan instrumentasi amplifier..... | 41 |
| Gambar 3.3 | Skematik perancangan low-pass filter 10 Hz..... | 42 |
| Gambar 3.4 | Skematik perancangan direct adder | 43 |
| Gambar 3.5 | Skematik perancangan pembagi tegangan | 43 |
| Gambar 3.6 | Skematik perancangan non-inverting amplifier | 44 |
| Gambar 3.7 | Alur software | 46 |
| Gambar 4.1 | Rangkaian pengujian penguatan | 47 |
| Gambar 4.2 | Grafik hasil uji penguatan 1 kali | 47 |
| Gambar 4.3 | Grafik hasil uji penguatan 25 kali | 48 |
| Gambar 4.4 | Grafik hasil uji penguatan 10 kali | 48 |
| Gambar 4.5 | Grafik hasil uji penguatan 50 kali | 49 |
| Gambar 4.6 | Grafik hasil uji low-pass filter 10 hz | 50 |
| Gambar 4.7 | Data analog tanpa filter | 51 |
| Gambar 4.8 | Data analog dengan low-pass filter | 51 |
| Gambar 4.9 | Data analog dengan penguatan 100 kali | 52 |
| Gambar 4.10 | Data hasil komunikasi ATMega342p dengan RS232 | 53 |
| Gambar 4.11 | Grafik uji linieritas ADC | 53 |
| Gambar 4.12 | Layout hasil | 54 |

BAB 1

PENDAHULUAN

1.1 LATAR BELAKANG

Elektrokardiogram merupakan sistem yang dapat mendeteksi sinyal listrik yang dikarenakan adanya aktifitas otot-otot jantung. Aktifitas tersebut secara normal akan menghasilkan ritme yang berulang secara kontinyu. Dalam bidang kesehatan, selain untuk memonitor aktifitas yang terjadi di jantung secara intensif, elektrokardiogram sering digunakan untuk mendiagnosa beberapa ketidaknormalan yang terjadi di jantung¹.

Elektrokardiogram menghasilkan sebuah gelombang analog yang kontinyu. Gelombang keluaran elektrokardiogram dapat dihubungkan dengan beberapa devais yang dapat menampilkan sebuah gelombang analog, umumnya digunakan osiloskop.

Penelitian difokuskan untuk membuat sebuah sistem yang dapat menampilkan data EKG melalui fasilitas LAN (Local Area Network). Dallas Semiconductor membuat mikrokontroller bernama TINI (*Tiny InterNet Interface*) yang memiliki beberapa pilihan untuk input dan output yang dapat digunakan seperti 1-Wire, i-Button ataupun *Ethernet*². Sistem yang diberikan oleh TINI merupakan sistem yang berbasis Networking³. Hal itu ditunjukkan dengan adanya ethernet yang bisa dijadikan input ataupun output.

Umumnya, TINI mikrokontroller digunakan untuk mempermudah sistem dalam otomotif, industri, robotik dan otomatisasi industri². Kemapanan sistem input dan output yang dimiliki oleh TINI membuat mikrokontroller ini digunakan untuk beberapa aplikasi industri.

Ethernet merupakan sebuah sarana penyampaian data menggunakan media network. Hal itulah yang melatarbelakangi Penulis untuk mengembangkan sarana ethernet pada TINI untuk menyampaikan data tampilan EKG. Data EKG yang didapatkan diharapkan dapat dimonitor oleh pengguna dari jarak jauh.

1.2 TUJUAN PENELITIAN

Tujuan penelitian ini adalah :

1. Mempelajari, memahami dan mengetahui cara kerja elektrokardiogram (EKG), TINI Mikrokontroller serta komunikasi yang terjadi antara mikrokontroller dengan komputer desktop
2. Mempelajari sistem pengukuran data detak jantung dengan EKG.
3. Membuat sebuah sistem penyampaian data EKG dengan aplikasi Ethernet TINI mikrokontroller ke komputer desktop.

1.3 PEMBATAAN MASALAH

Masalah yang dibahas mulai dari pendeteksian aktifitas jantung menggunakan elektrokardiogram (EKG) hingga proses pengolahan data. EKG menghasilkan output analog⁴. Sinyal yang diberikan tubuh sangat kecil, berkisar 50 milivolt¹, sehingga perlu menggunakan instrumentasi amplifier. Sinyal keluaran analog dikonversikan menjadi sebuah keluaran digital oleh ATmega 342p. Untuk mengurangi munculnya sinyal noise serta menghilangkan frekuensi 50 Hz,, Penulis menggunakan low-pass filter 10 Hz.

Keluaran digital pada konverter dihubungkan secara serial yang dimiliki TINI mikrokontroller kemudian disimpan di memori yang dimilikinya kemudian dikirim ke komputer desktop melalui fasilitas ethernet. Mikrokontroller bertindak sebagai perantara. Proses pengolahan digunakan Java. Pada penelitian ini digunakan Java J2SE dengan *editor script* Netbeans 6.5. Selanjutnya pembacaan hasil pengukuran ditampilkan pada komputer desktop.

Untuk menampilkan hasil yang diperoleh digunakan class form pada Netbeans. Program tersebut dirancang hanya untuk tujuan visualisasi dan menerima data melalui TCP/IP dari mikrokontroller.

Penulis hanya membahas bagaimana menampilkan gelombang hasil aktifitas listrik yang terjadi di jantung. Penulis tidak membahas bagaimana pengaruh umur pasien maupun beberapa keganjilan yang terdapat dalam gelombang yang dihasilkan.

1.4 METODOLOGI PENELITIAN

Metode penelitian yang dilakukan terdiri dari beberapa tahap antara lain :

a) Studi Peralatan

Studi Peralatan digunakan penulis untuk mempelajari karakteristik dan spesifikasi alat yang digunakan dalam pembuatan EKG.

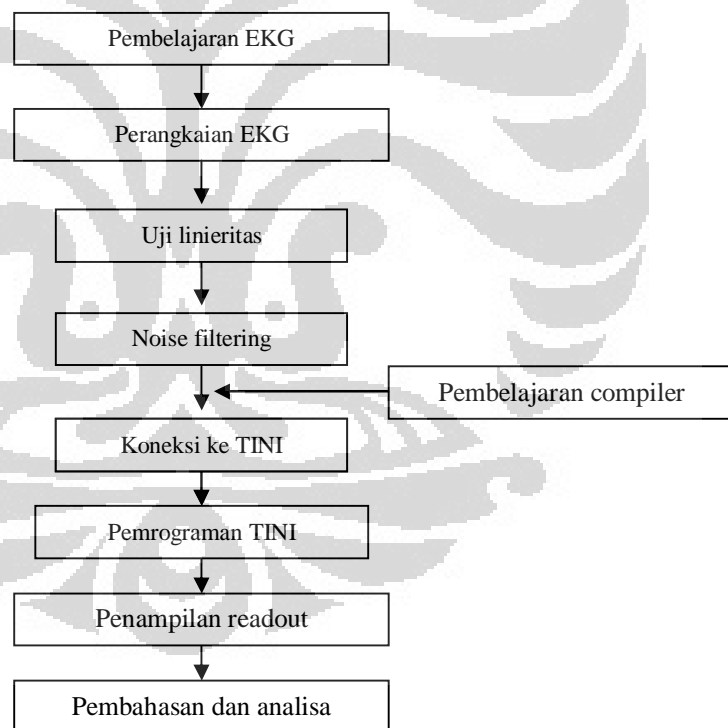
b) Studi kepustakaan

Studi kepustakaan dilakukan untuk memperoleh informasi dan pustaka yang berkaitan dengan masalah. Studi kepustakaan berupa literatur, penjelasan yang terdapat dalam internet dan buku-buku yang berhubungan.

c) Penelitian Laboratorium

Penelitian laboratorium dilakukan untuk merakit, membuat alat dan meneliti kerja alat yang telah dibuat.

Berikut ini adalah diagram langkah-langkah yang dilakukan dalam penelitian ini :



Gambar 1.1. Diagram langkah-langkah penelitian

Penjelasan diagram alur penelitian :

- 1) Pembelajaran EKG
Penulis berusaha mempelajari apa itu EKG, TINI, aplikasinya, beberapa perbedaan yang muncul dalam pembacaan serta beberapa rangkaian dan *device* yang digunakan dari beberapa literatur.
- 2) Perangkaian EKG
Perangkaian EKG dilakukan dengan merakit beberapa rangkaian. Penulis menggunakan beberapa software pendukung seperti EWB dan Diptrace.
- 3) Uji linieritas, pendeteksian keluaran dan *noise filtering*
Proses pengujian penguatan dan linieritas menggunakan skematik yang diberikan pada datasheet komponen yang dipakai. Pembacaan keluaran dilakukan dengan menghubungkan keluaran EKG yang sudah dilengkapi dengan *noise filtering* ke sebuah osiloskop dengan memperhitungkan beberapa pengkondisian. Data yang didapatkan akan dicatat, diamati serta dianalisis.
- 4) Pembelajaran Compiler dan pemrograman TINI.
Mempelajari dan menggunakan Java Development Environment, Java Communication API dan TINI Toll Kit untuk dapat mengkoneksikan dan mengkompile hasil yang didapatkan, baik dari sensor ke mikrokontroler ataupun dari mikrokontroler ke komputer desktop².
- 5) Penampilan readout
Readout ditampilkan pada komputer desktop. Sarana yang dipakai adalah Ethernet TINI Microcontroller. Layout diolah dengan tampilan Java.
- 6) Pembahasan dan analisa
Hasil dari penelitian yang telah dilakukan dalam pembuatan elektrokardiogram dengan menggunakan *Ethernet* TINI mikrokontroler.

1.5. SISTEMATIKA PENULISAN

Isi tulisan ini terdiri atas 5 bab. Diawali oleh Bab 1 yang berjudul Pendahuluan, yang berisi latar belakang, tujuan, metode penelitian, dan pembatasan masalah dalam penelitian yang dilakukan. Dilanjutkan dengan Bab 2 dengan judul Teori dasar. Bab ini menerangkan teori dasar dari EKG, hardware

yang digunakan serta beberapa teori tentang TINI Mikrokontroller. Selanjutnya Bab 3 membahas Perancangan Sistem. Bab ini menjelaskan alur sistem yang digunakan serta beberapa peralatan dan karakteristik dari peralatan yang digunakan. Pada Bab 4 yang berjudul Hasil dan Pembahasan, berisi data yang didapatkan, analisis alat yang telah dibuat, menganalisis list program yang telah dibuat, serta bagaimana proses pengolahan data. Di akhiri oleh bab terakhir, yaitu Bab 5, berisi kesimpulan dan saran. Pada skripsi ini akan disertakan beberapa lampiran tentang alur-alur yang dipakai, listing program yang dipakai, beberapa data yang didapatkan.



BAB 2 TEORI DASAR

2.1 ELEKTROKARDIOGRAM

2.1.1 PENGERTIAN ELEKTROKARDIOGRAM

Elektrokardiogram (EKG) merupakan perekaman secara elektrik kondisi jantung yang digunakan untuk menginvestigasi penyakit jantung. Pengukuran nilai potensial listrik akibat adanya aktivitas listrik yang terjadi di sekat jantung karena proses pembentukan beberapa ion. Sinyal yang dihasilkan ion-ion tersebut mengikuti pergerakan kontraksi otot jantung¹.

Secara harfiah elektrokardigram terdiri dari tiga unsur kata, yaitu *electro*, *cardio* dan *gram*. *Electro*, berhubungan dengan beberapa prinsip elektronika. Sinyal hasil aktifitas jantung akan terdeteksi oleh beberapa elektroda. Ditambah dengan penggunaan beberapa komponen elektronika, seperti instrumentasi ampliflier. Kedua, *cardio*, berasal dari bahasa Yunani yang berarti jantung. Alat ini bertujuan untuk mendeteksi ritme pompaan otot jantung ketika terjadi kontraksi dan relaksasi otot jantung dengan mengamati hasil keluaran yang diberikan oleh laju pergerakan ion dalam darah. Ketiga, *gram*, yang berarti pengukuran, yang dapat diartikan mendeteksi dan mengukur.

Gelombang listrik dihasilkan ketika otot jantung memompa darah. Gelombang tersebut dideteksi oleh beberapa elektroda yang dipasang pada kulit dalam beberapa titik. Beberapa elektroda pada titik berbeda akan mendeteksi perbedaan aktivitas dari otot jantung. Hasil ECG menunjukkan perbedaan tegangan yang terdeteksi dari beberapa elektroda yang berbeda dan pergerakan otot jantung yang terukur dalam beberapa arah.

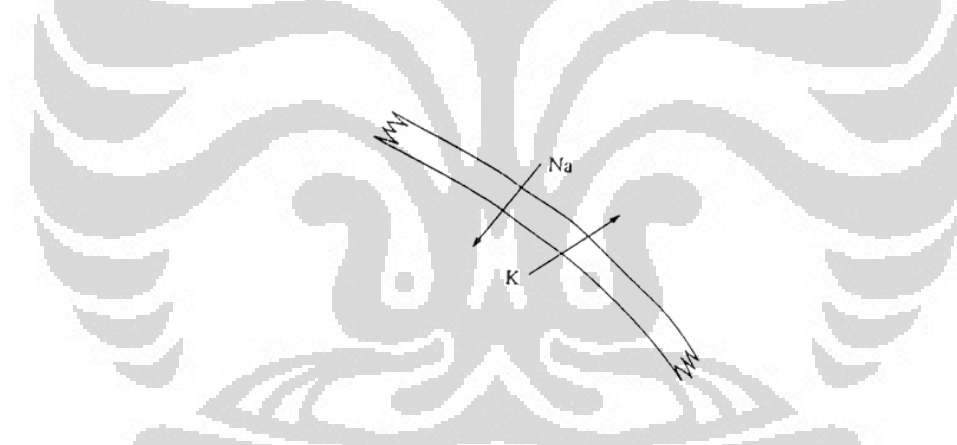
2.1.2. BIOELEKTRIK¹

Jantung merupakan sekumpulan otot-otot yang difungsikan untuk memompa dan mengalirkan darah ke seluruh tubuh. Otot-otot jantung akan berkontraksi dan memompa darah dikarenakan adanya rangsangan listrik pada sistem elektrokondutivitas dalam jantung¹. Kontraksi otot-otot jantung ditambah

dengan adanya potensial ionik menghasilkan beda potensial yang disebut sebagai “*action potential*”¹. Potensial ini menghasilkan arus listrik yang mengalir dari jantung dan menyebar ke seluruh tubuh.

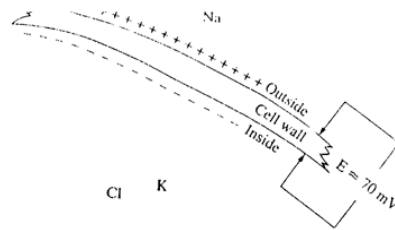
Potensial ionik dibentuk oleh sel tubuh manusia dikarenakan adanya perbedaan konsentrasi dari ion-ion kimia, ion tersebut adalah natrium (Na^+), klorida (Cl^-) dan kalium (K^+). Dinding sel menjadi media difusi beberapa ion tersebut. Permeabilitas merupakan tingkat kemampuan sebuah lapisan untuk melewatkan ion. Dinding sel merupakan lapisan semipermeable, berarti akan terjadi proses penyeleksian ion-ion dengan membiarkan atau melarangnya untuk lewat.

Konsentrasi natrium yang memiliki nilai positif yang berada di dalam sel akan lebih rendah dibandingkan konsentrasi kalium yang berada di luar sel. Peristiwa ini menjaga ion natrium lebih banyak berada di luar dan kalium berada di dalam.



Gambar 2.1. Pergerakan ion Na dan K dalam sebuah sel¹

Kalium kemudian dipompa ke dalam sel ketika natrium dipompa keluar dari sel dengan tingkat pompaan dari natrium lebih besar 2 sampai 5 kali pompaan kalium. Hal itu terjadi dikarenakan adanya perbedaan dari konsentrasi ion, potensial listrik dan tingkat polarisasi dari sel tersebut.



Gambar 2.2. Resting potential yang dihasilkan oleh sel¹

Beberapa penulisan tentang besar nilai *resting potential* sangat bervariasi, namun nilai resting potential berada antara 70 sampai 90 mV. Crouch dan McClintic menggunakan *resting potential* sebesar 85 mV. Untuk yang terbesar digunakan *resting potential* sebesar 90 mV. Pada gambar di bawah ini nilai *resting potential* dituliskan sebesar 70 mV.

Nilai potensial yang sebenarnya dapat diturunkan dari persamaan Nerst, diberikan dalam bentuk yang sederhana seperti di bawah ini :

$$E_{(mV)} = \pm 60 \ln \frac{C_o}{C_i} \quad (2.1)$$

$E_{(mV)}$ merupakan energi potensial yang dimiliki membran sel (*resting potential*), variabel C merupakan besarnya konsentrasi yang dimiliki oleh ion (C_o untuk konsentrasi ion intraselular dan C_i untuk konsentrasi ion ekstraselular).

Ketika sel dirangsang, lapisan sel berubah secara kasar dan menjadi lapisan permeable untuk ion natrium. Ion natrium didesak masuk ke sel dan kalium didesak keluar. Pergerakan ion ini menghasilkan potensial yang disebut dengan *action potential*. Nilai potensial aksi berkisar antara 20 - 40 mV lebih positif daripada daerah luar.

2.1.3 SISTIM KONDUKTIFITAS JANTUNG¹

Sinyal listrik dihasilkan jantung bermula dari sebuah titik yang disebut Sinoatrial Node (SA) yang terletak pada atrium kanan. Titik SA membuka jalan

di jantung dan menghasilkan sinyal perintis. Titik ini menghasilkan impuls listrik yang disebabkan adanya mekanisme biolistrik.

Ketika titik SA tidak menghasilkan pulsa, arus listrik yang dihasilkan menyebar melewati *atria* karena jantung berkontraksi. Kecepatan rambat untuk potensial aksi di SA sekitar 30 cm/s. Hal tersebut menyebabkan munculnya jarak antara titik SA dan AV (*Atrioventricular*), serta membuat kecepatan rambat potensial aksi mencapai 45 cm/s. Pada kecepatan rambat mencapai 45 cm/s, potensial aksi mencapai titik AV dalam 30 sampai 50 milisekon (ms) setelah titik SA dikosongkan dari muatan. Setelah itu, untuk 110 ms selanjutnya dilewatkan setelah titik AV mentransmisikan pulsa.

Pada titik AV, adanya operasi *delay line* melambatkan perambatan potensial aksi sepanjang sistem elektrokonduktivitas ke arah *ventricles*. Sel otot dari *ventricle* sebenarnya dirangsang oleh *Purkinje Fibers*. Potensial aksi merambat sepanjang serat tersebut lebih cepat, biasanya dalam orde 2 sampai 4 m/s (200-400 cm/s). Perkembangan konduktivitas dalam *purkinje fibers* tergolong cepat. Potensial aksi melewati daerah antara SA dan AV dengan waktu sekitar 40 ms dan ditunda dalam titik AV sekitar 110 ms.

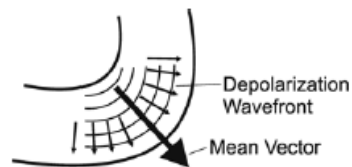
Potensial aksi yang dihasilkan oleh titik SA merangsang otot-otot serat untuk mengaktifkan *myocardium*. Ketika otot berkontraksi, volume di ruang *ventricular* akan berkurang, sehingga darah akan ditekan keluar. Kontraksi beberapa sel otot pada satu waktu memunculkan sinyal listrik yang mampu dideteksi oleh elektroda yang ditempatkan pada kulit pasien. Sinyal listrik yang dideteksi akan memunculkan sebuah bentuk gelombang.

2.1.4 FORCE VECTOR DAN DEPOLARISASI⁵

Munculnya potensial aksi berasal dari proses polarisasi dan depolarisasi yang dilakukan oleh sebuah elektron yang bergerak melewati sel. Gelombang depolarisasi yang muncul tidak bergerak lurus, tapi menyebar ke seluruh sel seperti layaknya sebuah riak dalam danau ketika kita melempar batu⁵.

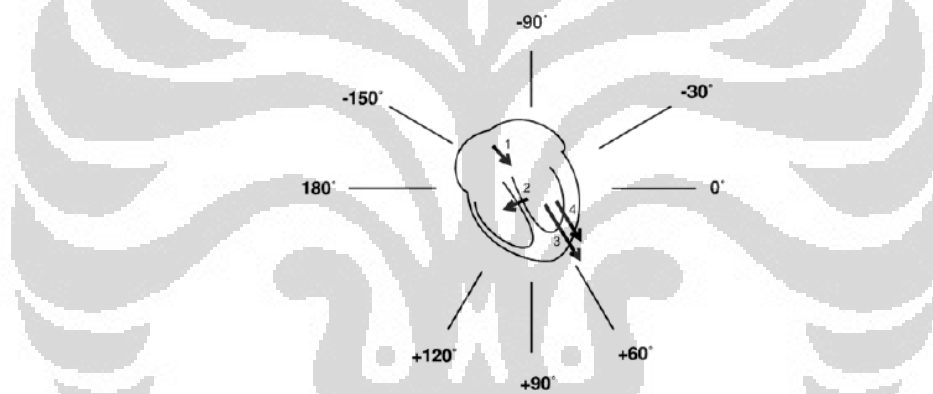
Gambar di bawah ini merupakan sebuah hasil depolarisasi yang menyebar dari *endocardium* ke *epicardium* seperti sebuah kurva. Tanda panah pendek menggambarkan arah dari pergerakan gelombang terhadap gaya (*force*), dalam hal

ini tegangan. Panah ini yang dikenal sebagai *force vectors*. Tanda panah besar merupakan *summing vector*, yang merupakan jumlah dari seluruh *force vector* ketika mengeliminasi kurva⁵.



Gambar 2.3. Force vector⁵

Gambar di bawah ini mengilustrasikan rangkaian utama dari depolarisasi jantung dan tegangan yang dihasilkan. Gambar tersebut dapat menjelaskan arah beberapa vektor yang mengalir dengan lingkaran penuh 360 derajat.



Gambar 2.4. Penjumlahan vektor dalam depolarisasi jantung⁵

Titik 0° dapat diletakkan pada daerah sebelah kanan sumbu *horizontal* dan berjalan searah jarum jam. Hal ini adalah ketentuan yang telah ditetapkan dalam pengukuran elektrokardiogram.

Vektor 1 merupakan depolarisasi di atria dan akan menyebar ke bawah atria melewati bagian kiri pasien lalu membesar sekitar 40 sampai 50° . Vektor ini akan merepresentasikan gelombang P dalam gelombang umum elektrokardiogram.

Vektor 2 merepresentasikan depolarisasi di sekat *ventricular*, bagian pertama dari *vertices* yang diaktifkan dan menentukan bagaimana munculnya gelombang kompleks QRS. Depolarisasi di sekat bergerak dari kiri ke kanan. Hal

tu dikarenakan secara normal impuls bergerak lebih cepat dari bilik kiri ke bilik kanan.

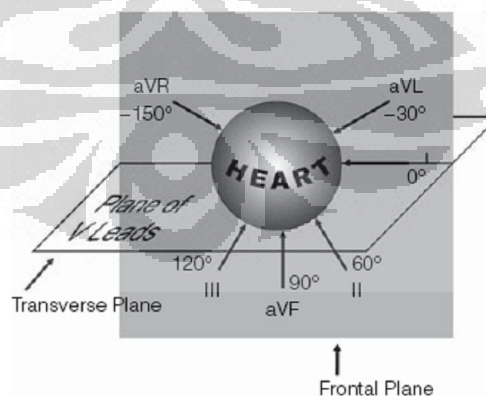
Vektor 3 merepresentasikan penjumlahan vektor depolarisasi pada bagian terbesar dari otot *ventricular* dan kemudian memunculkan gelombang QRS. Vektor menikung tajam ke sebelah kiri dengan sudut 60° .

Vektor 4 merupakan jumlah gaya listrik dikarenakan repolarisasi di vertikal dan digunakan untuk menghasilkan gelombang T. Dikarenakan repolarisasi lebih lambat dari pada polarisasi, gelombang T tidak memiliki ketebalan yang sama dengan gelombang QRS⁵.

2.1.5 LEAD¹

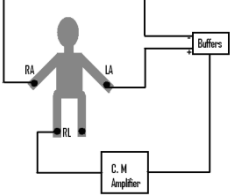
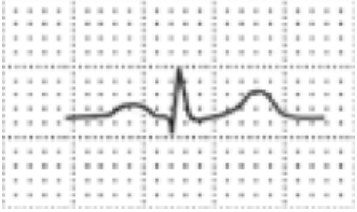
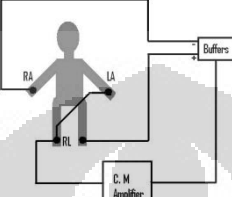

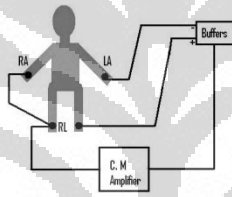
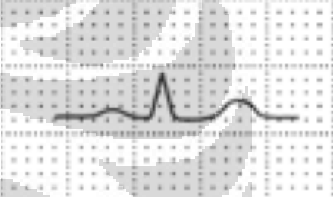
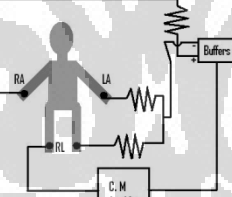
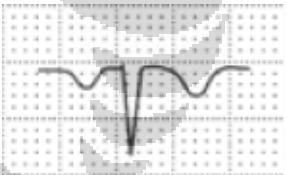
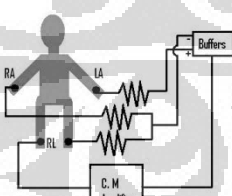
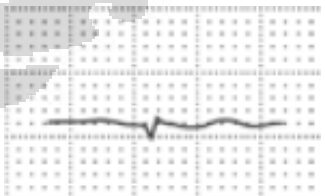
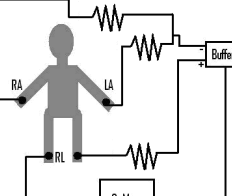
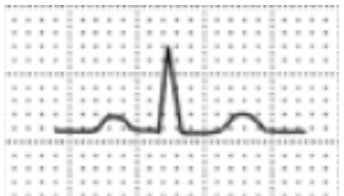
Dalam standar EKG terdapat 5 buah elektroda yang menempel di tubuh pasien, tangan kanan (RA), tangan kiri (LA), kaki kanan (RL), kaki kiri (LL) dan dada (Chest). Elektroda terhubung dengan sinyal input yang akan dikuatkan oleh rangkaian amplifier dengan beberapa pilihan penguatan.

Proses perekaman memerlukan sepasang elektroda yang menghasilkan keluaran berbeda yang menjadi input pada rangkaian elektrokardiogram. Dalam proses perekaman sinyal detak jantung terdapat 12 buah lead, yaitu I, II, III, AVR (augmented vector right), AVF (*augmented vector foot*), AVL (*augmented vector left*) serta 6 buah lead yang terhubung di dada¹.



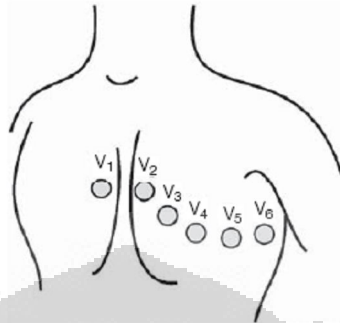
Gambar 2.5. Cardiac axis dari beberapa Lead¹

Bipolar limb leads diindikasikan dengan lead I, II, dan III yang sering dikenal dengan *Einthoven triangle*.

| Lead | Konfigurasi | Bentuk output |
|----------|---|---|
| Lead I |  |  |
| Lead II |  |  |
| Lead III |  |  |
| Lead AVR |  |  |
| Lead AVL |  |  |
| Lead AVF |  |  |

Tabel 2.1. Beberapa konfigurasi Leads⁵ (sudah diolah).

Selain ke-enam *lead* di atas terdapat enam buah *lead* yang terpasang pada dada.

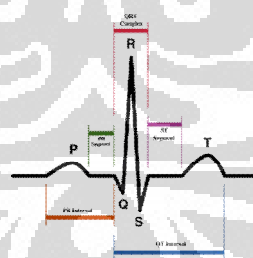


Gambar 2. 6 Lead yang berada di dada⁵

Beberapa *lead* tersebut dapat dijadikan pilihan dalam pengambilan data dan bentuk dari salah satu *lead* tersebut dapat dijadikan acuan dan sebagai pembandingan dengan beberapa lead lainnya.

2.1.6 EKG WAVEFORM

Elektrokardiogram *waveform* merupakan bentuk gelombang yang dihasilkan dari pendeteksian denyut jantung. Gelombang yang dihasilkan sangat khas.



Gambar 2.7 Waveform EKG

Beberapa alat EKG langsung menyalin berkas dari detak jantung secara normal yang terdiri dari beberapa bagian. Berdasarkan gambar di atas setiap satu gelombang terdapat 4 buah bagian yaitu gelombang P, gelombang QRS, gelombang T dan gelombang kecil U. Garis dasar tegangan yang menjadi dasar dari elektrokardiogram merupakan garis yang disebut sebagai *isoelectric line*. Biasanya garis isoelektrik terukur sebagai bagian dari berkas yang mengikuti gelombang T dan diteruskan dengan gelombang P selanjutnya¹.

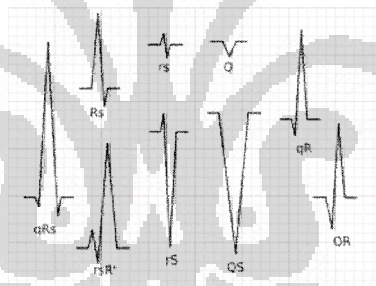
(1) P Wave

Selama proses depolarisasi pada *atrial*, vektor listrik utama mengalir langsung dari SA terhadap AV dan menyebar dari atrium kanan ke atrium kiri. ketika semua aktivitas listrik bergerak terhadap elektroda positif dalam beberapa lead, gelombang P dihasilkan. Gelombang ini bergerak naik dan berbelok ke kanan layaknya gelombang sinus positif. Hal itu membuat gelombang P merupakan sebuah ritme yang sering disebut dengan *Sinus Rhythm*.

(2) P – R Interval

Interval P-R terukur dari permulaan gelombang P sampai permulaan gelombang kompleks QRS.

(3) Gelombang Komplek QRS



Gambar 2.8. Beberapa bentuk gelombang kompleks QRS

Gelombang kompleks QRS merupakan sebuah struktur dari EKG yang menandakan adanya depolarisasi dalam ventrikel. Dikarenakan ventrikel berisi otot yang lebih banyak daripada atrial, gelombang kompleks QRS akan lebih besar daripada gelombang P. Selain itu, dikarenakan koordinat dari sistem *purkinje* mengalami depolarisasi dari *ventrikel*, gelombang kompleks QRS bergerak naik dengan cepat dikarenakan naiknya kecepatan konduktivitas di *ventrikel*.

Sebuah gelombang kompleks QRS normal biasanya berkisar 0.06 sampai 0.10s (60 – 100 ms). Adanya ketidaknormalan yang terjadi akan menyebabkan adanya pelebaran gelombang kompleks QRS.

Tidak semua gelombang kompleks QRS berisi gelombang Q, gelombang R dan gelombang S. Hal itu dipengaruhi oleh positif atau negatifnya pembelokan yang terjadi.

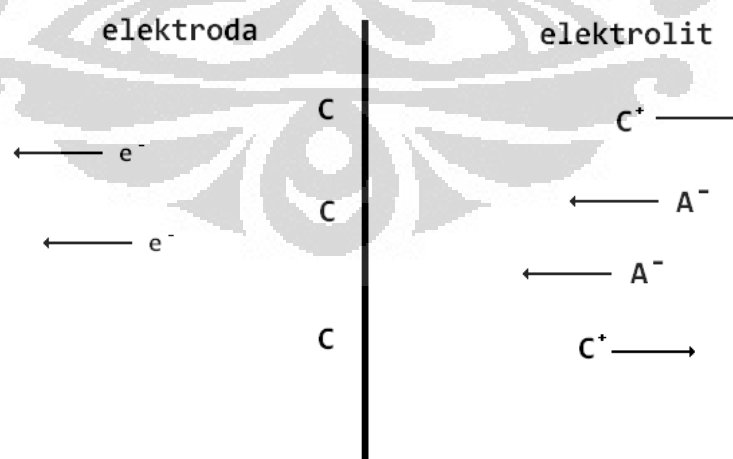
2.2 SISTEM ELEKTRONIKA

Rangkaian untuk mendeteksi aktifitas jantung terdiri dari sebuah rangkaian penguat sinyal berupa instrumentasi amplifier. Selanjutnya untuk membersihkan sinyal dari gangguan noise digunakan sebuah rangkaian filter aktif. Gelombang EKG dikuatkan dengan sebuah instrumentasi amplifier AD620AN. Untuk menghilangkan derau yang berasal dari rangkaian yang dipakai, Penulis menggunakan *low-pass filter* 10 Hz menggunakan IC OP 07.

2.2.1 BIOELEKTRODA

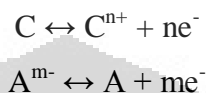
Untuk dapat mendeteksi sinyal potensial aksi yang dihasilkan oleh jantung perlu digunakan sebuah transduser. Transduser yang digunakan untuk mendeteksi potensial aksi EKG pada penelitian ini adalah bioelektroda Ag/AgCl. Transduser ini bersifat sekali pakai. Untuk sekali pengambilan data potensial aksi diperlukan minimal 3 buah bioelektroda Ag/AgCl.

Bioelektroda Ag/AgCl yang digunakan dibangun berdasarkan antarmuka elektroda-elektrolit seperti pada gambar 2.9.



Gambar 2.9. Antarmuka elektroda-elektrolit (sudah diolah)⁹

Pada gambar 2.9, arus yang bergerak dari elektroda ke elektrolit memiliki 3 komponen, yaitu elektron yang bergerak dengan arah yang berlawanan terhadap arus, kation yang bergerak searah arus (C^+) dan anion yang bergerak berlawanan arah arus (A^-). Konduksi elektron akan terjadi pada elektroda dan konduksi bionic akan terjadi pada elektrolit. Dikarenakan hal itu akan terjadi sebuah reaksi kimia seperti di bawah ini.

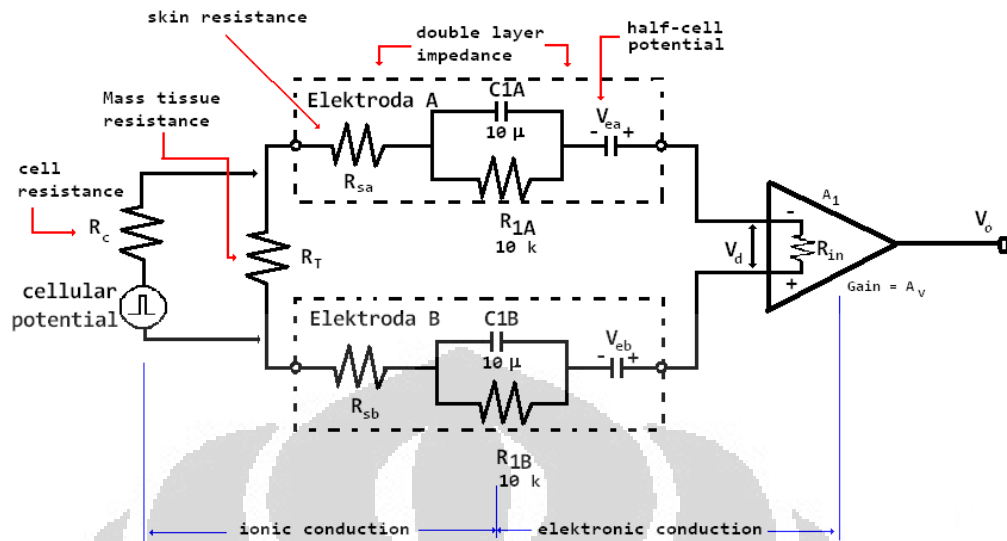


Dari kedua persamaan di atas, reaksi oksidasi lebih dominan dari pada reaksi reduksi.

Elektroda yang dipakai pada penelitian ini adalah Ag/AgCl. Elektroda ini merupakan elektroda *reversible*. Pada saat elektroda bersentuhan dengan cairan elektrolit yang mengandung anion dan kation, reaksi pertama akan terjadi. Konsentrasi lokal kation dan anion berubah dan menyebabkan kesetimbangan muatan pada daerah tersebut tidak sama. Hal itu menyebabkan munculnya beda potensial pada elektroda. Beda tegangan itu disebut *Half-cell potential*.

Jika dua elektroda ditempatkan pada dua garis potensial yang berbeda maka akan muncul beda tegangan. *Vector Lead* akan membangkitkan tegangan maksimum dari sepasang elektroda. Tegangan aksi yang dihasilkan oleh jantung akan terdeteksi oleh elektroda dengan memanfaatkan proses oksidasi dan reduksi yang terjadi di elektroda dan elektrolit.

Pada gambar 2.10, menampilkan sebuah model rangkaian dari biomedical elektroda. Differensial amplifier digunakan untuk memproses data yang dihasilkan dan akan menghilangkan efek dari potensial half-cell yang dihasilkan oleh elektroda (V_{ea} dan V_{eb}). R_T merupakan hambatan internal dari badan yang umumnya sangat kecil. Sinyal *bio-potential* merupakan perbedaan tegangan yang dihasilkan (V_d). Ketika dua atau lebih elektroda digunakan bersama-sama untuk merekam data medis, maka tegangan antara keduanya merupakan penjumlahan dari keduanya secara aljabar⁹.

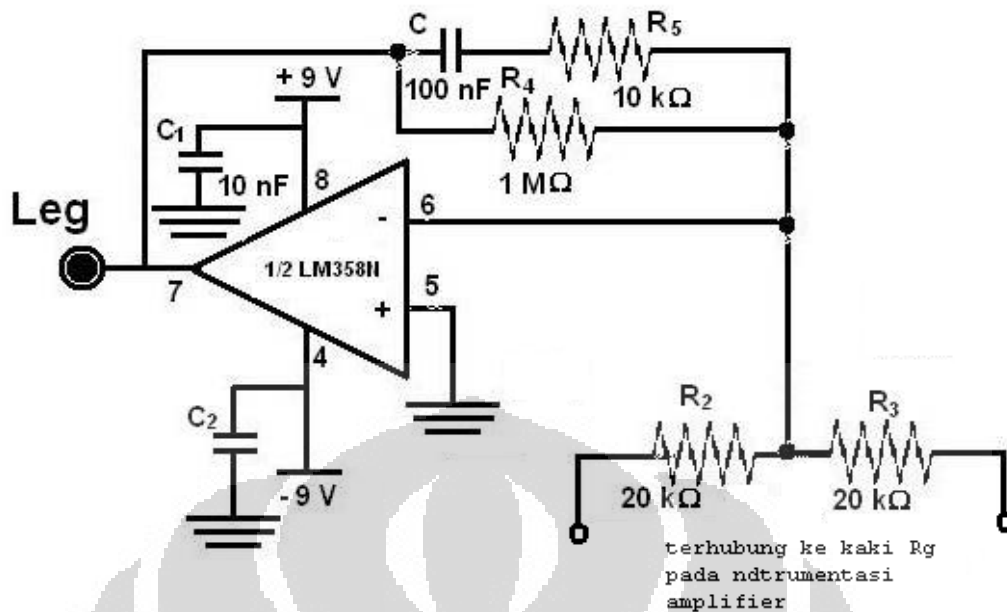


Gambar 2.10. model rangkaian untuk elektroda medis⁹.

2.2.2 SISTEM PENGGERAK KAKI KANAN

Sistem penggerak kaki kanan merupakan sebuah rangkaian elektronik yang dirancang untuk mengurangi interferensi oleh tegangan jala-jala akibat kopling kapasitor. Sistem penggerak kaki kanan akan menghasilkan arus ke ground melalui tubuh objek. Pada rangkaian yang dipakai terdapat sebuah op-amp (1/2 LM358AN) yang dipakai untuk tujuan ini.

Selain itu, sistem ini bertujuan untuk mereduksi noise. Hal itu dikarenakan kita memberikan sebuah hambatan *feedback*. Pada skematik yang dipakai kita menambahkan sebuah rangkaian filter aktif di sistem ini dengan menaruh sebuah kapasitor yang dihubungkan seri dengan sebuah resistor yang dijadikan sebuah umpan balik.



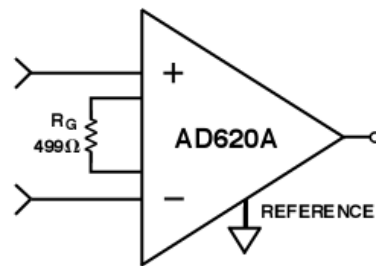
Gambar 2.11 skematik penggerak kaki kanan

Sistem penggerak kaki kanan akan memberikan proteksi pada output dari instrumentasi amplifier yang digunakan ketika terjadi penghantaran pulsa tegangan yang tinggi⁹. Sebenarnya kondisi ketika munculnya tegangan tinggi sangat jarang terjadi. Ketika hal itu terjadi op-amp yang dipakai akan rusak dan tegangan tidak akan mencelakai pasien, karna pasien secara langsung akan terhubung ke ground.

2.2.3 INSTRUMENTASI AMPLIFIER

Untuk menguatkan nilai potensial aksi dibutuhkan sebuah rangkaian penguat. Ditambah adanya noise yang muncul dari rangkaian maupun dari ketidakstabilan pemakaian alat. Pada rangkaian ini digunakan sebuah op-amp AD620AN. Op-amp ini memiliki penguatan yang cukup besar dengan range 1 sampai 1000 kali (datasheet AD620AN terdapat pada lampiran).

Input yang diberikan oleh sinyal aktifitas jantung dikuatkan hingga mencapai 10 kali. Hal itu membuat sinyal input potensial aksi jantung dapat terdeteksi oleh osiloskop analog.

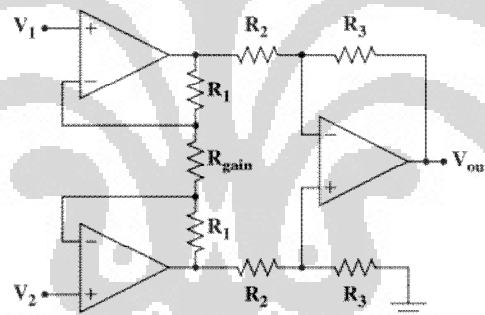


Gambar 2.12 Skematik AD620AN dengan penguatan 100

Nilai penguatan dapat diubah dengan mengubah nilai R_G dengan mengikuti persamaan di bawah ini :

$$G = \frac{49.4k\Omega}{R_G} + 1 \quad (2.2)$$

atau



Gambar 2. 13. Instrumentasi Amplifier

Instrumentasi amplifier merupakan sebuah tipe differensial amplifier dengan sebuah input *buffer* yang berfungsi untuk menghilangkan impedansi input yang muncul. Rangkaian ini memiliki karakteristik, antara lain nilai offset DC yang sangat kecil, *low drift*, *low noise*, penguatan *open-loop* yang sangat besar, CMRR yang besar dan impedansi input yang besar⁷.

Pada gambar di atas, instrumentasi amplifier secara skematik berisi 3 buah op-amp yang identik. Biasanya penguatan *common-mode* pada instrumentasi *amplifier* adalah nol⁶. Nilai tersebut muncul dikarenakan adanya ketidaksetaraan nilai jumlah dari semua resistor yang berbeda dan nilai input yang tidak nol.

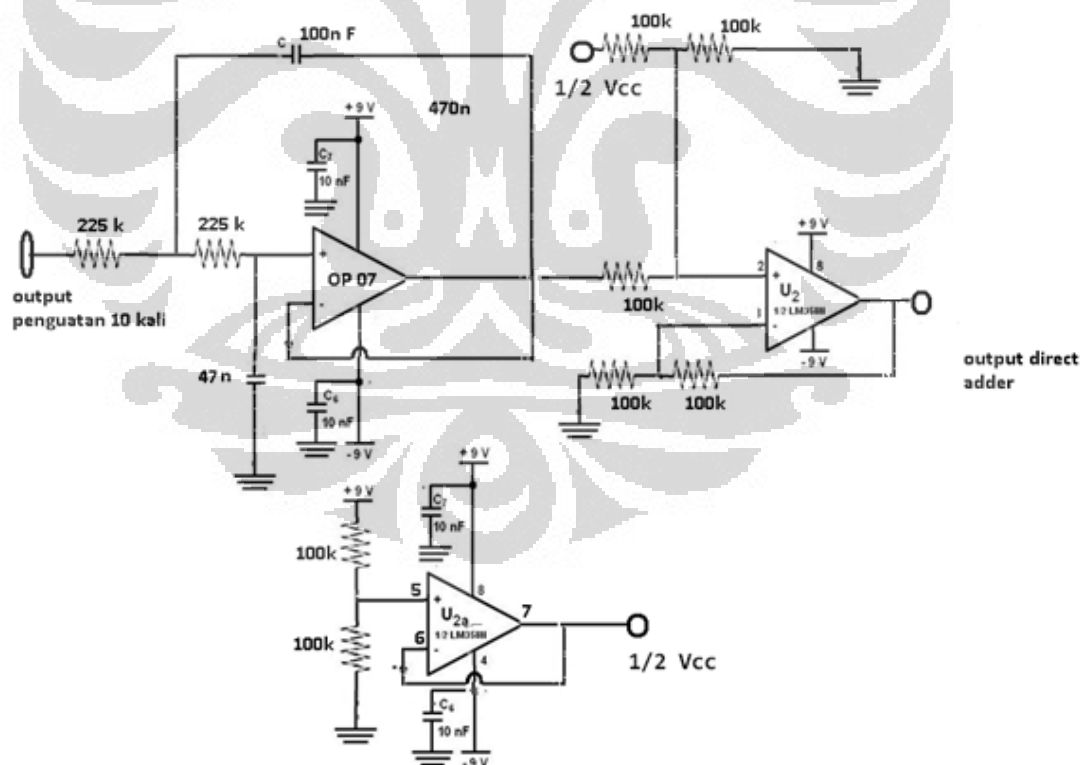
2.2.4 FILTER AKTIF⁶

Filter aktif merupakan rangkaian yang terdiri dari sebuah op-amp, resistor dan kapasitor⁶. Rangkaian filter aktif digunakan untuk mereduksi noise yang muncul. Konsep dasar dari rangkaian ini adalah memotong dan memblok pada frekuensi tertentu. Sinyal input yang diblok ditentukan dari besarnya kombinasi dari nilai resistor dan kapasitor yang dipakai.

Nilai untuk satu gelombang EKG berkisar antara 0.05 Hz sampai dengan 100 Hz. Untuk menghitung besarnya nilai frekuensi dapat digunakan persamaan :

$$f_c = \frac{1}{2\pi.R.C} \quad (2.3)$$

Penelitian ini menggunakan sebuah rangkaian low-pass filter 10 Hz. Adapun skematik lengkap filter low-pass yang digunakan tertera pada gambar 2.14.



Gambar 2. 14. Scematik filter aktif beserta direct adder yang digunakan(sudah diolah)

Dengan mengambil nilai $R_1 = R_2 = R$ dan $C_1 = 100 \text{ nF}$ dan $C_2 = 47 \text{ nF}$ akan didapatkan nilai $R = 225 \text{ k}\Omega$. Untuk mengatur tingkat kecuraman dan jenis respon filter yang digunakan, kita dapat mengatur nilai α pada filter aktif.

2.2.5 ATmega 324 p

Mikrokontroler ATmega 342p merupakan mikrokontroler cmos buatan ATMEL. Mikrokontroler ATmega 324p memiliki pada penelitian ini digunakan sebagai ADC. ADC yang dimiliki ATmega 342P memiliki konfigurasi antara lain :

- 8 Channel input.
- Akurasi 10 bit.
- Akurasi absolute $\pm 2 \text{ LSB}$.
- Waktu konversi $13 - 260 \mu\text{s}$.
- Resolusi maksimum mencapai 15 kSPS.
- Jangkauan tegangan masukan $0 - V_{cc}$.
- Memiliki 3 buah tipe tegangan referensi internal, yaitu 1.14 V, 2.56 V dan sama dengan V_{cc} -nya.
- Pengurangan noise pada mode sleep.

Pada penelitian ini ATmega342p hanya dipakai sebagai sebuah konverter. Keluaran digital yang dihasilkan oleh ADC yang terdapat pada ATmega 342p akan diolah oleh TINI. Dikarenakan keluaran ADC hanya 10 bit, maka kisaran bilangan desimal (hex) yang diterima oleh TINI berkisar antara 0 sampai 1024.

Untuk pemrogramannya digunakan Bascom. Adapun list program yang digunakan adalah sebagai berikut.

```
$regfile = "m324pdef.dat"
$crystal = 8000000
$hwstack = 32
$swstack = 10
$framesize = 40
```

```
Open "comd.3:19200,8,n,1" For Output As #1
```

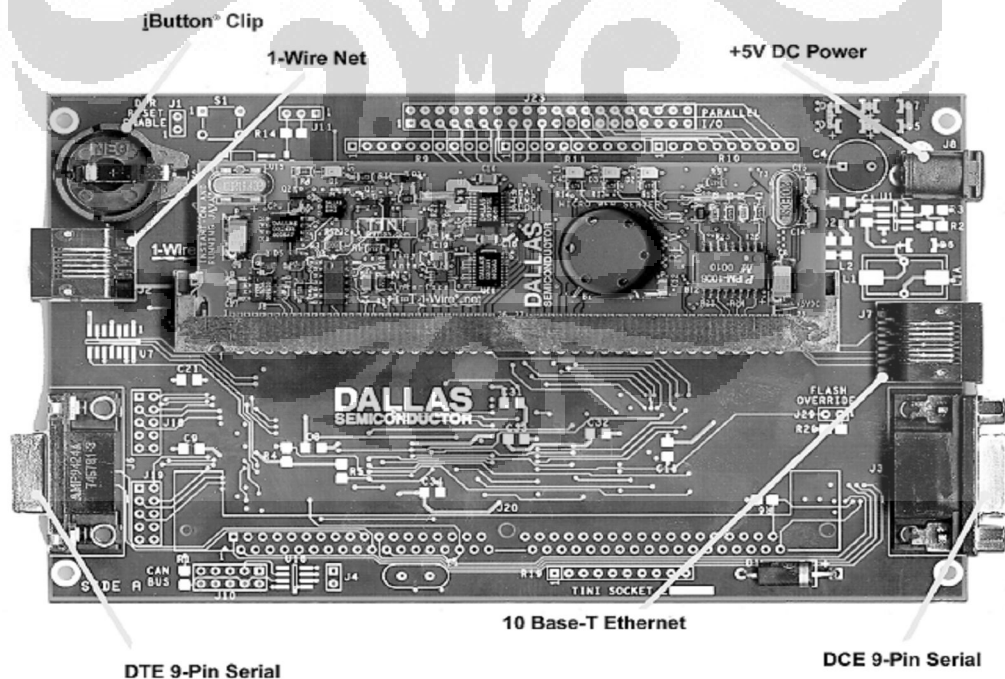
```

Config Adc = Single , Prescaler = Auto , Reference = Avcc
Start Adc
Do
Print #1 , Getadc(0)
Waitms 100
Loop
End

```

2.3 TINI MIKROKONTROLLER

Tiny InterNet Interface atau TINI merupakan sebuah mikrokontroller buatan Dallas Semikonduktor yang difokuskan pada perancangan sistem dan pengembangan software yang sederhana, *flexible*, dan memiliki efisiensi yang bisa ditekan seminimal mungkin dengan cara memvariasikan serta mengkoneksikannya dengan sistem jaringan². Untuk memprogram TINI menggunakan bahasa pemrograman Java. *Chip-set* yang dimiliki TINI dikembangkan untuk tujuan prosesing, kontrol, dan komunikasi dan jaringan dari beberapa media, mulai dari komunikasi serial hingga *Local Area Network* (LAN).



Gambar 2.15. TINI Board

Untuk saat ini, dengan melihat beberapa keunggulan yang dimiliki oleh TINI, mikrokontroler ini merupakan sebuah *device* untuk pendistribusian data. Hal itu dapat dilihat dari besarnya memori yang dimiliki oleh TINI².

TINI mikrokontroler memiliki fungsi utama yaitu *networking*. Beberapa fungsi yang dapat dikerjakan dengan menggunakan TINI antara lain² :

a) *Industrial control*

Fasilitas CAN (*Controller Area Network*) yang dimiliki TINI mendukung sebuah aplikasi otomatisasi sistem industri. TINI dapat digunakan untuk menggerakkan *switch*, *attenuator*, dan lain-lain.

b) *Web-based equipment monitoring and control*

Untuk dapat dan mengontrol suatu proses dapat dilakukan dengan menjalin komunikasi antara peralatan yang digunakan dengan TINI. Untuk mengendalikan dan menganalisa data dapat menggunakan fasilitas *Ethernet*. Penganalisaan data dapat dilakukan dengan web.

c) *Protocol conversion*

TINI didasarkan pada sebuah sistem yang dikoneksikan dengan *Ethernet Networking*. Hal ini tergantung dari kelayakan dari I/O yang digunakan pada sistem. Dengan menggunakan TINI yang dikoneksikan dengan computer desktop pengguna dapat meminimalkan pengeluaran dan biaya.

d) *Environmental monitors*

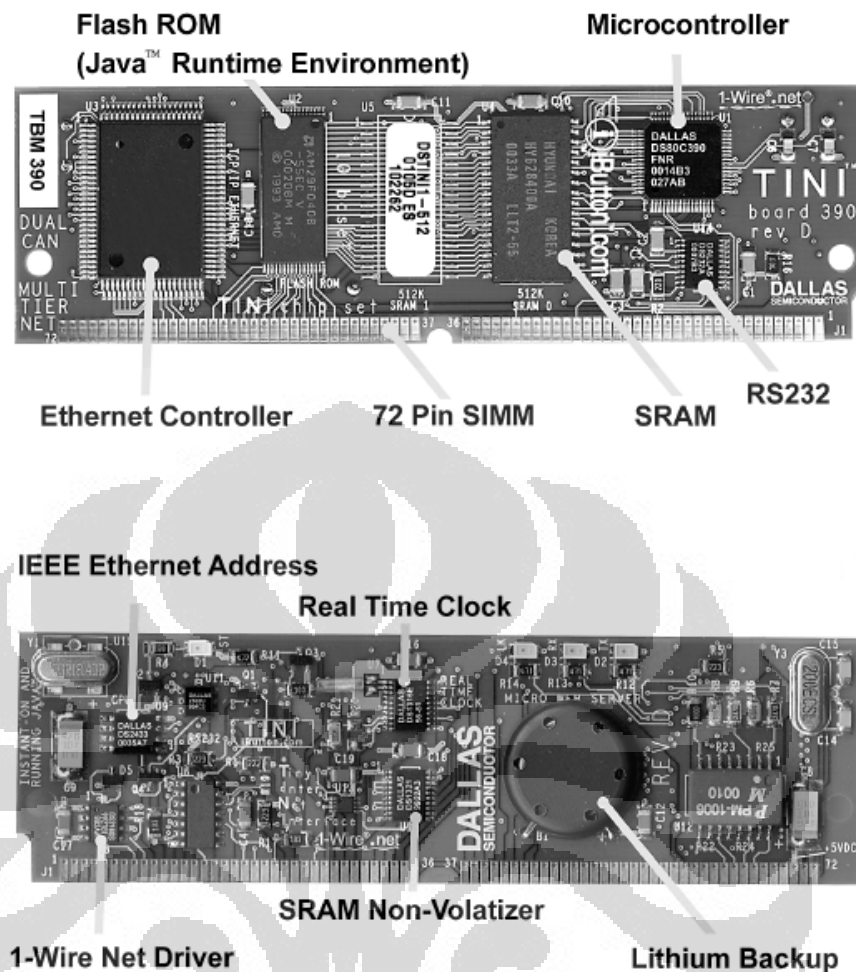
Dengan menggunakan fasilitas *i-Wire*, sistem dapat dikendalikan dan hasil yang didapatkan dari setiap pengguna dapat diamati.

2.3.1 HARDWARE

Sistem hardware yang dipakai merupakan rangkaian proses penghantaran data yang diterima mulai dari rangkaian elektrokardiogram itu sendiri (telah dibahas dalam Bab 2), *analog to digital converter* (yang berada dalam AVR) serta TINI dengan PC.

Tiny InterNet Interface (TINI) merupakan sebuah mikrokontroler yang memiliki beberapa fasilitas *hardware* antara lain² :

- Konektor 72-pin SIMM



Gambar 2.16. 72-pin SIMM connector

TBM390 merupakan sebuah contoh 72-pin SIMM *connector* yang digunakan oleh TINI E10. SIMM tipe ini sudah siap untuk dijadikan sebagai *hardware* secara Ethernet dan dapat mendukung beberapa fungsi, seperti 1-Wire, CAN, serial, dan lain-lain.

- Konektor 9-pin *female* DB-9
Konektor ini merupakan sebuah media komunikasi yang dimiliki mikrokontroler untuk komunikasi serial.
- Konektor 9-pin *male* DB-9
Konektor ini memiliki fungsi sama dengan konektor 9-pin *female* DB-9 sebagai media komunikasi secara serial. Konektor 9-pin *female* DB-9 merupakan kebalikan konektor dengan 9-pin *female*. Konektor ini

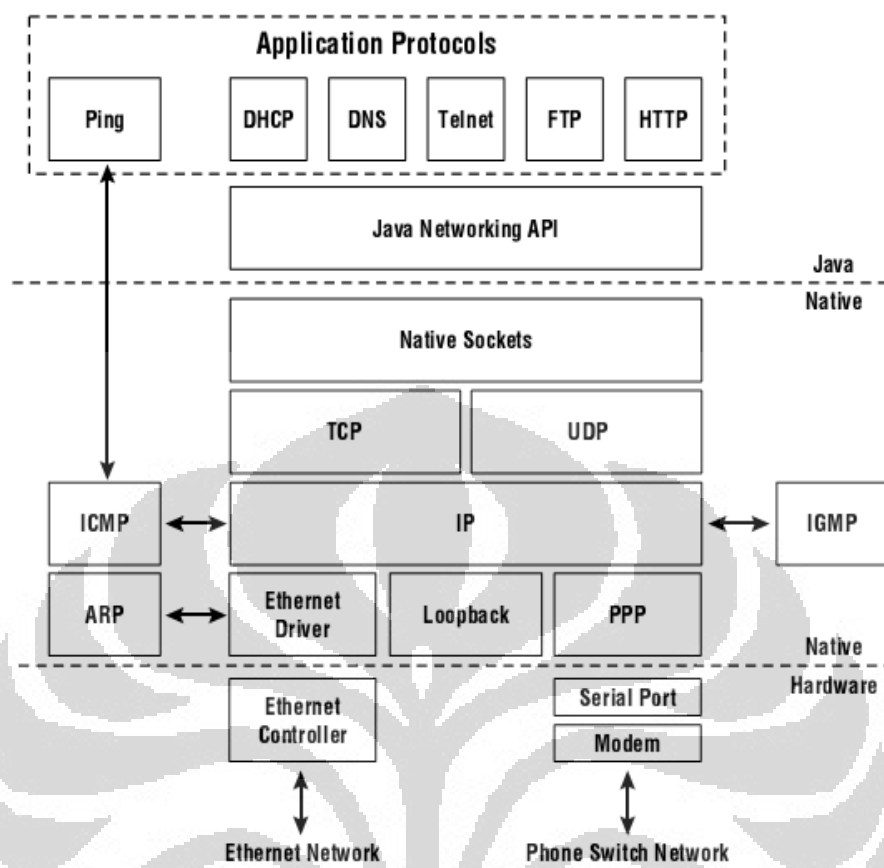
menggunakan fasilitas DTE yang akan dihubungkan dengan DCE media lain, seperti *analog modem*.

- **RJ45**
Konektor RJ45 merupakan standar dari *10-Base Ethernet* yang dikembangkan pada TINI. Untuk dapat menggunakan ethernet dapat digunakan sebuah *straight-through cable* untuk menghubungkan TINI dengan sebuah jaringan atau dengan sebuah *crossover cable* jika ingin langsung mengkomunikasikan TINI dengan sebuah komputer.
- **RJ11**
Konektor RJ11 merupakan fasilitas terbaru yang dimiliki TINI, yaitu 1-Wire. Untuk dapat menggunakan fasilitas ini dapat digunakan kabel telepon.
- **Power Jack**
Power yang dibutuhkan oleh mikrokontroler ini sebesar + 5 V.

2.3.2 ETHERNET (TCP/IP NETWORKING)

Sistem pengolahan data akan dilakukan di TINI. TINI melakukan 2 buah kerja yaitu menerima data dari AVR dan berkomunikasi dengan PC melalui ethernet. TINI dikembangkan untuk menjalankan beberapa aplikasi *non-ethernet* menjadi *ethernet*. *Ethernet* merupakan aplikasi yang dimiliki TINI agar mampu beroperasi dalam ruang lingkup LAN (*Local Area Networking*).

Gambar di bawah merupakan diagram ruang lingkup kerja *networking*. Terdapat enam lapisan protokol yang harus diketahui, yaitu HTTP (*Hypertext Transfer Protokol*), DNS (*Domain Name System*), DHCP (*Dynamic Host Configuration Protokol*), Telnet, FTP (*File Transfer Protokol*), Ping (ICMP *echo request/reply*). Keenam protokol tersebut terangkum dalam *Java Networking API*. Semua protokol tersebut dapat dibaca Java dan dapat menjalankan TINI tanpa harus merubah *source code*.



Gambar 2.17 TINI Runtime Environment

Proses *interface* terjadi dalam tiga tipe, ethernet, *PPP over a serial link* dan *loopback*. Untuk dapat mengetahui konfigurasi dari masing-masing antarmuka, asumsikan semua bermula dari angka 0. *Ethernet* merupakan antarmuka 0 (*eth0*), *loopback* merupakan antarmuka 1 (*eth1*) serta *PPP over a serial link* akan menjadi antarmuka 2 dan 3. Untuk dapat mengetahui pernyataan dari semua antarmuka dapat digunakan perintah "i pconfig -x".

```
Interface 0 is active
Name       : eth0 (default)
Type       : Ethernet
IP Address  : 192.168.0.15
Subnet Mask : 255.255.255.0
Gateway    : 192.168.0.1
```

Interface 1 is active

Name : lo
 Type : Local Loopback
 IP Address : 127.0.0.1
 Subnet Mask : 255.0.0.0
 Gateway : 0.0.0.0

Interface 2 is not active

Interface 3 is not active

Tampilan dapat diubah secara programming menggunakan metode TINI Net' saddl nterfaceEntry dan RemoveI nterfaceEntry.

Ethernet menggunakan sebuah konektor LAN untuk mengirim dan menerima pesan. Semua ethernet memiliki 48-bit, yang dikenal sebagai alamat *ethernet*. Alamat *ethernet* disimpan dalam EPROM. Alamat tersebut tidak pasti tetapi dapat diprogram. Alamat ethernet dapat juga digunakan untuk mengidentifikasi TINI ketika beroperasi.

2.3.3 KOMUNIKASI SERIAL

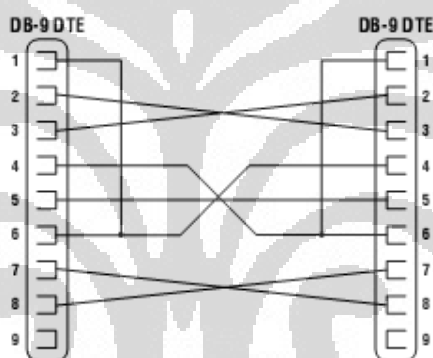
Sistem komunikasi serial yang dikembangkan di TINI merupakan sistem serial DTE dan DCE. Untuk beberapa aplikasi serial, TINI bertindak sebagai jembatan komunikasi untuk DCE dan beberapa media yang menggunakan sistem terminal data.

| pin # | Signal Name | DTE Sense | Description |
|-------|---------------------------|-----------|--|
| 1 | CD (Carrier Detect) | INPUT | Asserted by DCE when it has received data carrier signal |
| 2 | RD (Receive Data) | INPUT | Data receive from DCE |
| 3 | TD (Transmit Data) | OUTPUT | Data transmit to DCE |
| 4 | DTR (Data Terminal Ready) | OUTPUT | Asserted by DCE when it has ready to communication |

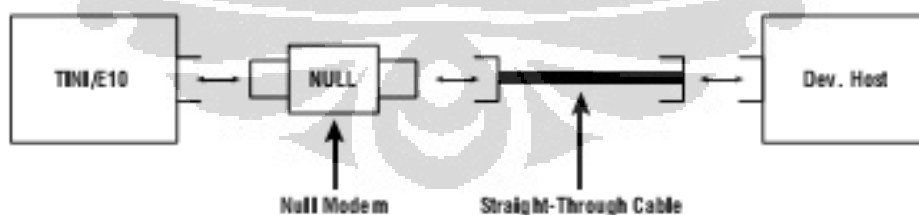
| | | | |
|---|------------------------|--------|--|
| 5 | Common (Signal Ground) | N/A | 0 Volt reference |
| 6 | DSR (Data Set Ready) | INPUT | Asserted by DCE when it has established a communication channel and is ready to transmit |
| 7 | RTS (Request To Send) | OUTPUT | Asserted by DCE to request permission to transmit data |
| 8 | CTS (Clear to Send) | INPUT | Asserted by DCE to grant a permission to DTE to transmit data |
| 9 | RI (Ring Indicator) | INPUT | Asserted by DCE when it receive a ringing tone |

Tabel 2.2. Beberapa fungsi pin DB-9 yang digunakan dalam TINI

Komunikasi serial pada DTE menggunakan sistem *null modem*. Hal itu dikarenakan ketika menggunakan kabel *straight* biasa tidak dapat diterima oleh TINI.



Gambar 2.18 Konfigurasi serial *null modem*



Gambar 2.19 Hubungan *server* (TINI) dengan *host*

Untuk memulai pengolahan data dengan komunikasi serial, harus diperhatikan alur yang terjadi. *Flow control* merupakan mekanisme yang diperbolehkan penerima memberi pesan pada pengirim untuk menunda

pengiriman ketika data internal yang diperoleh sudah terisi penuh. Adapun beberapa flow mode yang disediakan oleh comm API antara lain :

- None
- RTS/CTS
- XON/XOFF

Ketika tak terdapat *flow control*, kedua sisi komunikasi akan mengirimkan apa yang harus dikirimkan. Ini tak begitu bermasalah, tergantung protokol yang digunakan. Ketika hanya terjadi komunikasi satu arah secara terus-menerus, penerima harus menyediakan beberapa pilihan dengan menyaring data yang diterima atau dengan resiko kehilangan data.

Untuk sistem XON/XOFF, ketika sebuah penerima (sebut penerima A) mulai mencapai kapasitasnya, ia akan mengirim sebuah pesan XOFF (0x13) kembali kepada si pengirim (sebut pengirim B) untuk menunda pengiriman data. Setelah semuanya sudah kembali normal, penerima A akan mengirimkan sebuah pesan XON (0x11) yang menyatakan bahwa B dapat mengirim kembali. Aplikasi XON/XOFF memiliki keuntungan dengan tidak membutuhkan adanya komunikasi jabat tangan antara dua pihak (*handshake lines*).

RTS/CTS merupakan jalan terbaik untuk mencegah adanya overdosis data. Untuk aplikasi ini, diperlukan beberapa hardware. Untuk dapat mengirimkan data sebuah hardware A harus menghidupkan sinyal RTS (*Request To Send*). Jika hardware lainnya (sebut B) menyediakan ruang pada buffernya dan bersiap untuk menerima data, *hardware* B akan mengirim dan menghidupkan sinyal CTS pada *hardware* A dan A mulai mengirim data.

Biasanya pada beberapa aplikasi tidak menyediakan *flow control*. Hal itu dikarenakan banyak dari aplikasi tidak didukung oleh adanya hardware dan software *flow control*.

Pada komunikasi serial terdapat beberapa hal yang harus diperhatikan. Untuk dapat mengirim dan menerima data, harus diperhatikan beberapa karakteristik pada komunikasi serial, antara lain :

- *Baud rate*
- Jumlah bit data
- Jumlah stop bit

- Tipe dari *parity*
- *Flow control*

Pada TINI, pengaturan standart adalah 115.000³ bps dengan ketersediaan 8-bit, 1 stop bit, tidak terdapat *parity bit* dan tidak disediakan *flow control*.

2.3.4 TINI KERNEL²

Untuk mengoperasikan TINI diperlukan beberapa perangkat lunak seperti Java Development Environment (NetBeans 6.5), TINI Software Development Kit dan Java Communication API. NetBeans digunakan untuk mengolah data yang didapatkan dari ADC.

Untuk dapat mengakses dan menggunakan TINI, diharuskan untuk membuat kernel pada TINI. Setelah itu, harus mengkonfigurasi PATH dan CLASSPATH dan menyusun semua kelengkapan alat.

Untuk dapat menjalankan *system* dengan Java SDK diperlukan beberapa hal, antara lain:

- *tini.jar*
- *tiniclasses.jar*
- *tini.db*
- *tbin files*

Semua bagian di atas sudah termasuk ke dalam sebuah paket yang diberikan oleh Dallas Semiconductor. Ketika semua program yang dibutuhkan telah selesai diinstal, instal *runtime environment* pada TINI. Untuk dapat menginstal program pada TINI dibutuhkan tiga hal, yaitu kita file *tini.tbin*, *slush.tbin* serta mengidentifikasi *heap*. Ketiga proses dilakukan dengan Javakit.

Untuk memulai Javakit, TINI dan komputer harus terhubung dengan *straight-through serial*. Untuk mengaktifkan ketik perintah seperti dibawah ini pada *command prompt*.

```
c:\jdk1.2.2\bin\javaw -classpath c:\tini1.02\bin\tini.jar Javakit
```


Ketika yang dilakukan berhasil akan muncul tampilan seperti pada gambar 3.6. Gunakan komunikasi serial dengan memilih “COM1” pada pilihan “PORT NAME”, kemudian tekan “OPEN PORT”.

Ketika operasi gagal dilakukan akan muncul pesan yang menyatakan bahwa operasi gagal, untuk itu harus mengecek kembali koneksi komputer dengan TINI. Ketika operasi berhasil dilakukan akan muncul tampilan seperti di bawah ini :

TINI loader 15-08-08 17.45

Copyright (C) 2008 Dallas Semiconductor. All right reserves

>



Gambar 2.20. Aplikasi JavaKit

Setelah itu harus dibuka 2 buah file seperti `tini.tbin` dan `slush.tbin` dengan memilih “LOAD FILE” pada menu “FILE”. Ketika selesai akan muncul tampilan seperti dibawah ini :

```

Loading file: C:\tini1.02\bin\slush.tbin.
Please wait ... (ESC to abort)
Load complete.

```

```

Loading file: C:\tini1.02\bin\tini.tbin
Please wait ... (ESC to abort)
Load complete.

```

Kemudian kosongkan *heap* dengan melakukan hal di bawah ini :

```

>BANK 18
>FILL 0

```

Ketika semua telah di-load secara benar, kemudian kita tekan “*enter*” untuk mengeksekusi dan akan muncul tampilan di bawah ini :

```

----> TINI Boot <----
API Version 8006
Copyright (C) 1999, 2000 Dallas Semiconductor Corporation
01000000
Doing First Birthday
Memory Size: 07E600
Addresses : 181A00, 200000
Skip List MM

L01
Running POR Code

```

Memory POR Routines

000020

Transient block freed : 0000, size : 000000

Persistent block freed : 0000, size : 000000

KM Init Passed

Ethernet MAC Address Part Found

TTS Revision : 154, Date : 7/19/00 3:13p

Thread Init Passed

External Serial Port Init

External serial ports not enabled

Memory Available : 075F00

Creating Task :

0100

01

Loading application at 0x070100

Creating Task :

0200

02

Application Task Complete

```

[== slush Version 1.01 ==]
[   System coming up.   ]
[   Beginning initialization... ]
[   Not generating log file.   ] [Info]
[   Initializing shell command... ] [Done]
[   Checking system files...   ] [Done]
[   Initializing and parsing startup... ]
[   Initializing network...   ]
[   Network configurations not set. ] [Skip]
[   Network configuration     ] [Done]
[   System init routines     ] [Done]

```

```

Hit any key to login.
Welcome to slush. (Version 1.01)
TINI login : root
TINI password :
TINI />

```

Terdapat dua buah *account* standard, isi “guest” dengan password “guest” dan “root” dengan password “tini”.

2.3.5 INISIALISASI SERIAL

Untuk mengirim dan menerima data dengan komunikasi serial pada TINI, digunakan metode `read` dan `write` pada daerah input dan outputnya. `java.io.InputStream` dan `java.io.OutputStream` diperlukan pada aplikasi ini.

```

public InputStream getInputStream() throws IOException
public OutputStream getOutputStream() throws IOException

```

Komunikasi serial yang disediakan oleh TINI adalah komunikasi 8 bit. Pada saat berkomunikasi, sering terdapat *overloaded* pada *buffer*. Untuk dapat memberikan *buffer* dengan memori yang telah ditentukan dapat digunakan metode `setInputBufferSize`

```

Public void setInputBufferSize(int size)
Public int getOutputBufferSize()

```

Untuk dapat mengaktifkan serial port terdapat beberapa hal yang harus diperhatikan. Di sini diharuskan untuk mengaktifkan *Event* yang akan dilakukan serial port dengan metode `addEventListener`.

```

public void addEventListener(SerialPortEventListener listener)

```

```

    Throws TooManyListenersException
    public void removeEventListener()

```

Setelah langkah di atas dapat serial port dapat diaktifkan dengan metode enableSerialPort.

```

public static final void enableSerialPort()
public static void setExternalSerialPortEnable(int portNum,
boolean enable)

```

Setelah hal tersebut dapat serial port dapat dikonfigurasi dengan beberapa langkah seperti di bawah ini :

```

private SerialPort sp;
private InputStream sin;
private OutputStream;

private Electrocardiogram(String portName, int baudRate)
    throws NoSuchPortException,
        PortInUseException,
        UnsupportedOperationException,
        IOException {
    try{
        // create SerialPort object for specified port
        sp = (SerialPort)
            CommPortIdentifier.getPortIdentifier(portName).open(
                "Electrocardiogram", 5000);
        //configuration for 8 databits, 1 stop bit and no
        parity
        sp.setSerialPortParams(baudRate, SerialPort.DATABITS_8,
            SerialPort.STOPBIT

```

2.3.6 INISIALISASI TCP/IP

Untuk dapat menginisialisasi *ethernet*, diharuskan mengeluarkan sebuah class yang nantinya digunakan untuk mengaktifkan *ethernet*.

```
Import com.dal.semi.tini.net.TINI.net;
```

Setelah hal tersebut, kita dapat mengaktifkan alamat ethernet yang akan digunakan.

```
Public static void getEthernetAddress(byte[] address)
Public static String getEthernetAddress()
```

Pada metode yang pertama, tanda kurung tutup yang berada dibelakang byte dapat diisi dengan beberapa byte array dengan alamat pada ethernet untuk dapat mengisi ruang 0th. Array akan berkisar 6 karakter atau lebih saat alamat ethernet adalah 48 bit. Metode kedua digunakan untuk mengaktifkan alamat *ethernet* dalam konteks string.

2.3.7 NETBEANS 6.5

Untuk dapat menjalankan sebuah aplikasi dengan TINI, diperlukan *Java Communication API* untuk dapat membuat sebuah rancangan program serta untuk mengkompile *source code* yang akan dimasukkan dalam kernel TINI.

Pada aplikasi ini digunakan sebuah bagian dari serangkaian aplikasi yang disediakan oleh TINI, yaitu Netbeans. Netbeans merupakan salah satu media pemrograman yang disediakan oleh SUN Corporation. Untuk aplikasi pemrograman NetBeans menyediakan media JDK (*J2SE Development Kit*).

Untuk merealisasikan source code yang sudah dibuat ke dalam bentuk program (.exe atau .jar) terdapat beberapa tahap. Beberapa tahap tersebut antara lain⁸ :

1. Membuat sebuah program dalam bentuk *source code*.

Pada bagian ini, programmer merancang *source code* sebuah aplikasi yang ingin dibuat dalam bentuk file berekstension .java. Netbeans

menyediakan beberapa *tool* yang dapat mempermudah dalam merancang sebuah *source code*.

2. Mengkompilasi *source* tersebut menjadi program berekstensi `.class`. Untuk dapat mengkompilasi sebuah file `.java`, diperlukan sebuah perintah `javac` (*Java Compiler*). Ketika mengkompilasi akan dihasilkan *file* dengan ekstensi `.class`. Untuk mengkonversi dapat digunakan perintah,

```
javac electrocardiogram.java
```

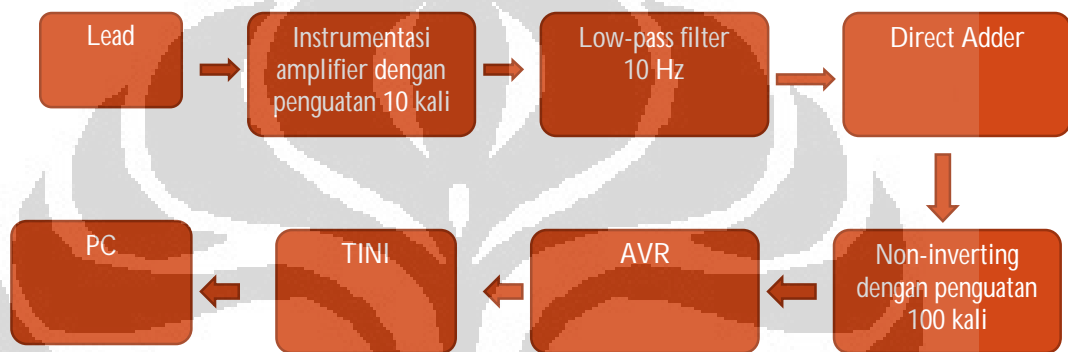
`electrocardiogram.java` merupakan *file source* yang telah dibuat dalam Netbeans.

3. Menempatkan program tersebut dari memori. Pada bagian ini, program harus diletakkan pada sebuah memori utama sebelum akhirnya dieksekusi. Dalam hal ini, kita dapat menggunakan sebuah *harddisk* dalam CPU atau mungkin *harddisk eksternal* untuk menyimpan *file* tersebut.
4. Memeriksa *bytecode*-nya. Sistem keamanan yang dimiliki Java akan memeriksa *bytecode* *file* `.class` tersebut, apakah *file* tersebut benar atau tidak.
5. Menjalankannya. Pada bagian ini, JVM (*Java Virtual Machine*) menjalankan *bytecode* yang dimiliki program. JVM membuat pengguna dapat melakukan sebuah aplikasi yang dirancang dalam program tersebut.

BAB 3 PERANCANGAN SISTEM

3.1 ALUR PENELITIAN

Penelitian ini memfokuskan pada perancangan sistem untuk dapat menampilkan gelombang EKG melalui devais ethernet. Adapun alur prose kerja yang dilakukan adalah sebagai berikut.



Gambar 3.1 Diagram alur sistem penelitian

secara umum, penelitian ini terbagi dalam 2 proses, yaitu perancangan hardware dan software. Sistem hardware bermula dari pendeteksian data aktifitas listrik oleh bioelektroda hingga oenguatan 100 kali non inverting. Adapun perancangan software berawal dari perancangan ADC hingga penampilan data di PC.

Instrumentasi amplifier digunakan untuk menguatkan sinyal yang diterima oleh bioelektroda hingga berkisar ratusan milivolt. Hal itu dilakukan untuk menaikkan tegangan potensial aksi agar dapat dimonitor dan ditangkap oleh osiloskop digital.

Penggunaan low-pass filter 10 Hz ditujukan untuk menyaring noise dan memblok frekuensi 50 Hz. Sinyal aktifitas listrik yang telah difilter akan dinaikkan tegangan referensinya oleh direct adder. Direct adder akan menaikkan tegangan referensi sebesar $\frac{1}{2} V_{cc}$. Hal itu dimaksudkan agar sinyal dapat sepenuhnya terbaca oleh ADC.

Sinyal yang dihasilkan oleh direct adder akan dikuatkan oleh inverting amplifier sebesar 100 kali. Secara keseluruhan penguatan yang dilakukan sebesar 1000 kali. Adapun untuk proses pengambilan data dilakukan pada masing-masing blok dan juga penggabungannya.

3.2 PERALATAN

Adapun peralatan yang dipakai pada penelitian ini antara lain :

1. DC Power Supply (GW INSTEK GPS-18300)
Merupakan regulator power supply dengan batasan maksimum tegangan DC yang dapat dikeluarkan sebesar + 15 Volt dan - 15 Volt. Alat ini disertai dengan pengaturan arus yang keluar dengan nilai maksimum 0.8 Ampere. Pada penelitian ini digunakan nilai tegangan sebesar + 9 Volt dan - 9 Volt dengan nilai arus tak lebih dari 0.3 mA.
2. Fungsi generator (FEEDBACK G 601)
Fungsi generator ini dilengkapi dengan pengaturan frekuensi pada range 0.001 Hz hingga 10 MHz. Alat ini pun dilengkapi dengan pengaturan tegangan keluaran dengan range mulai dari 1 mV hingga 5 V. Pada penelitian ini digunakan tegangan keluaran dari fungsi generator sebesar 2 V p-p dan variasi frekuensi untuk menguji filter aktif yang digunakan.
3. Digital Storage Oscilloscope (LEADER 31000)
Osiloskop ini digunakan karena fasilitas storage yang dimilikinya dan juga kemampuan osiloskop ini untuk menangkap sinyal rendah.
4. Digital Multimeter (CONSTANT Instrument)
Multimeter digital ini digunakan untuk proses pengecekan rangkaian yang digunakan dan untuk proses uji linieritas instrumentasi amplifier yang digunakan.
5. Tiny InterNet Interface (DALLAS Semiconductor)
TINI Mikrokontroller digunakan sebagai sarana penyimpan data serta sebagai tempat berjalannya semua rangkaian proses software.
6. Cross-over Ethernet cable
Kabel ini digunakan untuk koneksi Ethernet yang dilakukan antara TINI dengan PC.

7. Electroda Ag/AgCl

Digunakan untuk mendeteksi transduser yang dapat mendeteksi sinyal potensial aksi. Elektroda ini hanya bisa dipakai untuk sekali pengambilan data. Adapun untuk pengambilan data diperlukan minimal 3 buah elektroda untuk sekali pengambilan data.

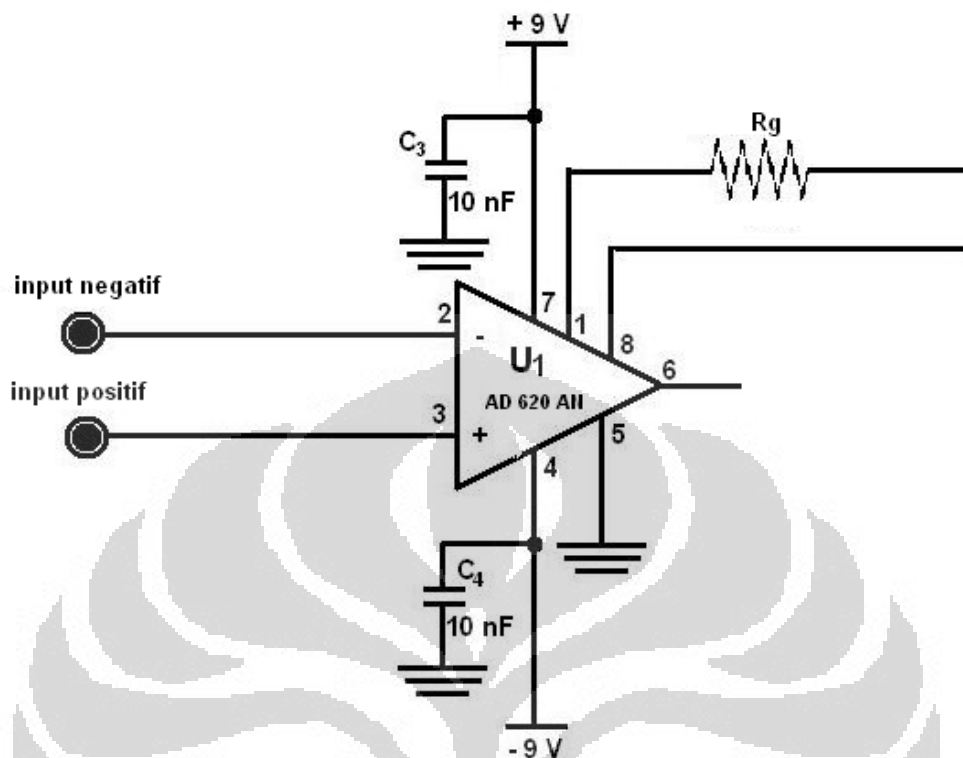
3.3 SISTEM HARDWARE

Untuk merancang sistem hardware yang akan digunakan untuk mendeteksi sinyal yang dikeluarkan oleh jantung diperlukan beberapa tahapan. Sinyal aktifitas listrik yang dikeluarkan oleh jantung sangatlah kecil dan untuk tujuan itu diperlukan sebuah penguatan.

Penelitian ini menggunakan sebuah instrumentasi amplifier sebagai penguat pertama. Untuk tahapan selanjutnya diperlukan sebuah filter aktif untuk menyaring dan mereduksi sinyal-sinyal yang tak diinginkan. Penggunaan direct adder dan non-inverting amplifier pada gambar 3.1 ditujukan untuk mengubah tegangan referensi pada hasil keluaran sehingga gelombang keluaran akan sepenuhnya bernilai positif. Non-inverting amplifier digunakan untuk menguatkan sinyal sampai skala satuan volt.

3.3.1 INSTRUMENTASI AMPLIFIER

Pada penelitian ini digunakan sebuah IC instrumentasi amplifier (IA) AD620AN. Pada IA ini terdapat sebuah hambatan yang difungsikan sebagai gain resistance (R_g). Hambatan R_g akan mengatur berapa nilai penguatan yang akan diberikan oleh IA. AD620AN merupakan sebuah IA yang memiliki impedansi input yang cukup tinggi sehingga dapat menyangga hasil keluaran penguatannya. Kedua input AD620AN akan terpasang langsung dengan bioelektroda yang ditempel pada kulit dengan sebuah kabel koaxial dengan metode common ground.



Gambar 3.2 Skematik perancangan instrumentasi amplifier

Pada gambar di atas, nilai R_g merupakan nilai yang dapat diubah untuk menentukan nilai penguatan yang diinginkan berdasarkan persamaan (2.2). Dengan mengatur penguatan yang diberikan IA AD620AN sebesar 10 kali akan didapatkan nilai R_g sebesar 50.5 k Ω . Untuk proses pengujian kelinieritas AD620AN dilakukan uji linieritas. Adapun data uji linieritas dibahas pada Bab 4.

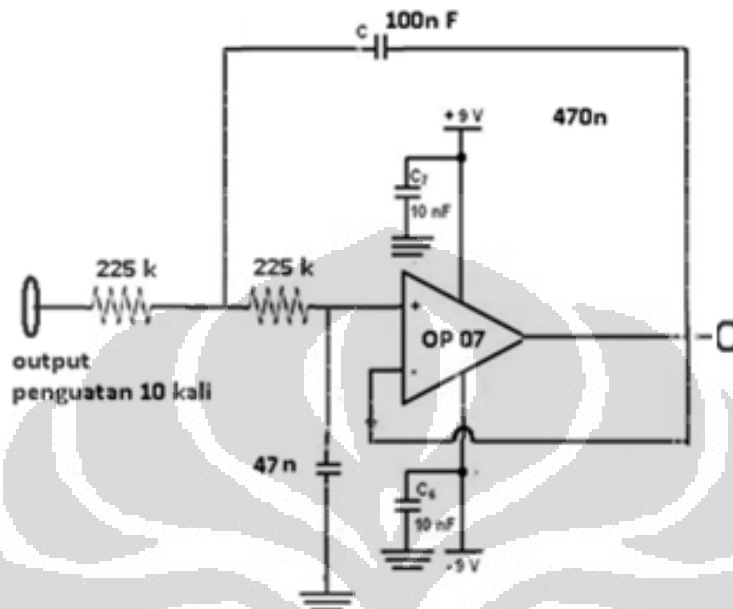
Pada penelitian ini untuk mendapatkan akurasi pada penguatan diberikan nilai R_g sebesar 50,5 k Ω dengan memakai multitone 100 k Ω .

3.3.2 PERANCANGAN FILTER AKTIF

Sinyal Elektrokardiogram merupakan sebuah sinyal lemah yang memiliki range frekuensi mulai dari 0.05 Hz hingga 100 Hz. Pada penelitian ini digunakan peralatan yang sebagian besar merupakan alat yang dapat memberikan sinyal yang tak diinginkan pada proses pengambilan data.

Dengan mengasumsikan bahwa sinyal aktiitas listrik yang terjadi pada jantung lebih dominan bekerja pada daerah range 1 Hz sampai dengan 2 Hz

(sekitar 80 detik per menit) diberikan sebuah low-pass filter 10 Hz untuk mereduksi sinyal PLN yang diberikan oleh peralatan yang digunakan.



Gambar 3.3 Skematik perancangan low-pass filter 10 Hz

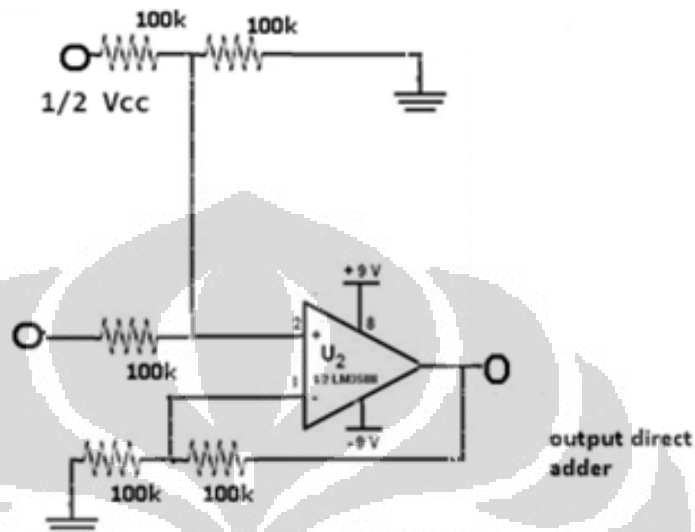
Pada penelitian ini digunakan IC OP 07. IC ini memiliki nilai offset yang rendah. Adapun konfigurasi OP 07 tidak begitu berbeda dengan LM741, tetapi OP07 memiliki kemampuan untuk mereduksi noise.

Dengan menggunakan persamaan $f = 1/(2 \pi RC)$ dan dengan mengasumsi nilai f merupakan nilai frekuensi cutoff yang diinginkan serta nilai $R_1 = R_2 = R = 225 \text{ k}\Omega$ didapatkan nilai $C_1 = C_2 = C$ sebesar 100 nF.

3.3.3 PERANCANGAN DIRECT ADDER DAN PENGUATAN 100 KALI

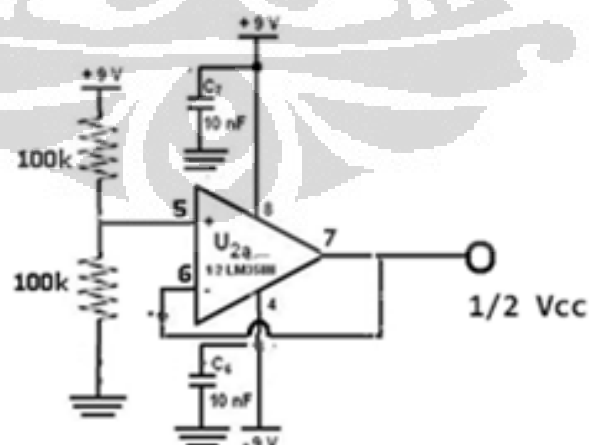
Gelombang elektrokardiogram yang terdeteksi merupakan sebuah gelombang yang memiliki nilai positif serta negatif. Ketika diberikan sebuah penguatan oleh instrumentasi amplifier sebesar 10 kali akan didapatkan sebuah output dengan besar 10 kali nilai input.

Direct adder merupakan sebuah rangkaian penjumlah. Pada penelitian ini digunakan direct adder untuk menaikkan tegangan referensi yang dipakai sebesar $\frac{1}{2} V_{cc}$. Hal itu ditujukan untuk menghasilkan sinyal keluaran positif penuh.



Gambar 3.4 Skematik perancangan direct adder

Dengan mengasumsikan nilai $R_f = R_1 = R_f' = R_1'$ sebesar $100\text{ k}\Omega$ maka akan didapatkan V_{out} sebesar penjumlahan kedua inputnya. Pada penelitian ini kedua input tersebut adalah output yang dihasilkan low-pass filter pada tahap sebelumnya dan $\frac{1}{2} V_{cc}$ yang diberikan.

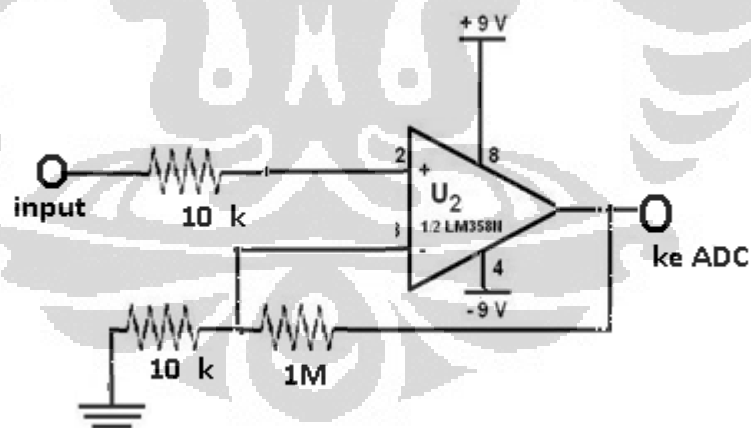


Gambar 3.5 Skematik perancangan pembagi tegangan

Untuk memberikan nilai $\frac{1}{2} V_{cc}$ pada input kedua digunakan sebuah pembagi tegangan dengan nilai hambatan yang sama ($R_1=R_2$) yang terhubung pada $+V_s$ dan ground. Nilai output pada pembagi tegangan tersebut akan menjadi $\frac{1}{2} V_{cc}$ dengan persamaan $V_{out} = (R_1/R_1+R_2)V_{in}$. Pada penelitian ini, V_{in} dihubungkan dengan $+V_s$ bernilai $+9\text{ V}$ dan $R_1=R_2$ bernilai $100\text{ k}\Omega$ sehingga menghasilkan keluaran sebesar 4.5 V .

Untuk mendapatkan hasil yang stabil pada pembagi tegangan digunakan sebuah rangkaian voltage follower dengan penguatan 1 kali. Voltage follower akan memberikan impedansi yang besar agar V_{out} yang dihasilkan direct adder tidak menyimpang terlalu jauh. Adapun penggunaan kapasitor 10 nF pada setiap tegangan sumber diberikan ditujukan sebagai kapasitor bypass.

Penguatan yang diberikan oleh IA akan memberikan keluaran yang masih sangat kecil. Untuk itu diperlukan sebuah devais untuk memberikan penguatan yang cukup besar agar output yang terdeteksi berorde satuan volt. Untuk tujuan itu digunakan sebuah rangkaian non-inverting amplifier dengan penguatan 100 kali.



Gambar 3.6 Skematik perancangan non-inverting amplifier

Dengan menggunakan persamaan $A = (R_f/R_1) + 1$, serta dengan mengasumsikan nilai $R_f = 1\text{ M}\Omega$ didapatkan nilai R_1 sebesar $10\text{ k}\Omega$. Untuk menjaga op-amp yang dipakai digunakan sebuah hambatan sebesar $10\text{ k}\Omega$ pada kaki non-inverting.

3.4 PERANCANGAN ADC

Pada penelitian ini menggunakan AVR ATmega 342p sebagai ADC. ATmega 342 p memiliki resolusi 10-bit. Untuk mengeset ATmega diperlukan Bascom AVR untuk merancang aplikasi yang ingin dilakukan oleh ATmega. Untuk membuat ADC dapat dilakukan dengan perintah config ADC.

```

$regfile = "m324pdef.dat"
$crystal = 8000000
$hwstack = 32      ' default use 32 for the hardware stack
$swstack = 10     ' default use 10 for the SW stack
$framesize = 40   ' default use 40 for the frame space
Open "comd.3:19200,8,n,1" For Output As #1
Config Adc = Single , Prescaler = Auto , Reference = Avcc
Start Adc
Do
Print #1 , Getadc(0)
Waitms 10
Loop
End

```

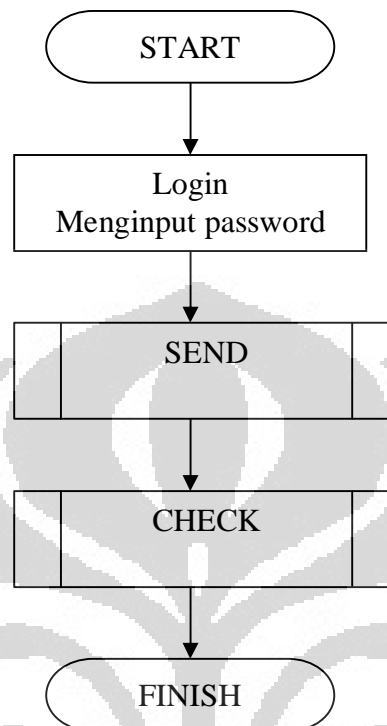
Pada penelitian ini ATmega diset untuk mengambil data setiap 10 ms. Untuk menguji dilakukan menggunakan input berasal dari potensiometer yang dipantau melalui Hyper Terminal secara serial.

3.5 ALUR SOFTWARE

Untuk mulai menggunakan TINI sebagai aplikasi diperlukan beberapa tahapan. TINI harus terlebih dahulu diisi kernel untuk dapat dipergunakan dan diisi aplikasi. Untuk mengisi kernel pada TINI telah dibahas pada Bab 2.

Tahapan selanjutnya diperlukan beberapa library yang diperluakn untuk membuat aplikasi menggunakan java, terutama library untuk beberapa class yang diperlukan untuk TINI, yaitu tini.jar. untuk menggunakan excel diperlukan library jxl.jar dan untuk menggunakan grafik pada java diperlukan library JfreeChart.jar dan JCommon.jar.

Perancangan software sepenuhnya terjadi pada TNI mikrokontroller. Adapun alur kerja software secara umum adalah sebagai berikut :



Gambar 3.7 Alur software

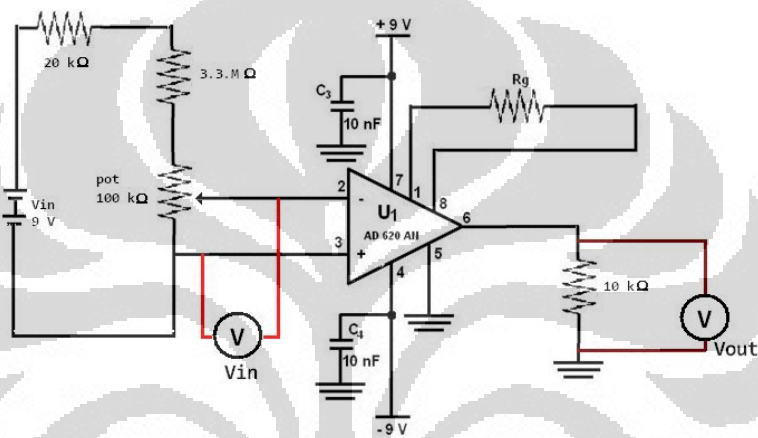
Input password dijadikan sebagai pintu gerbang proses pengambilan, penyimpanan dan aliran data. Selanjutnya setelah proses input password berhasil, TNI langsung mengirim perintah untuk pengambilan data melalui inputStream dan disimpan dalam memory yang dimiliki TNI. Pada subrutin CHECK, dilakukan inisialisasi IP yang digunakan dan dimulainya proses transfer data melalui ethernet ke PC lalu disimpan dalam file berbentuk excel. Hasil yang diterima akan disampaikan dalam bentuk grafik waktu versus tegangan.

BAB 4 HASIL DAN ANALISA

4.1 HASIL PENELITIAN

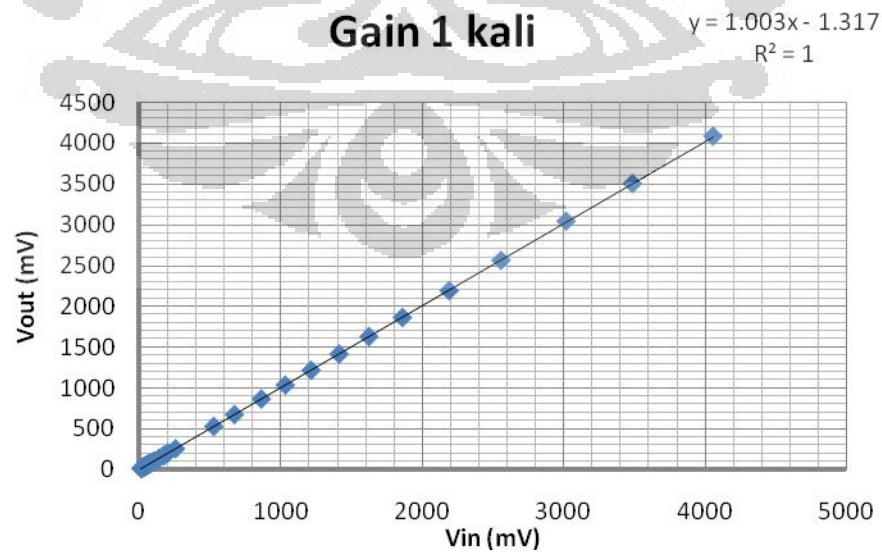
4.1.1 UJI PENGUATAN

Untuk menguji penguatan dari AD 620 AN digunakan sebuah rangkaian uji tes penguatan yang tertera dalam datasheet. Rangkaian lengkap pengujian sebagaimana terlihat pada gambar dibawah ini.

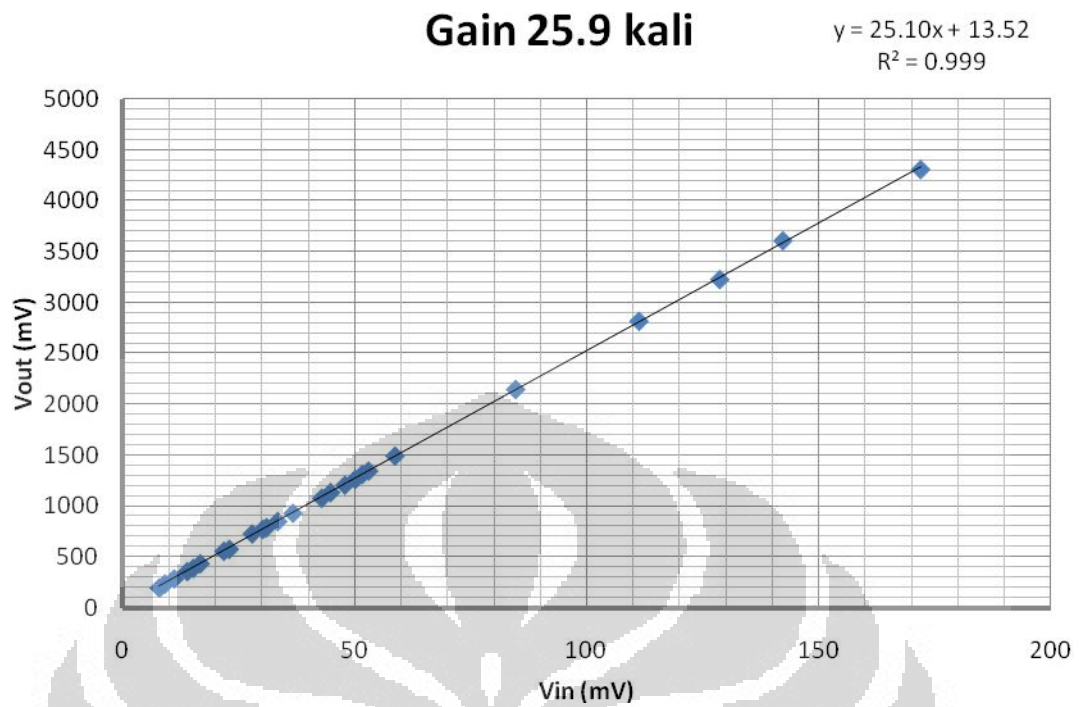


Gambar 4.1 Rangkaian pengujian penguatan (sudah diolah)

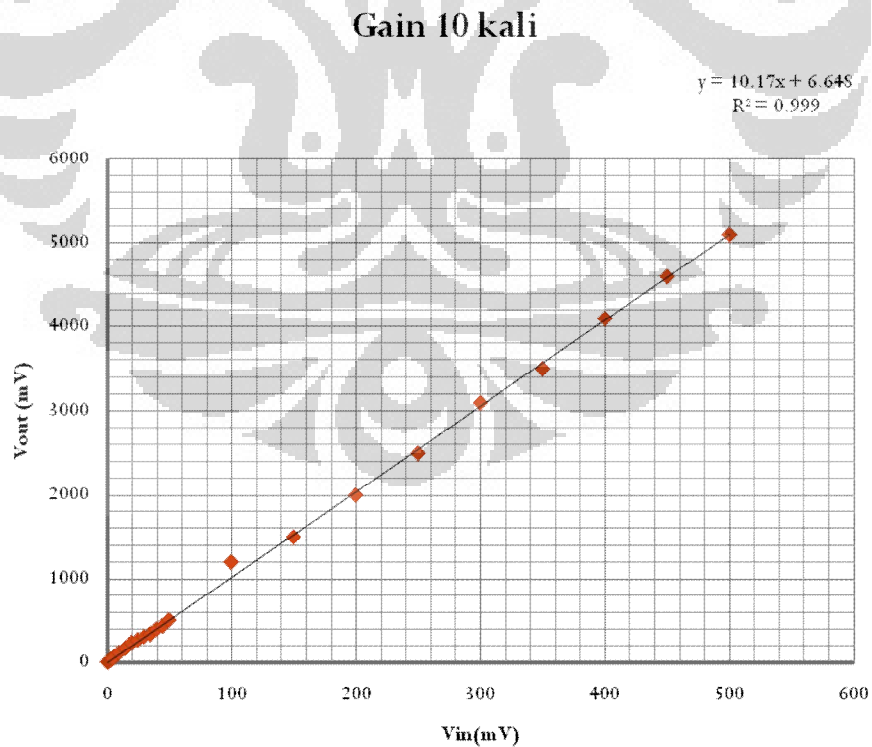
Adapun data yang diperoleh tertera pada grafik di bawah ini.



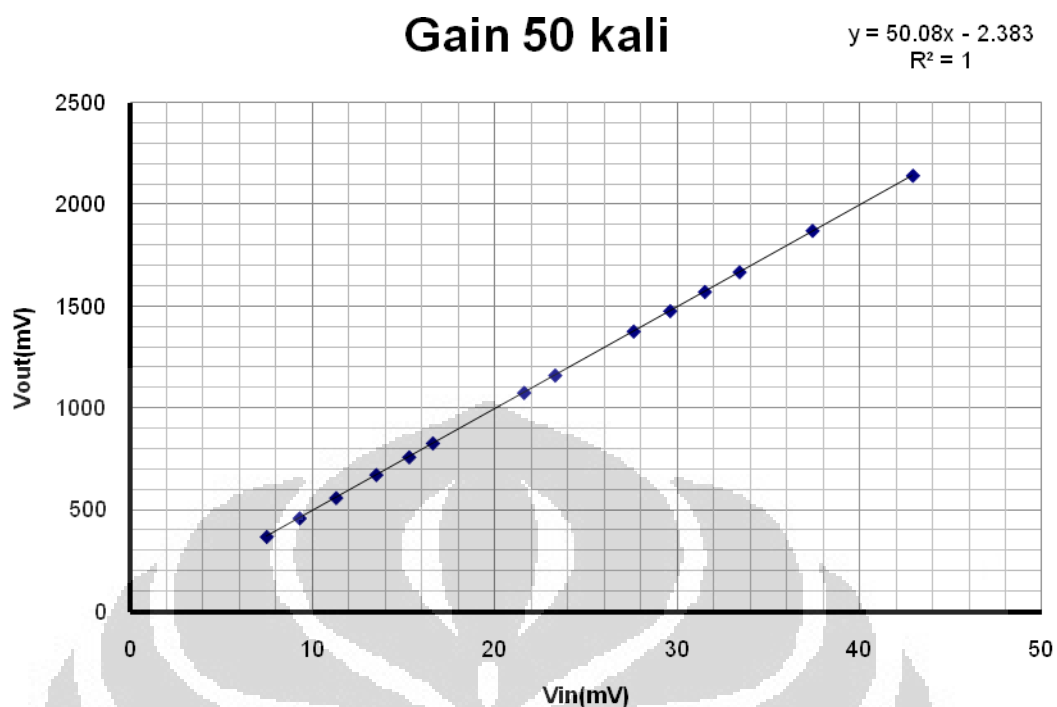
Gambar 4.2 Grafik hasil uji penguatan 1 kali (sudah diolah)



Gambar 4.3 Grafik hasil uji penguatan 25 kali (sudah diolah)



Gambar 4.4 Grafik hasil uji penguatan 10 kali (sudah diolah)



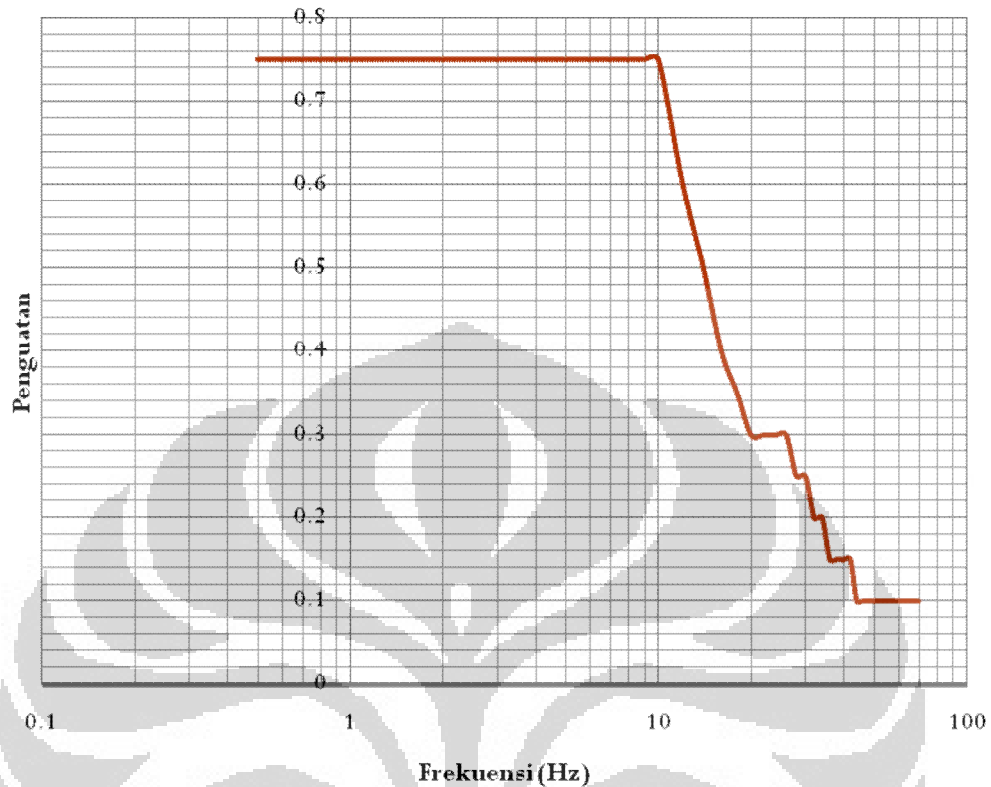
Gambar 4.5 Grafik hasil uji penguatan 50 kali (sudah diolah)

Data di atas menggunakan nilai R_g sebesar $2 \text{ k}\Omega$ dengan toleransi 1% untuk penguatan 25,9 kali dan $1 \text{ k}\Omega$ dengan toleransi 1% untuk penguatan 50 kali. Adapun untuk penguatan 1 kali tidak menggunakan R_g . Penelitian ini menggunakan penguatan sebesar 10 kali untuk IC AD620AN.

4.1.2 UJI LOW-PASS FILTER

Penelitian ini menggunakan low-pass filter 10 Hz. Rangkaian dan perancangan low-pass sudah dibahas pada Bab 2. Gambar 4.6 merupakan grafik hasil uji low-pass filter 10 Hz. Adapun nilai frekuensi cut-off pada uji low-pass filter yang digunakan adalah 10.6 Hz. Data yang diambil mulai dari frekuensi rendah bernilai 1 Hz hingga frekuensi 100 Hz untuk melihat nilai penguatan yang dihasilkan pada frekuensi 50 Hz.

Uji Low-Pass Filter 10 Hz



Gambar 4.6 Grafik hasil uji low-pass filter 10 Hz (sudah diolah)

4.1.3 DATA ELEKTROCARDIOGRAM

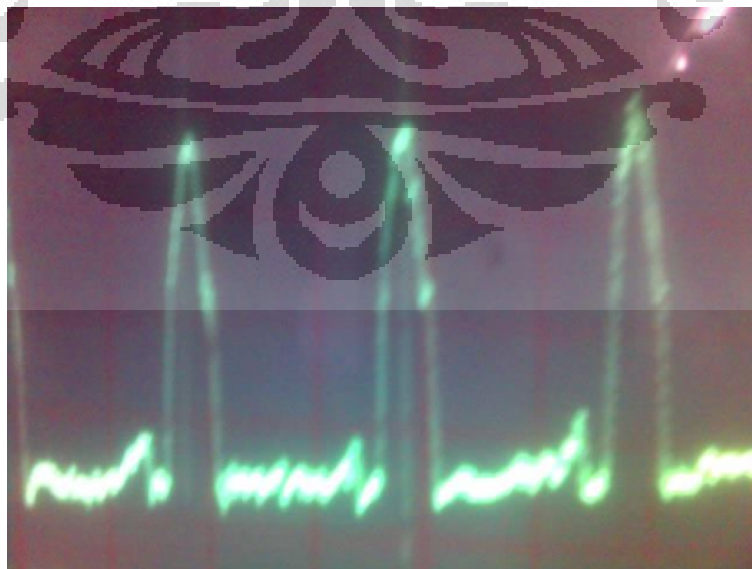
Menggunakan sebuah rangkaian elektrokardiogram yang telah dijelaskan pada Bab 2. Rangkaian tersebut menggunakan 2 buah Op-Amp untuk menguatkan dan membuat *low-pass filter*. Op-amp AD620AN diset untuk dapat menguatkan input yang dihasilkan sebanyak 10 kali. Untuk melihat hasil penguatan yang dihasilkan, Penulis menggunakan sebuah osiloskop storage digital.

Gambar 4.7 hingga 4.8 merupakan data yang diambil dengan menggunakan osiloskop storage mulai dari data sebelum difilter, setelah difilter hingga setelah dikuatkan 100 kali oleh non-inverting amplifier.



Gambar 4.7. Data analog tanpa filter (sudah diolah)

Gelombang di atas didapatkan dengan proses pengukuran pada dada (V1 dan V2). Adapun kalibrasi osiloskop storage yang dipakai adalah 0.1 V/div dan 5 ms/div. Dengan beranggapan bahwa penguatan yang dipakai adalah 10 kali dan potensial aksi berkisar 50 mV, maka data yang dihasilkan memiliki nilai V_{p-p} sebesar 0.7 V dan frekuensi 50 Hz. Data pada gambar 4.7 merupakan data yang dihasilkan oleh AD620AN tanpa filter dengan tegangan sumber sebesar +5 V dan -5 V.



Gambar 4.8. Data analog dengan low-pass filter (sudah diolah)

Gambar 4.8 merupakan gelombang yang diambil dengan menggunakan osiloskop storage dengan kalibrasi 5 mV/div. Dengan melihat data yang maka data yang ditunjukkan oleh osiloskop digital memiliki V p-p sebesar 20 mv dengan frekuensi sebesar 100 hz.

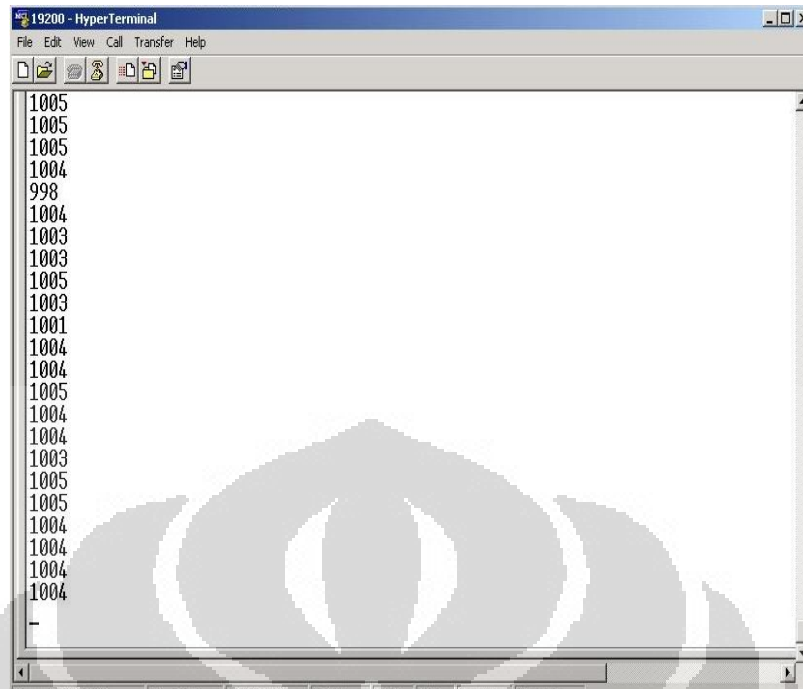


Gambar 4.9. Data analog dengan penguatan 1000 kali (sudah diolah)

Gambar 4.9 merupakan data yang diperoleh dengan menambahkan rangkaian penguatan non-inverting sebesar 100 kali sehingga penguatan total yang diberikan sebesar 1000 kali. Dengan melihat gambar 4.9, tegangan output yang diperoleh bernilai 2 V p-p dengan frekuensi sebesar 50 Hz.

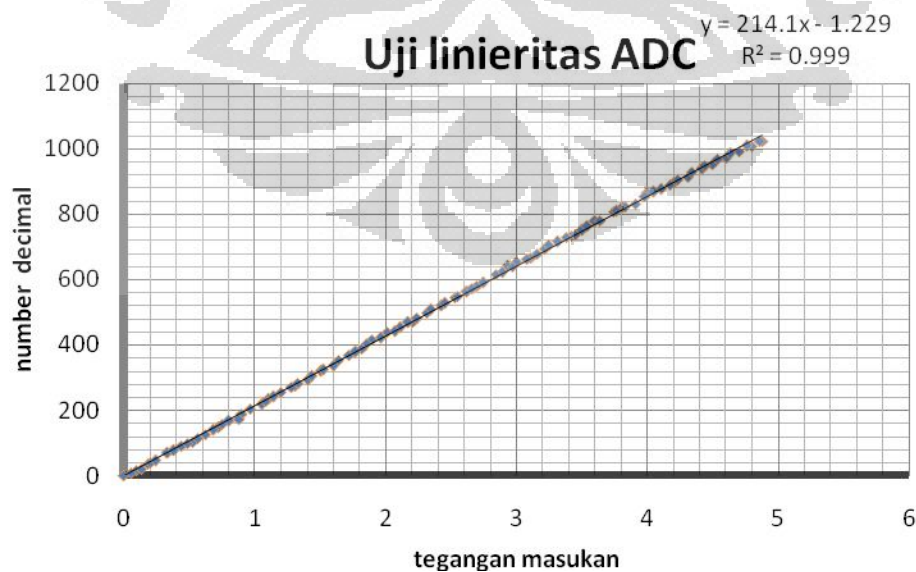
4.1.4 UJI ADC

Pada proses pengujian keluaran dari ADC ATmega342p digunakan sebuah potensiometer 50 kohm untuk memberikan input acak kepada ADC. Keluaran coba dihubungkan ke RS232. Hasil dari komunikasi ditampilkan dalam Hyper Terminal didapatkan hasil pada gambar 4.10 dengan *baud rate* 19200 dengan satu buah pin keluaran. Gambar 4.10 diperoleh dengan melihat bit keluaran untuk setiap level tegangan input yang diberikan.



Gambar 4.10 Data hasil komunikasi ATmega 342 P dengan RS232 (sudah diolah)

Untuk menguji linieritas dari ADC diperlukan sebuah sumber tegangan. Sumber tegangan didapatkan dengan menggunakan tegangan supply dari ADC yang diatur dengan potensiometer. Adapun data yang didapatkan dari uji linieritas tertera dalam lampiran. Grafik yang dihasilkan adalah sebagai berikut.

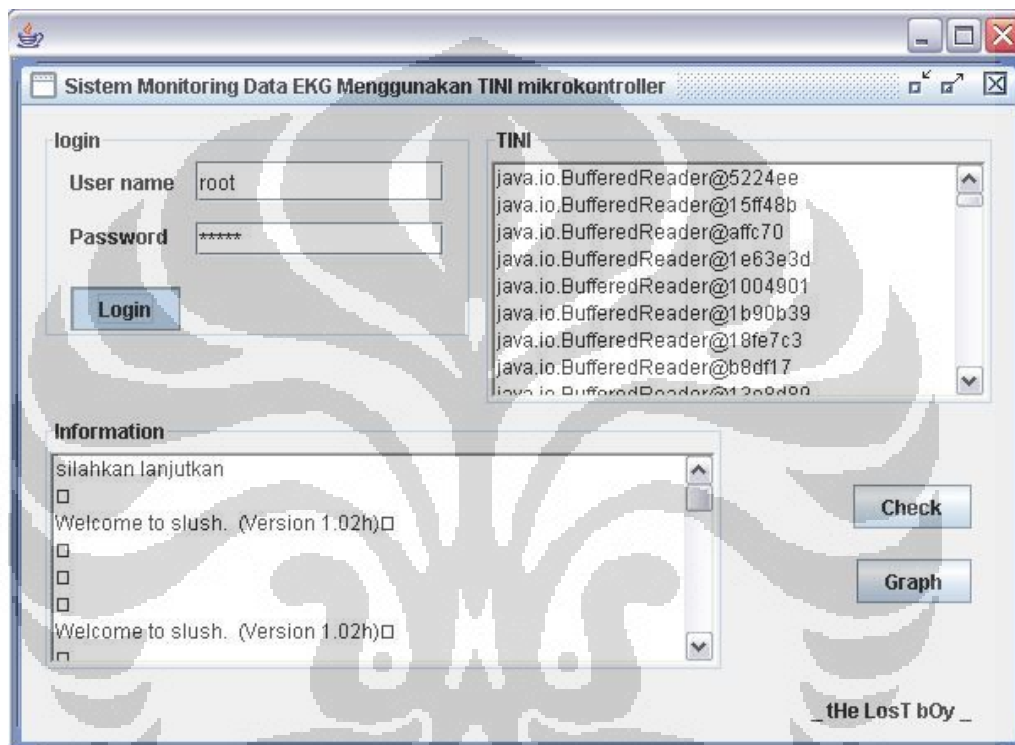


Gambar 4.11 Grafik hasil uji linieritas ADC (sudah diolah)

4.1.5 LAYOUT PC

Adapun untuk display layout digunakan NetBeans dengan memanfaatkan aplikasi JFrame yang disediakan. Untuk dapat mengaktifkannya dari main project, kita dapat mengklik add new JFrame yang terdapat pada menu pop-up di *project* kita.

Adapun rancangan display yang akan dikerjakan adalah sebagai berikut:



Gambar 4.12. Layout hasil

Pada gambar 4.12, terdapat 3 buah *button* yang difungsikan untuk mengaktifkan *user*, mengecek apakah data sudah masuk ke TINI dan untuk menampilkan grafik. Sebuah JTextArea pada panel informasi akan berisi beberapa informasi tentang proses pengambilan data dan JTextArea pada panel TINI akan berisi keterangan bahwa TINI mengambil data serta proses *buffering* data di TINI. Tombol Check akan memberikan perintah pada TINI untuk membuat sebuah workbook excel dengan nama file *ecg.xls* yang akan diisi oleh data hasil buffer yang dilakukan sebelumnya.

4.2 ANALISA

Berdasarkan alur yang dilakukan, analisa yang akan dijabarkan di sini akan terbagi dalam beberapa bagian.

4.2.1 UJI LINIERITAS

Pada penelitian ini dilakukan uji linieritas dengan tiga buah besar penguatan yang berbeda-beda. Hal itu dilakukan untuk membuktikan bahwa AD 620 AN menghasilkan penguatan yang linier untuk tiap besar penguatan. Adapun beberapa penguatan yang digunakan adalah 1, 25.7 dan 50. Pada penelitian ini digunakan penguatan 10kali.

Hasil yang diperoleh untuk ketiga penguatan tersebut membuktikan AD620AN yang digunakan masih cukup baik ditunjukkan dengan nilai R^2 yang mendekati 1. Dari ketiga grafik dihasilkan kesalahan literatur untuk penguatan 1 kali sebesar 0.002 %, untuk penguatan 25.9 kali sebesar 0.03 % dan untuk penguatan 50 kali sebesar 0.01 %.

Instrumentasi amplifier digunakan karena beberapa keunggulan yang dimiliki, antara lain pengguna dapat mengatur berapa penguatan yang diinginkan hanya dengan mengganti nilai R_g dengan menggunakan persamaan (2.4). Kedua, hasil penguatan *band-width* akan meningkat jika digunakan penguatan yang berbeda kemudian akan memaksimalkan respon dari frekuensinya. Ketiga, tegangan derau pada nput akan berkurang sebesar $9 \text{ nV}/\sqrt{\text{Hz}}$ dikarenakan adanya hambatan basis dan arus kolektor pada transistor di kedua sumbernya.

Pada pengambilan data uji linieritas besar nilai $A \cdot V_{in}$ tidak akan terus menerus meningkat seiring ditambahkannya V_{in} . Nilai V_{out} akan terhenti pada nilai 2 Volt di bawah V_s dan bernilai stabil. Oleh karena itu perlu sekali mengatur nilai V_s yang cukup besar, Penulis menggunakan + 9 V dan - 9V.

Pada proses uji linieritas AD620AN Kedua input diberikan besar tegangan bernilai positif dan negatif serta bernilai kecil. Hal itu dilakukan untuk mensimulasikan potensial aksi yang dikeluarkan jantung. Besar hasil differential input yang terjadi merupakan penjumlahan V_+ dengan V_- .

$$V_{out} = A(V_2 - V_1)$$

Dikarenakan V_1 masuk ke dalam masuk bernilai negatif dan berperan sebagai input inverting maka V^{in} merupakan penjumlahan dari kedua inputnya.

4.2.2 ANALISA DATA ANALOG

Nilai gelombang potensial aksi hanya berkisar $50 \text{ mV}^{[9]}$ /elektroda. Adapun jumlah total penguatan yang muncul pada kedua input AD620AN berkisar antara 500 mV sampai 700 mV dengan mengasumsikan nilai input yang diberikan oleh sinyal aktifitas jantung sekitar 50 mv sampai 70 mV. Penguatan itu diberikan oleh op-amp AD620AN sebanyak 10 kali. Berdasarkan datasheet yang digunakan untuk AD620AN di dapatkan besar R_g untuk penguatan 10 sebesar $50.5 \text{ k}\Omega$. Pada penelitian ini digunakan multitone sebesar $100 \text{ k}\Omega$ untuk memberikan nilai R_g sebesar $50.5 \text{ k}\Omega$ Penguatan tersebut akan ditambahkan dengan penguatan yang diberikan oleh non-inverting pada $\frac{1}{2}$ LM358N sebesar 100 kali.

Untuk menghilangkan *noise*, digunakan beberapa metode. Selain digunakan sistem penggerak kaki kanan seperti yang telah dijabarkan pada Bab 2, digunakan sebuah rangkaian low-pass filter dengan frekuensi cut-off sebesar 10.6 Hz. Pemakaian low-pass filter ini akan mereduksi semua frekuensi di atas 10.6 Hz. Hal ini yang akan merugikan pemakaian low-pass filter 10 Hz. Pada Bab 2 telah dijelaskan bahwa frekuensi yang dikeluarkan oleh jantung berkisar antara 0.05 Hz sampai 100 Hz. Dengan demikian pemakai low-pass filter efektif untuk menghilangkan frekuensi 50 Hz tapi tidak begitu efisien dengan hilangnya semua output yang memiliki frekuensi dari 60 Hz hingga 100 Hz.

Pada proses pengambilan data analog penguatan tidak dilakukan langsung 1000 kali pada AD620AN. Hal itu dimaksudkan untuk mencegah adanya penguatan noise yang dikeluarkan oleh hardware. AD620AN memberikan penguatan sebesar 10 kali hanya untuk mengangkat sinyal aktifitas jantung yang kecil agar mampu terdeteksi oleh low-pass filter. Low-pass filter akan mereduksi noise yang muncul serta mereduksi besar tegangan yang didapatkan. Ketika semua sinyal yang tak diinginkan berhasil dibuang/diblok, maka penguatan dilakukan sebesar mungkin hingga keluaran dapat terdeteksi oleh ADC.

Selain dengan menggunakan rangkaian *low-pass filter*, digunakan sebuah metode plane ground, penggunaan komponen bertoleransi rendah, penggunaan kapasitor *polyester*, yang dapat mereduksi adanya munculnya *noise* dari luar.

4.2.2 PEMBAHASAN SOFTWARE

Saat data sudah digital, TINI siap memproses data tersebut. JVM (*Java Virtual Machine*) yang di miliki akan mengeksekusi software yang sudah di simpan dalam TINI.

Pada penelitian ini, user yang dapat dipakai masih terbatas yaitu hanya memperkenankan satu buah user. Hal itu dikarenakan ketika dipergunakan multi-user diperlukan sebuah database. Untuk mengaktifkan database diharuskan mengaktifkan driver untuk MySQL. Penelitian ini tidak mempergunakan MySQL dikarenakan variabel data yang dipakai sangat sedikit dan merupakan sebuah pemborosan jika menggunakan MySQL.

Data digital yang dihasilkan oleh AVR diolah oleh TINI akan di tampung ke dalam sebuah memori. Memori yang terdapat di TINI sebesar 512 kbyte². Data EKG yang berasal dari AVR di simpan dalam TINI dan menunggu perintah untuk mengalirkannya ke TINI untuk diproses selanjutnya.

Data akan disimpan dalam sebuah notepad menggunakan File class yang sudah terdapat dalam package netbeans⁸. Untuk proses selanjutnya yaitu mengintegrasikan data ke dalam bentuk excel diperlukan library jxl.jar dan menampilkan grafik diperlukan library JfreeChart.jar. Library ini berisi beberapa class yang digunakan untuk menampilkan beberapa data ke bentuk grafik.

AVR digunakan untuk memproses data analog yang didapatkan ke dalam bentuk digital. Data keluaran yang dihasilkan merupakan data digital 10-bit. Pada penelitian ini, AVR hanya difungsikan sebagai konverter bukan sebagai pengolah data. Pada proses uji linieritas data di AVR dihasilkan nilai R² sebesar 0.999 dengan nilai kesalahan sebesar ± 2 LSB.

BAB 5

KESIMPULAN DAN SARAN

5.1 KESIMPULAN

Pada penelitian ini dapat disimpulkan beberapa hal, yaitu :

1. Instrumentasi amplifier AD 620 cukup linier ditunjukkan dengan nilai R^2 lebih besar dari 0.988 untuk empat penguatan, yaitu 1 kali, 10 kali, 25.9 kali dan 50 kali. Pada penelitian ini digunakan penguatan sebesar 10 kali dengan kesalahan 1.7 %.
2. Nilai kesalahan ADC yang diberikan oleh AVR berkisar ± 2 LSB dengan linieritas yang cukup baik ($R^2 = 0.999$).
3. Dibutuhkan beberapa library package tambahan untuk memaksimalkan peran Java seperti Jxl (untuk membuat dan menyimpan data dalam bentuk Excel) dan JfreeChart (untuk menampilkan sebuah grafik dalam aplikasi menggunakan Java).
4. Pada proses pereduksian sinyal dengan frekuensi 50 Hz menggunakan notch filter 50 hz didapat sinyal yang berhasil mereduksi frekuensi 50 Hz, namun ketika dilakukan penggabungan dengan instrumentasi amplifier sinyal 50 Hz masih sering muncul.
5. Frekuensi sinyal EKG yang digunakan tidak dapat sepenuhnya menghilangkan frekuensi 50Hz. Walau sudah dibeikan notch filter 50 hz.
6. Instrumentasi Amplifier AD620AN cocok untuk menguatkan input yang sangat kecil ($\sim mV$)
7. Proses penyanpaian data melalui ethernet sudah dapat dilakukan namun data ang dihasilkan masih dalam bentuk string.

5.2 SARAN

Penelitian ini sepenuhnya ditujukan untuk mengeksplere kembali apa yang dapat dilakukan dengan menggunakan TINI Mikrokontroller. Pada mikrokontroller ini terdapat beberapa media I/O, mulai dari koneksi serial hingga 1-wire yang sedang populer.

Oleh karena itu adanya sebuah penelitian berkesinambungan untuk pengembangan mikrokontroller ini sangat diperlukan. Untuk mengeksplorasi mikrokontroller ini tidak mudah, diperlukan sebuah pemahaman akan sebuah kompuler yang belum begitu umum, yaitu Java. Namum, secara garis besar untuk sebuah pengembangan teknologi, keberadaan TINI sangatlah membantu untuk membuat sebuah aplikasi berbasis *web-design* atau mungkin untuk beberapa aplikasi *databases system*.

Kembali pada penelitian yang dilakukan. Pada penelitian ini, penulis hanya dapat menampilkan sebuah data dari aktivitas jantung. Mungkin untuk penelitian selanjutnya dapat pula diamati sebuah perbedaan dari hasil yang didapatkan ketika pasien melakukan kerja (dapat diambil contoh, akan adanya perbedaan hasil data dari pasien berjalan dengan ketika pasien berlari), atau mungkin dengan adanya analisa pengaruh umur pada hasil yang didapatkan. Selain hal-hal yang bersifat metode seperti di atas, mungkin peneliti selanjutnya dapat mengubah hasil penelitian ini dengan berbasis web.

DAFTAR PUSTAKA

1. Carr, J. J and Brown, J. M. *Introduction to Biomedical Equipment Technology 3e*. USA : Prentice Hall. 1998.
2. Loomis, D. *The TINITM Specification and Developer's Guide*. USA : Addison-Wesley. 2001
3. <http://www.dallassemiconductor.com/TINI/> (25 Februari 2009)
4. Proakis, J. G and Malokalis, G. D. *Digital Signal Processing : Principles, Algorithms, and Application 3e*. USA:Prantice Hall Inc. 1995.
5. Foster, D. B. *Twelve-Lead Electrocardiography: Theory and Interpretasi 2e*.UK : Springer-Verlag.2007
6. Faulkenberry, L. M. *An Introduction To Operation Amplifiers With Linear IC Applications 2e*. Taiwan : John Wiley & Sons Inc.1983
7. <http://www.alldatasheet.com/AD620AN/> (12 Januari 2009)
8. Deitel, H. M and Deitel, P. J. *JAVATM, How To Program 6e*. USA.: Pearson Education. Inc. 2005
9. John G, W. *Medical Instrumentation Application and Design*. Canada : John Wiley & Sons. 1998
10. <http://www.planetnetbeans.com> (24 April 2009).
11. <http://www.alldatasheet.com/ATMega342P/> (12 Mei 2009)

LAMPIRAN 1

Data Uji Linieritas penguatan untuk Gain 1

| input positif (mV) | input negatif (mV) | hasil penjumlahan diferensial input (mV) | Vout (mV) |
|-----------------------|-----------------------|--|-----------|
| 8.10 | 4.70 | 12.80 | 12.70 |
| 8.70 | 5.40 | 14.10 | 14.10 |
| 9.60 | 6.10 | 15.70 | 15.70 |
| 10.30 | 6.70 | 17.00 | 17.00 |
| 11.90 | 8.30 | 20.20 | 20.00 |
| 12.40 | 8.90 | 21.30 | 21.10 |
| 13.10 | 9.70 | 22.80 | 22.80 |
| 14.10 | 10.70 | 24.80 | 24.60 |
| 15.80 | 12.20 | 28.00 | 27.60 |
| 16.70 | 13.00 | 29.70 | 29.20 |
| 17.10 | 13.30 | 30.40 | 30.90 |
| 18.40 | 14.90 | 33.30 | 33.20 |
| 19.50 | 16.90 | 36.40 | 35.20 |
| 20.30 | 16.90 | 37.20 | 36.90 |
| 21.50 | 17.90 | 39.40 | 38.70 |
| 22.60 | 19.00 | 41.60 | 41.10 |
| 23.70 | 20.00 | 43.70 | 43.20 |
| 24.20 | 20.50 | 44.70 | 43.60 |
| 25.00 | 21.40 | 46.40 | 45.70 |
| 26.60 | 22.90 | 49.50 | 48.70 |
| 27.30 | 23.40 | 50.70 | 50.10 |
| 28.50 | 24.60 | 53.10 | 52.40 |
| 28.60 | 24.80 | 53.40 | 52.50 |
| 29.60 | 25.50 | 55.10 | 54.10 |
| 30.40 | 26.30 | 56.70 | 55.60 |
| 31.20 | 27.50 | 58.70 | 58.10 |
| 32.00 | 28.20 | 60.20 | 59.30 |
| 32.10 | 28.30 | 60.40 | 59.50 |
| 34.50 | 30.30 | 64.80 | 63.50 |
| 35.50 | 31.70 | 67.20 | 66.20 |
| 36.30 | 32.50 | 68.80 | 68.10 |
| 38.70 | 35.10 | 73.80 | 72.70 |
| 41.40 | 37.80 | 79.20 | 78.30 |
| 46.50 | 42.50 | 89.00 | 87.10 |
| 55.10 | 51.20 | 106.30 | 105.60 |
| 71.30 | 67.80 | 139.10 | 138.10 |
| 88.30 | 84.10 | 172.40 | 170.40 |

| | | | |
|---------|---------|---------|---------|
| 90.50 | 86.60 | 177.10 | 175.80 |
| 102.70 | 98.90 | 201.60 | 202.00 |
| 130.00 | 126.00 | 256.00 | 254.00 |
| 265.00 | 261.00 | 526.00 | 527.00 |
| 338.00 | 335.00 | 673.00 | 672.00 |
| 434.00 | 429.00 | 863.00 | 863.00 |
| 518.00 | 515.00 | 1033.00 | 1033.00 |
| 609.00 | 606.00 | 1215.00 | 1215.00 |
| 709.00 | 705.00 | 1414.00 | 1413.00 |
| 814.00 | 811.00 | 1625.00 | 1626.00 |
| 932.00 | 930.00 | 1862.00 | 1861.00 |
| 1098.00 | 1095.00 | 2193.00 | 2190.00 |
| 1280.00 | 1280.00 | 2560.00 | 2560.00 |
| 1510.00 | 1510.00 | 3020.00 | 3040.00 |
| 1750.00 | 1740.00 | 3490.00 | 3500.00 |
| 2030.00 | 2030.00 | 4060.00 | 4080.00 |

Data Uji Linieritas penguatan untuk Gain 25,7

$$R_g = 2 \text{ k}\Omega$$

| input positif (mV) | input negatif (mV) | hasil penjumlahan differential input (mV) | Vout (mV) |
|--------------------|--------------------|---|-----------|
| 5.7 | 2.2 | 7.9 | 200 |
| 6.3 | 2.8 | 9.1 | 239 |
| 7.3 | 3.8 | 11.1 | 290 |
| 8.7 | 5.2 | 13.9 | 361 |
| 9.4 | 5.8 | 15.2 | 393 |
| 10.1 | 6.6 | 16.7 | 438 |
| 12.7 | 9.1 | 21.8 | 561 |
| 13.3 | 9.7 | 23 | 580 |
| 15.8 | 12.1 | 27.9 | 731 |
| 16.9 | 13.3 | 30.2 | 776 |
| 17.3 | 13.7 | 31 | 793 |
| 18.5 | 14.9 | 33.4 | 852 |
| 20.2 | 16.5 | 36.7 | 934 |
| 23.4 | 19.5 | 42.9 | 1082 |
| 24.2 | 20.6 | 44.8 | 1138 |
| 25.8 | 22.1 | 47.9 | 1210 |
| 27 | 23.1 | 50.1 | 1266 |
| 27.6 | 24 | 51.6 | 1314 |
| 28.4 | 24.6 | 53 | 1349 |
| 31.3 | 27.4 | 58.7 | 1497 |
| 44.3 | 40.4 | 84.7 | 2150 |

| | | | |
|------|------|-------|------|
| 57.8 | 53.5 | 111.3 | 2820 |
| 66.5 | 62.2 | 128.7 | 3230 |
| 73.4 | 68.9 | 142.3 | 3610 |
| 88.4 | 83.6 | 172 | 4310 |

Data Uji Linieritas penguatan untuk Gain 50,9

$$R_g = 1 \text{ k}\Omega$$

| input positif (mV) | input negatif (mV) | hasil penjumlahan differential input (mV) | Vout (mV) |
|--------------------|--------------------|---|-----------|
| 5.3 | 2.2 | 7.5 | 372 |
| 6.5 | 2.8 | 9.3 | 463 |
| 7.5 | 3.8 | 11.3 | 563 |
| 8.3 | 5.2 | 13.5 | 676 |
| 9.5 | 5.8 | 15.3 | 763 |
| 10 | 6.6 | 16.6 | 831 |
| 12.5 | 9.1 | 21.6 | 1079 |
| 13.6 | 9.7 | 23.3 | 1164 |
| 15.5 | 12.1 | 27.6 | 1380 |
| 16.3 | 13.3 | 29.6 | 1480 |
| 17.8 | 13.7 | 31.5 | 1574 |
| 18.5 | 14.9 | 33.4 | 1671 |
| 20.9 | 16.5 | 37.4 | 1874 |
| 23.4 | 19.5 | 42.9 | 2144 |
| 24.2 | 20.6 | 44.8 | 2242 |
| 25.5 | 22.1 | 47.6 | 2381 |
| 27.3 | 23.1 | 50.4 | 2521 |
| 27.9 | 24 | 51.9 | 2593 |
| 28.2 | 24.6 | 52.8 | 2638 |
| 31.5 | 27.4 | 58.9 | 2944 |
| 32.6 | 30.6 | 63.2 | 3150 |

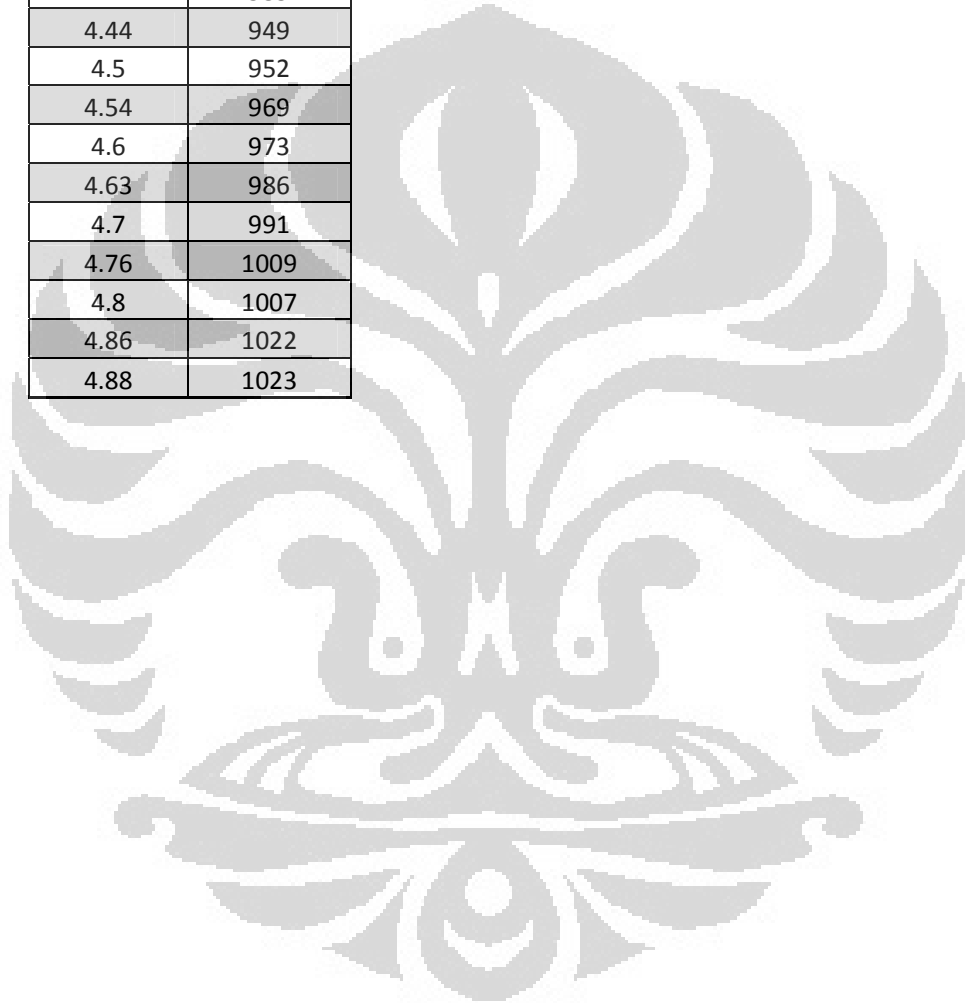
LAMPIRAN 2

Data linieritas ADC

| Vin (V) | Nilai Biner |
|---------|-------------|
| 0 | 0 |
| 0.05 | 8 |
| 0.09 | 15 |
| 0.13 | 21 |
| 0.18 | 36 |
| 0.2 | 39 |
| 0.24 | 46 |
| 0.33 | 69 |
| 0.38 | 77 |
| 0.44 | 90 |
| 0.48 | 98 |
| 0.52 | 103 |
| 0.56 | 115 |
| 0.62 | 125 |
| 0.68 | 139 |
| 0.72 | 148 |
| 0.75 | 158 |
| 0.8 | 167 |
| 0.88 | 172 |
| 0.9 | 188 |
| 0.96 | 204 |
| 1.05 | 218 |
| 1.08 | 228 |
| 1.1 | 234 |
| 1.14 | 242 |
| 1.2 | 254 |
| 1.28 | 272 |
| 1.31 | 276 |
| 1.33 | 282 |
| 1.4 | 293 |
| 1.43 | 304 |
| 1.5 | 320 |
| 1.52 | 324 |
| 1.6 | 335 |
| 1.64 | 354 |
| 1.72 | 368 |
| 1.77 | 379 |
| 1.82 | 389 |

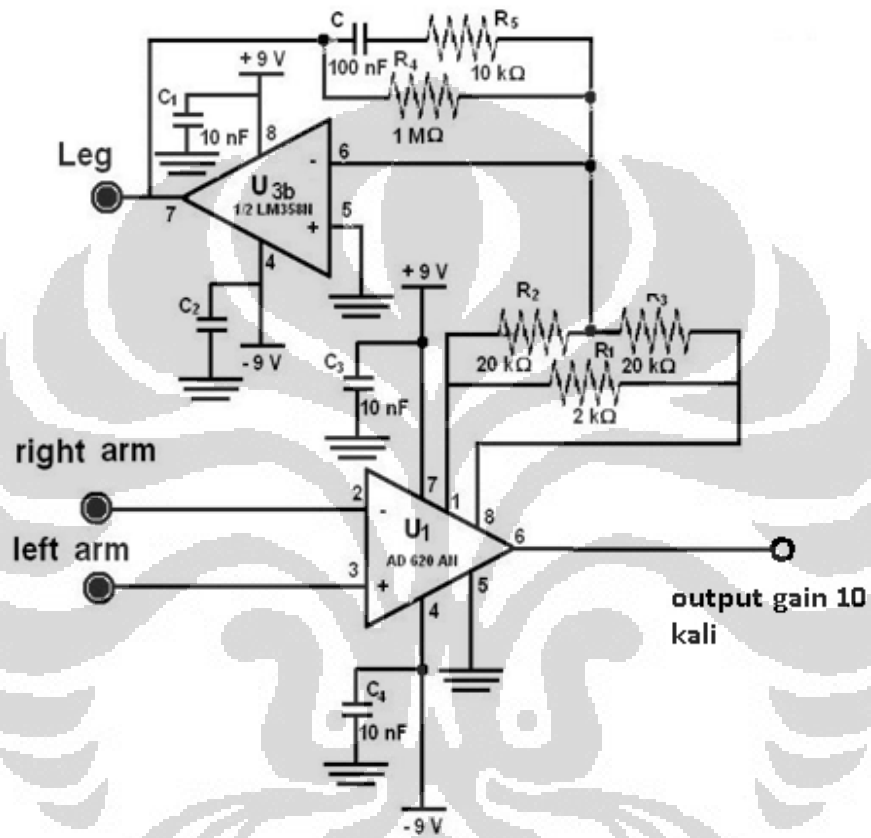
| | |
|------|-----|
| 1.85 | 403 |
| 1.89 | 416 |
| 1.96 | 422 |
| 2.01 | 439 |
| 2.07 | 441 |
| 2.11 | 453 |
| 2.17 | 468 |
| 2.21 | 471 |
| 2.24 | 482 |
| 2.31 | 499 |
| 2.34 | 508 |
| 2.42 | 520 |
| 2.45 | 528 |
| 2.53 | 545 |
| 2.55 | 550 |
| 2.62 | 562 |
| 2.66 | 572 |
| 2.7 | 581 |
| 2.74 | 590 |
| 2.84 | 616 |
| 2.89 | 621 |
| 2.93 | 641 |
| 2.97 | 640 |
| 3 | 654 |
| 3.08 | 662 |
| 3.11 | 667 |
| 3.16 | 679 |
| 3.21 | 692 |
| 3.24 | 703 |
| 3.31 | 715 |
| 3.38 | 729 |
| 3.42 | 733 |
| 3.45 | 738 |
| 3.5 | 751 |
| 3.54 | 762 |
| 3.6 | 778 |
| 3.64 | 779 |
| 3.74 | 807 |
| 3.76 | 810 |
| 3.81 | 821 |
| 3.83 | 823 |
| 3.91 | 831 |
| 3.98 | 855 |

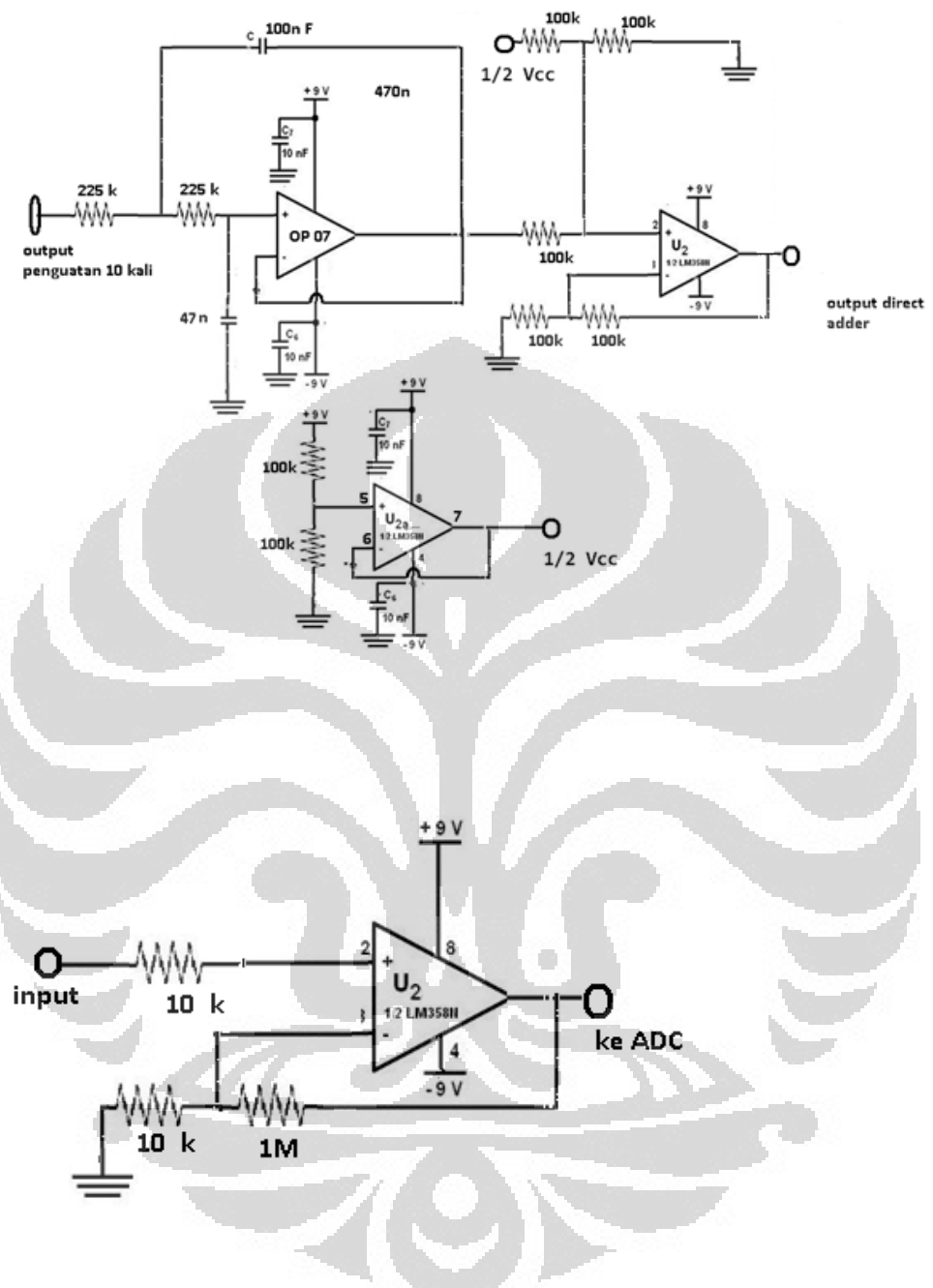
| | |
|------|------|
| 4 | 866 |
| 4.05 | 868 |
| 4.1 | 876 |
| 4.17 | 891 |
| 4.2 | 895 |
| 4.23 | 905 |
| 4.31 | 914 |
| 4.34 | 929 |
| 4.42 | 939 |
| 4.44 | 949 |
| 4.5 | 952 |
| 4.54 | 969 |
| 4.6 | 973 |
| 4.63 | 986 |
| 4.7 | 991 |
| 4.76 | 1009 |
| 4.8 | 1007 |
| 4.86 | 1022 |
| 4.88 | 1023 |



LAMPIRAN 3

Skematik Elektrokardiogram





LAMPIRAN 4

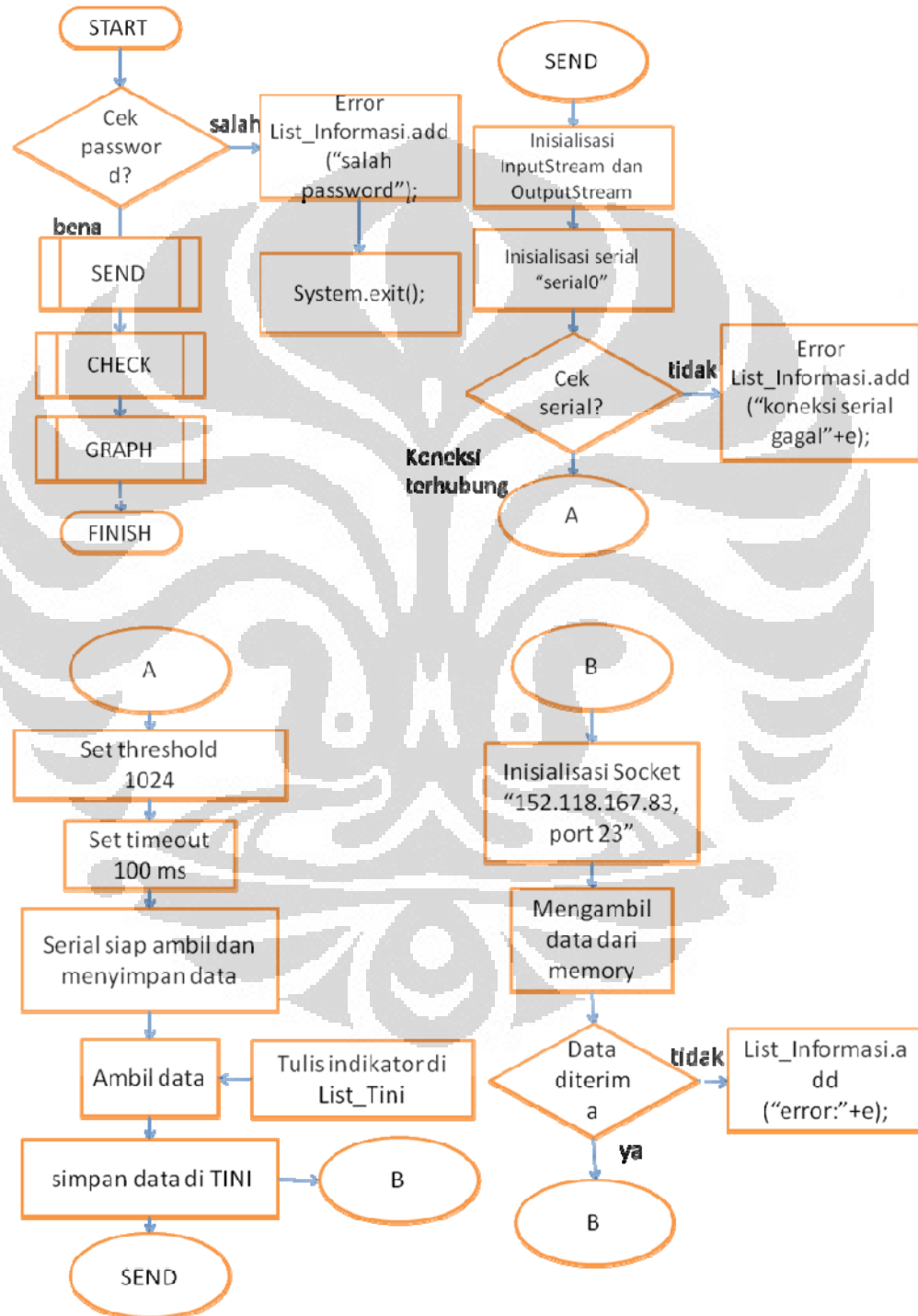
Data uji Low-Pass Filter 10 Hz

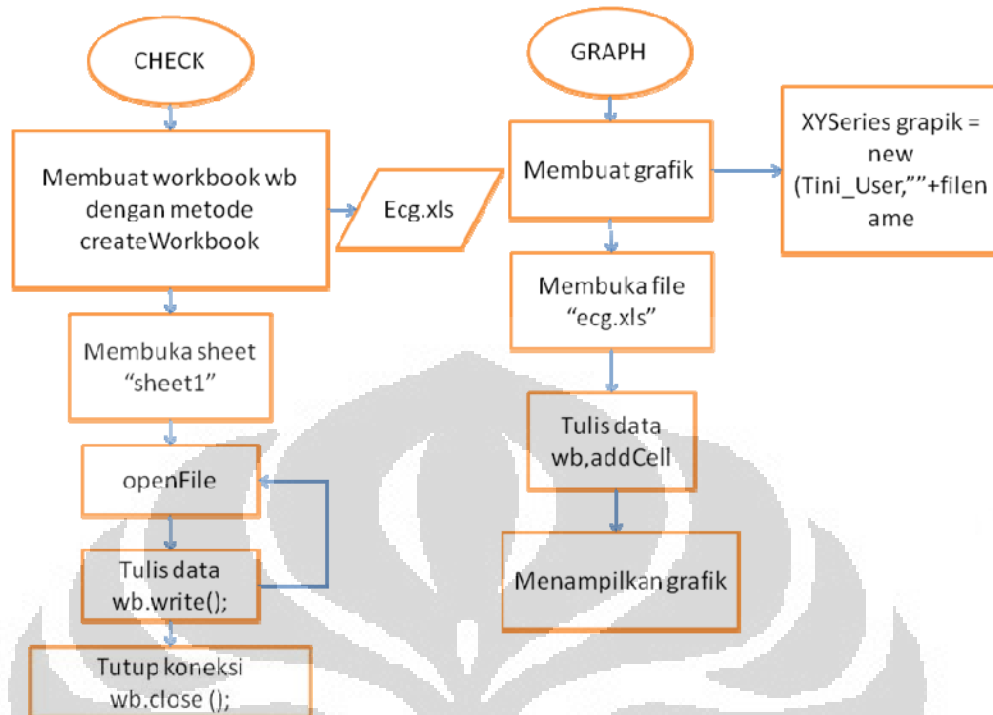
| Frekuensi (Hz) | Vout (V) | A |
|----------------|----------|------|
| 0.5 | 1.5 | 0.75 |
| 1 | 1.5 | 0.75 |
| 2 | 1.5 | 0.75 |
| 3 | 1.5 | 0.75 |
| 4 | 1.5 | 0.75 |
| 5 | 1.5 | 0.75 |
| 6 | 1.5 | 0.75 |
| 7 | 1.5 | 0.75 |
| 8 | 1.5 | 0.75 |
| 9 | 1.5 | 0.75 |
| 10 | 1.5 | 0.75 |
| 11 | 1.5 | 0.75 |
| 12 | 1.2 | 0.60 |
| 13 | 1.2 | 0.60 |
| 14 | 1 | 0.50 |
| 15 | 1 | 0.50 |
| 16 | 0.8 | 0.40 |
| 17 | 0.8 | 0.40 |
| 18 | 0.7 | 0.35 |
| 19 | 0.7 | 0.35 |
| 20 | 0.6 | 0.30 |
| 21 | 0.6 | 0.30 |
| 22 | 0.6 | 0.30 |
| 23 | 0.6 | 0.30 |
| 24 | 0.6 | 0.30 |
| 25 | 0.6 | 0.30 |
| 26 | 0.6 | 0.30 |
| 27 | 0.6 | 0.30 |
| 28 | 0.5 | 0.25 |
| 29 | 0.5 | 0.25 |
| 30 | 0.5 | 0.25 |
| 31 | 0.5 | 0.25 |
| 32 | 0.4 | 0.20 |
| 33 | 0.4 | 0.20 |
| 34 | 0.4 | 0.20 |
| 35 | 0.4 | 0.20 |
| 36 | 0.3 | 0.15 |
| 37 | 0.3 | 0.15 |
| 38 | 0.3 | 0.15 |
| 39 | 0.3 | 0.15 |
| 40 | 0.3 | 0.15 |
| 41 | 0.3 | 0.15 |

| | | |
|----|-----|------|
| 42 | 0.3 | 0.15 |
| 43 | 0.3 | 0.15 |
| 44 | 0.2 | 0.10 |
| 45 | 0.2 | 0.10 |
| 46 | 0.2 | 0.10 |
| 47 | 0.2 | 0.10 |
| 48 | 0.2 | 0.10 |
| 49 | 0.2 | 0.10 |
| 50 | 0.2 | 0.10 |
| 51 | 0.2 | 0.10 |
| 52 | 0.2 | 0.10 |
| 53 | 0.2 | 0.10 |
| 54 | 0.2 | 0.10 |
| 55 | 0.2 | 0.10 |
| 56 | 0.2 | 0.10 |
| 57 | 0.2 | 0.10 |
| 58 | 0.2 | 0.10 |
| 59 | 0.2 | 0.10 |
| 60 | 0.2 | 0.10 |
| 61 | 0.2 | 0.10 |
| 62 | 0.2 | 0.10 |
| 63 | 0.2 | 0.10 |
| 64 | 0.2 | 0.10 |
| 65 | 0.2 | 0.10 |
| 66 | 0.2 | 0.10 |

LAMPIRAN 5

Flowchart





LAMPIRAN 6

Tini_Thread.java

```

import java.io.*;
import java.net.*;

public class Tini_Thread extends Thread {

    public Tini_Thread(BufferedReader in, Tini_User tini) {
        m_in = in;
        m_tini_Client = tini;
    }

    /**
    * Start thread */

    public void run() {
        String strBuffer;
        for(;;) {
            try {
                strBuffer = m_in.readLine();
                if(strBuffer != null) {
                    m_tini_Client.log(strBuffer);
                }
            }
            else {
                System.out.println("Connection closed");
                break;
            }
        }
        catch(IOException e) {
            System.out.println("IOException by server");
            break;
        }
    }
    //m_tini_Client.closeConnection();
    System.out.println("Bye");
}

/** Input stream from socket */
private BufferedReader m_in;
/** Tini_Client for callbacks */
private Tini_User m_tini_Client;
}

```

LAMPIRAN 7

BufferReader.java

```

import com.dalsemi.system.TINIOS;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.Socket;
import javax.comm.CommPortIdentifier;
import javax.comm.SerialPort;

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author Sastra
 */
class BufferReader extends Thread{

    private static final int INPUT_BUF = 1024;

    private SerialPort sp;
    private InputStream spin;
    private OutputStream spout;

    //socket
    private Socket s;
    private InputStream sin;
    private OutputStream sout;

    private BufferReader() throws Exception{
        //identifikasi serial port
        sp =
        (SerialPort)CommPortIdentifier.getPortIdentifier("serial0").open("
        BufferReader",5000);

        TINIOS.setRTSCTSFlowControlEnable(0, true);
        //set parameter
        sp.setSerialPortParams(19200, SerialPort.DATABITS_8,
        SerialPort.STOPBITS_1,SerialPort.PARITY_NONE);
        //set flow control
        sp.setFlowControlMode(SerialPort.FLOWCONTROL_NONE);

        //inisialisasi input dan output stream

        spin = sp.getInputStream();
        spout = sp.getOutputStream();

        //set 100 milisokon timeout
        sp.enableReceiveTimeout(100);
    }
}

```

```

//set threshold
sp.enableReceiveThreshold(INPUT_BUF);

//konek ke socket
Socket s = new Socket("152.118.167.83",23);
sin = s.getInputStream();
sout = s.getOutputStream();
SerialReader sr = new SerialReader(this,
INPUT_BUF).start();

super.start();

}
private volatile boolean running = true;
private int serialTotal = 0;
private int networkTotal = 0;
public void run() {

while (running) {
try {
Thread.sleep(100);
} catch (InterruptedException ie) {}
System.out.println("Bytes received from
serial:"+serialTotal);
System.out.println("Bytes received from
network:"+networkTotal);
}
}

class SerialReader implements Runnable {
private byte[] serBuf;
private Thread maint;
private SerialReader(Thread maint, int size) {
serBuf = new byte[size];
this.maint = maint;
}
public void run() {
while (running) {
try {
// Read all available data in serial input buffer
int count = spin.read(serBuf, 0, serBuf.length);
if (count > 0) {
// Blast serial data to network server
sout.write(serBuf, 0, count);
serialTotal += count;
}
} catch (IOException ioe) {
// Trouble communicating with server
System.out.println(ioe.getMessage());
ioe.printStackTrace();
running = false;
maint.interrupt();
break;
}
}
}
}

```

```
private SerialReader start() {  
    throw new UnsupportedOperationException("Not yet  
    implemented");  
}  
}
```

