

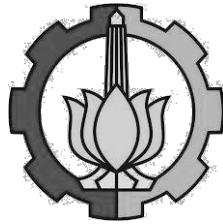
LAPORAN TUGAS AKHIR

SISTEM MONITORING SUHU DAN JUMLAH GELAS PADA MESIN PRODUKSI TEH TAWAR KEMASAN BERBASIS MIKROKONTROLER ATMega32

ERCHE BERLIAN FAJARINDAH
10 51 15 000 00 025

Dosen Pembimbing:
Ir. Ronny Dwi Noriyati, M.Kes.
Dwi Oktavianto Wahyu Nugroho, ST., M.T.

PROGRAM STUDI D3 TEKNOLOGI INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



LAPORAN TUGAS AKHIR

SISTEM MONITORING SUHU DAN JUMLAH GELAS PADA MESIN PRODUKSI TEH TAWAR KEMASAN BERBASIS MIKROKONTROLER ATMega32

ERCHE BERLIAN FAJARINDAH
10 51 15 000 00 025

Dosen Pembimbing:
Ir. Ronny Dwi Noriyati, M.Kes.
Dwi Oktavianto Wahyu Nugroho, S.T., M.T.

PROGRAM STUDI D3 TEKNOLOGI INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

LEMBAR PENGESAHAN

LEMBAR PENGESAHAN

SISTEM MONITORING SUHU DAN JUMLAH GELAS PADA MESIN PRODUKSI TEH TAWAR KEMASAN BERBASIS MIKROKONTROLER ATMega32

TUGAS AKHIR

Oleh:

ERCHE BERLIAN FAJARINDAH
NRP. 10 51 15 000 00 025

Surabaya, 19 Juli 2018
Mengetahui / Menyetujui

Dosen Pembimbing I

Ir.Ronny Dwi Noriyati M.Kes

NIP. 19571126 198403 2 002

Dosen Pembimbing II

Dwi Oktavianto W.N, ST, MT

NPP. 1983201711054



Kepala Departemen
Teknik Instrumentasi, FV

Dr.Ir. Purwadi Agus Darwito, M.Sc

NIP. 19620822 198803 1 001

LEMBAR PERSETUJUAN

SISTEM MONITORING SUHU DAN JUMLAH GELAS PADA MESIN PRODUKSI TEH TAWAR KEMASAN BERBASIS MIKROKONTROLER ATMEGA32

TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat

Memperoleh Gelar Ahli Madya

pada

Departemen Teknik Instrumentasi

Fakultas Vokasi

Institut Teknologi Sepuluh Nopember

Oleh:

ERCHE BERLIAN FAJARINDAH

NRP 10 51 15 000 00 025

Disetujui Tim Penguji: Tanggal Ujian: 19 Juli 2018

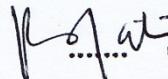
Periode Wisuda: September 2018

Ir. Ronny Dwi Noriyati, M.Kes

Dwi Oktavianto W,N, S.T., M.T

Dr. Ir. Totok Soehartanto, DEA.

Sefi Novendra Patrialova, S.Si., M.T

 (Pembimbing I)

 (Pembimbing II)

 (Penguji I)

 (Penguji II)

SURABAYA

19 Juli 2018

ABSTRAK

SISTEM MONITORING SUHU DAN JUMLAH GELAS PADA MESIN PRODUKSI TEH TAWAR KEMASAN BERBASIS MIKROKONTROLER ATMEGA32

Nama : Erche Berlian Fajarindah
NRP : 10 51 15 000 00 025
Jurusan : Teknik Instrumentasi, ITS Surabaya
Pembimbing : Ir. Ronny Dwi Noriyati, M.Kes
Pembimbing : Dwi Oktavianto Wahyu Nugroho, ST, MT

Abstrak

Telah dilakukan perancangan sistem monitoring *temperature* dan *counter glass* pada mesin produksi teh tawar kemasan, sistem monitoring ini bertujuan untuk menampilkan data dari hasil pembacaan sensor suhu pada dua proses yakni proses pemasakan teh dan *heat exchanger*. Pada sistem monitoring *temperature* menggunakan alat ukur sensor termokopel baut tipe K dengan pengkondisian sinyal MAX6675. Dari kedua sensor tersebut terdapat perbedaan *setpoint* suhu yang diukur, yakni sebesar 70 °C untuk proses pemasakan dan sebesar sekitar 28 °C – 32 °C untuk prndinginan pada *heat exchanger*. Sedangkan untuk *counter glass* digunakan sensor *proximity*. Dalam penampilan display *Human Mechine Interface* (HMI) menggunakan LCD TFT dan sebagai penyimpanan hasil monitoring memanfaatkan SD Card Slot yang telah disediakan oleh LCD TFT 2.8” tersebut.

Kata Kunci: Monitoring Sistem, Temperature, Counter Glass.

ABSTRACT

***MONITORING SYSTEM OF TEMPERATURE AND
COUNTER GLASS ON TEA PRODUCTION MACHINE
BASED ON ATMEGA32 MICROCONTROLLER***

Name : Erche Berlian Fajarindah
NRP : 10 51 15 000 00 025
Department : Instrumentation Engineering, ITS Surabaya
Advisor : Ir. Ronny Dwi Noriyati, M.Kes
Advisor : Dwi Oktavianto Wahyu Nugroho, ST, MT

Abstract

The design of a temperature monitoring system and counter glass has been carried out on a packaged tea production machine, this monitoring system aims to display data from the results of temperature sensor readings in two processes namely the tea cooking process and heat exchanger. In a temperature monitoring system using a K type thermocouple sensor gauge with MAX6675 signal conditioning. From these two sensors there is a difference in the setpoint of the measured temperature, which is 70 °C for the cooking process and for about 28 °C – 32 °C for cooling of the heat exchanger. As for the glass counter used proximity sensor. In the appearance of the Human Machine Interface (HMI) display using TFT LCD and as a result storage monitoring utilizes the SD Card Slot that has been provided by the 2.8" TFT LCD.

Keywords: *Monitoring System, Temperature, Counter Glass*

KATA PENGANTAR

KATA PENGANTAR

Segala puji dan syukur kehadirat Allah SWT atas limpahan rahmat, hidayah, dan kasih sayang-Nya serta shalawat dan salam atas junjungan Nabi Muhammad SAW atas selesainya Tugas Akhir dengan judul:

“SISTEM MONITORING SUHU DAN JUMLAH GELAS PADA MESIN PRODUKSI TEH KEMASAN BERBASIS MIKROKONTROLER ATMEGA32”

Sampai selesainya Tugas Akhir ini ada berbagai pihak yang telah membantu baik dukungan secara moril dan fisik, tidak lupa terima kasih penulis berikan pada:

1. Dr. Ir. Purwadi Agus Darwito, M.Sc. selaku Kepala Departemen Teknik Instrumentasi FV-ITS.
2. Ir Ronny Dwi Noriyati, M.Kes selaku Deosen Pembimbing I, atas segala bimbingannya, kerja sama, nasehat, dan bantuannya selama ini
3. Dwi Oktavianto Wahyu Nugroho, S.T., M.T. selaku Dosen Pembimbing II, atas segala bimbingannya, kerja sama, nasehat, dan bantuannya selama ini.
4. Dr. Ir. Totok Soehartanto., DEA. selaku Dosen Penguji I, atas segala bimbingannya, kerja sama, nasehat, dan bantuannya selama ini.
5. Sefi Novendra Patrialova, S.T, M.T selaku Dosen Penguji II, atas segala bimbingannya, kerja sama, nasehat, dan bantuannya selama ini.
6. Bapak/Ibu dosen yang telah menjadi media transfer ilmu sehingga kami dapat merampungkan jenjang perkuliahan hingga terselesaikannya Tugas Akhir ini.
7. Kedua orang tua yang senantiasa memberikan dukungan dan doa.
8. Kakak saya, Wahyu Panca Nugraha yang memberikan motivasi untuk penggerjaan Tugas Akhir ini.
9. Seluruh staf pegawai Teknik Instrumentasi atas semua bentuk kerja samanya selama masa kuliah.

10. Sahabat-sahabat saya Rossa, Lindra, dan Nisa yang senantiasa bersedia mendengar curahan hati saya ketika proses penggerjaan Tugas Akhir ini mengalami gangguan.
11. Seluruh teman-teman dan segala pihak yang tidak dapat penulis sebut satu persatu terima kasih atas segala bantuannya.

Penulis menyadari bahwa penyusunan laporan ini masih jauh dari sempurna. Untuk itu saran serta kritik yang membangun sangat diharapkan. Semoga laporan ini bermanfaat bagi pembacanya dan dapat menjadi refrensi untuk pembacanya. Akhir kata penulis mohon maaf atas setiap kesalahan yang dilakukan selama pelaksanaan sampai penyusunan laporan ini.

Surabaya, 19 Juli 2018

Penulis

Erche Berlian Fajarindah
NRP 10 51 15 000 00 025

DAFTAR ISI

DAFTAR ISI

LEMBAR PENGESAHAN	i
LEMBAR PERSETUJUAN.....	ii
ABSTRAK	iii
ABSTRACT	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	x
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah.....	2
1.5 Manfaat	3
BAB II DASAR TEORI	3
2.1 Sistem Monitoring.....	3
2.2 Sistem Akuisisi Data.....	5
2.3 Komunikasi Data.....	6
2.3.1 Komponen Komunikasi Data.....	6
2.3.2 Transmisi Data	6
2.4 Penggunaan Komponen pada <i>system monitoring</i>	7
2.4.1 Mikrokontroler ATMega32.....	7
2.4.2 Code Vision AVR	10
2.4.3 LCD TFT 2.8”	12
BAB III PERANCANGAN DAN PEMBUATAN ALAT	15
3.1 <i>Design Sistem Plan</i>	15
3.2 <i>Piping and instrumentation Diagram</i>	16
3.3 Perancangan Sistem Monitoring	17
3.3.1 FlowChart Perancangan Sistem Monitoring	17
3.4 Penggerjaan Sistem Monitoring	21
3.4.1 Penggerjaan Perangkat Lunak.....	21
3.4.2 Penggerjaan Design Tampilan LCD	23
3.4.3 Penggerjaan Design Penyimpanan Data	31

BAB IV ANALISIS DATA DAN PEMBAHASAN	35
4.1 Hasil Rancang Bangun Alat	35
4.1.1 Validasi Tampilan	35
4.1.2 Validasi Proses Penyimpanan Data	40
4.2 Komunikasi Data	47
4.2.1 Penerimaan Data Dari Sensor Ke <i>Display</i>	47
4.2.2 Jenis Komunikasi Data	58
BAB V KESIMPULAN DAN SARAN	51
5.1 Kesimpulan	51
5.2 Saran	51
DAFTAR PUSTAKA	
LAMPIRAN	
BIODATA PENULIS	

DAFTAR GAMBAR

DAFTAR GAMBAR

Gambar 2.1 Diagram Sistem Pengukuran	3
Gambar 2.2 Diagram Blok Sistem Akuisisi Data.....	5
Gambar 2.3 Transmisi Simplex.....	6
Gambar 2.4 <i>Transmisi Half Duplex</i>	7
Gambar 2.5 <i>Transmisi Full Duplex</i>	7
Gambar 2.6 Mikrokontroler ATMega-32.....	8
Gambar 2.7 Konfigurasi Pin ATMega-32	9
Gambar 2.8 LCD TFT 2.8"	12
Gambar 2.9 PinOut LCD TFT 2.8"	13
Gambar 3.1 P&ID Mesin Produksi Teh Tawar Kemasan	17
Gambar 3.2 FlowChart Perancangan Sistem Monitoring	19
Gambar 3.3 Konfigurasi pin ATMega32 pada plant	22
Gambar 3.4 Design Tampilan Suhu	24
Gambar 3.5 <i>Design</i> tampilan suhu pada <i>heat exchanger</i>	26
Gambar 3.6 Design Tampilan Kekeruhan	27
Gambar 3.7 Design Tampilan Counter Glass	29
Gambar 3.8 <i>Wiring diagram design hardware</i>	30
Gambar 3.9 <i>Design DataLogger</i>	33
Gambar 4.1 Hasil Tampilan Panel Monitoring	35
Gambar 4.2 Hasil Penyimpanan Data Suhu pada <i>data logger</i> ..	40
Gambar 4.3 Grafik penyimpanan data suhu pemasakan teh	42
Gambar 4.4 Hasil Penyimpanan Data tingkat kepekatan pada <i>data logger</i>	42
Gambar 4.5 Hasil Penyimpanan data <i>counter glass</i> pada <i>data logger</i>	43
Gambar 4.6 Grafik penyimpanan data <i>counter</i>	44
Gambar 4.7 Hasil penyimpanan data suhu <i>heat exchanger</i>	45
Gambar 4.8 Diagram blok sistem komunikasi data	48

DAFTAR TABEL

DAFTAR TABEL

Tabel 4.1 Validasi tampilan suhu <i>heat exchanger</i>	36
Tabel 4.2 Validasi tampilan suhu pemasakan	37
Tabel 4.3 Validasi tampilan <i>counter glass</i>	38
Tabel 4.4 Validasi tampilan lcd system monitoring dengan lcd panel <i>system pengendalian</i>	39
Tabel 4.5 Penyimpanan data suhu pemasakan.....	41
Tabel 4.6 Penyimpanan data perhitungan gelas.....	44
Tabel 4.7 Penyimpanan data suhu <i>heat exchanger</i>	46

BAB I

PENDAHULUAN

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kekayaan alam Indonesia merupakan suatu sumber daya alam yang harus dijaga kelestariannya. Salah satunya dibidang perkebunan yang merupakan sektor perluasan dari bidang pertanian. Banyak jenis tanaman perkebunan yang tumbuh subur dan dikembangkan di Indonesia. Teh merupakan tanaman perkebunan yang mempunyai nilai ekonomi yang tinggi dan dapat dikembangkan lebih luas. Teh mempunyai nilai ekonomi tinggi terbukti dengan meningkatnya jumlah devisa negara karena telah dieksport ke luar negeri. Dengan adanya nilai jual yang tinggi tersebut maka tanaman teh banyak dibudidayakan di Indonesia.

Teh sebagai bahan minuman dibuat dari pucuk muda daun teh yang telah mengalami proses pengolahan seperti pelayungan, penggilingan dan pengeringan. Manfaat yang dihasilkan dari minuman teh adalah memberi rasa segar dan dapat memulihkan kesehatan badan yang juga terbukti tidak menimbulkan dampak negatif^[3].

Sehubungan dengan hal tersebut membuat kami termotivasi untuk membuat rancang bangun mesin produksi teh tawar kemasan dengan memanfaat kekayaan alam Indonesia yang ada. Dalam perkembangannya, mesin pembuatan minuman di industri kian hari kian meningkat seiring dengan kebutuhan akan sistem yang dapat melakukan monitoring sebuah mesin di industri pun juga semakin meningkat dari tahun ketahun. Hal ini sejalan dengan perkembangan peradaban manusia yang tak lagi mengenal batas dan jarak sehingga menuntut sebuah sistem yang cerdas, efisien, informatif, dan fungsional sehingga dapat mempermudah hidup manusia^[11]. Sehingga terciptalah sebuah sistem monitoring yang memanfaatkan mikrokontroler dikarenakan biayanya murah dan luas pemanfaatannya.

Oleh karena itu dibuatkan tugas akhir dengan judul “Sistem Monitoring Suhu Dan Jumlah Gelas Pada *Mesin Produksi Teh Tawar Kemasan Berbasis Mikrokontroler*”. Hal ini digunakan untuk memonitoring besarnya temperature yang terdapat pada tank penyeduhan teh, temperature pada *heat exchanger* dan juga proses *packaging* teh kemasan agar semua data yang dihasilkan dapat terekam dan ditampilkan..

1.2 Rumusan Masalah

Berdasarkan latar belakang yang dijelaskan diatas, maka rumusan masalah dalam Tugas Akhir ini adalah bagaimana cara merancang sistem monitoring pada Mesin Produksi Teh Tawar Kemasan?

1.3 Tujuan

Tujuan utama dari rancang bangun alat ini adalah untuk memberikan solusi pada rumusan masalah yaitu untuk merancang sistem monitoring pada Mesin Produksi Teh Tawar Kemasan

1.4 Ruang Lingkup Tugas Akhir

Adapun batas ruang lingkup dari penelitian tugas akhir ini yaitu hanya membahas mengenai sistem monitoring suhu pemasakan teh, suhu pada *heat exchanger* dan jumlah gelas pada proses *packaging* perancangan mesin produksi teh tawar kemasan.

1.5 Manfaat

Adapun manfaat dari Tugas Akhir ini adalah sebagai berikut:

1. Sebagai alat penunjang untuk praktikum Sistem Pneumatic dan Hidrolik serta mata kuliah lain yang berhubungan dengan proses pada alat.
2. Sebagai sarana pengenalan dan pembelajaran tentang alat yang ada dalam dunia industri.

BAB II
TINJAUAN PUSTAKA

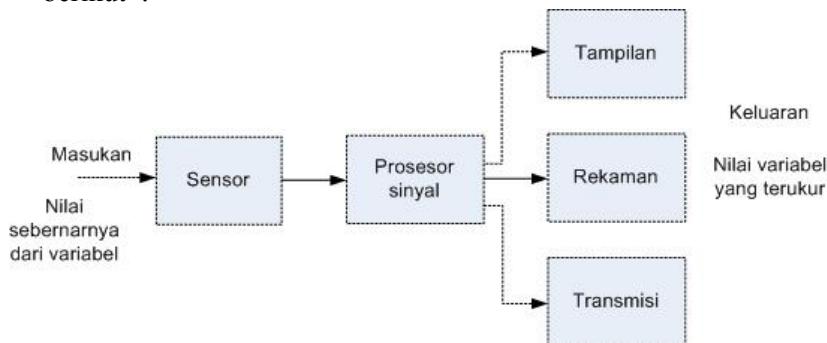
BAB II

DASAR TEORI

2.1 Sistem Monitoring

Monitoring merupakan pemantauan pada suatu proses atau sistem yang mencapai hasil yang diharapkan baik secara kualitas dan kuantitas dengan efektif. Monitoring dilakukan dengan tujuan untuk meminimalkan gangguan jika terdeteksi suatu gangguan atau kegagalan dalam suatu sistem. Monitoring selain berfungsi sebagai pengawasan juga berfungsi untuk merekam apa yang terjadi pada sistem yang dimonitor dalam bentuk data table maupun grafik yang ditampilkan dalam bentuk *display*^[7].

Untuk mempermudah dalam melakukan pengukuran maka perlu diketahui mengenai prinsip dasar pengukuran seperti gambar berikut :



Gambar 2.1 Diagram Blok Sistem Pengukuran^[1]

Sensor

Sensor adalah elemen sistem yang secara efektif berhubungan dengan proses dimana suatu variabel sedang diukur dan menghasilkan suatu keluaran dalam bentuk tertentu tergantung pada variabel masukannya, dan dapat digunakan oleh bagian sistem pengukuran yang lain untuk mengenali nilai variabel tersebut. sebagai contoh adalah sensor termokopel yang memiliki

masukan berupa temperatur serta keluaran berupa gaya gerak listrik (GGL) yang kecil. GGL yang kecil ini oleh bagian sistem pengukuran yang lain dapat diperkuat sehingga diperoleh pembacaan pada alat ukur.

Prosesor sinyal

Prosesor yang dimaksud disini merupakan elemen sistem instrumentasi yang akan mengambil keluaran dari sensor dan mengubahnya menjadi suatu bentuk besaran yang cocok untuk tampilan dan transmisi selanjutnya dalam beberapa sistem kontrol. Seperti pengondisi sinyal (*signal conditioner*) merupakan salah satu bentuk prosesor sinyal.

Penampil data

Elemen ini menampilkan nilai-nilai yang terukur dalam bentuk yang bisa dikenali oleh pengamat, seperti melalui sebuah alat penampil (*display*), misalnya sebuah jarum penunjuk (*pointer*) yang bergerak disepanjang skala suatu alat ukur. Selain ditampilkan, sinyal tersebut juga dapat direkam, misalnya pada kertas perekam diagram. Untuk penyimpanan disini dapat menggunakan *data logger* yang dapat diakses menggunakan kartu SD atau semacamnya dan mode penyimpanan menggunakan *database* yang dapat diakses secara *online*. Sedangkan yang dimaksud transmisi disini ialah proses pengiriman data yang telah dilakukan oleh proses pengendalian menggunakan *controller* misalnya, bagaimana bisa sampai *kedisplay* ialah tugas dari *sub-elemen* transmisi ini. Pada umumnya, proses transmisi ini menggunakan komunikasi data sesuai dengan kebutuhan *user*.

2.2 Sistem Akuisisi Data

Sistem akuisisi data dapat didefinisikan sebagai suatu sistem yang berfungsi untuk mengambil, mengumpulkan dan menyiapkan data, hingga memprosesnya untuk menghasilkan data yang dikehendaki. Jenis serta metode yang dipilih pada umumnya bertujuan untuk menyederhanakan setiap langkah yang dilaksanakan pada keseluruhan proses.

Suatu sistem akuisisi data pada umumnya dibentuk sedemikian rupa sehingga sistem tersebut berfungsi untuk mengambil, mengumpulkan dan menyimpan data dalam bentuk yang siap untuk diproses lebih lanjut. Sebuah sistem akuisisi data atau biasa dikenal *Data Acquisition System* (DAS) merupakan system instrumentasi elektronik yang terdiri dari sejumlah elemen yang secara bersama-sama bertujuan untuk melakukan pengukuran, menyimpan, dan mengolah hasil pengukuran. Secara aktual DAS berupa *interface* antara lingkungan *analog* dengan lingkungan digital. Lingkungan analog meliputi tranduser dan pengkondisian sinyal dengan segala kelengkapannya, sedangkan lingkungan digital meliputi *analog to digital converter* dan selanjutnya pemrosesan digital yang dilakukan oleh mikrokontroler. Komputer yang digunakan untuk system akuisisi transfer pada komputer yang bersangkutan mempengaruhi kinerja dari sistem akuisisi data secara keseluruhan.



Gambar 2.2 Diagram Blok Sistem Akuisisi Data^[1]

Suatu konfigurasi sistem akuisisi data sangat tergantung pada jenis dan jumlah tranduser serta teknik pengolahan yang akan digunakan. Konfigurasi ini dapat lihat dari banyaknya tranduser atau kanal yang digunakan, kecepatan pemrosesan data dan letak masing-masing komponen pada sistem akuisisi data.

2.3 Komunikasi Data

Komunikasi data adalah proses pengiriman dan penerimaan data/informasi dari dua atau lebih *device* seperti komputer, laptop, printer, dan alat komunikasi lain yang terhubung dalam sebuah jaringan. Baik lokal maupun yang luas, seperti internet. Pada dasarnya komunikasi data merupakan proses pengiriman informasi di antara dua titik menggunakan kode biner melewati saluran transmisi, bisa

antara komputer dan komputer, komputer dengan terminal, atau komputer dengan peralatan, atau peralatan dengan peralatan.

2.3.1 *Komponen Komunikasi Data*

Adapun komponen-komponen yang diperlukan guna dapat berlangsungnya komunikasi data yang baik, komponen-komponen tersebut ialah:

1. Pengirim : piranti yang mengirimkan data
2. Penerima : piranti yang menerima data
3. Data yang dikirim : informasi yang akan dipindahkan dapat berupa teks, angka, gambar dan semacamnya.
4. Media pengirimian : media atau saluran yang digunakan untuk mengirimkan data, bisa berupa kabel dll.
5. Protokol : aturan-aturan yang harus disepakati oleh dua atau lebih alatuntuk dapat saling berkomunikasi.

2.3.2 *Transmisi Data*

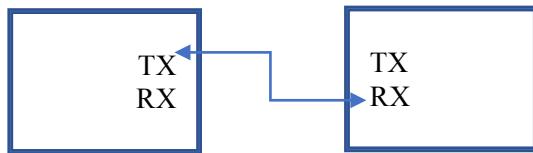
1. *Simplex* : Media komunikasi antara pengirim dan penerima yang bersifat satu arah, dimana sisi pengirim hanya berfungsi sebagai pengirim informasi, sedangkan sisi penerima berfungsi sebagai penerima. Mode ini juga sering disebut *one way only*.



Gambar 2.3 Transmisi Simplex^[1]

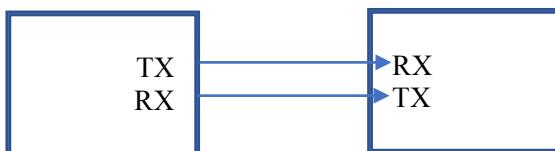
2. *Half Duplex (HDX)* : Pada sistem *Half Duplex* pesan dapat dikirimkan dalam dua arah. Jenis komunikasi half duplex terjadi ketika data dapat mengalir dalam dua arah, namun hanya satu arah

pada satu waktu. Ilustrasi jenis komunikasi *half duplex* tampak pada gambar dibawah ini:



Gambar 2.4 Transmisi *Half Duplex*^[1]

3. Sistem *Ful Duplex* : Pada jenis komunikasi *full duplex*, komunikasi dapat terjadi dalam dua arah secara bersamaan, karena jalur pengiriman dan penerimaan data berbeda. Ilustrasi jenis komunikasi full duplex tampak pada gambar dibawah ini:



Gambar 2.5 Transmisi Full Duplex^[1]

2.4 Penggunaan komponen pada system monitoring

Pada system monitoring mesin produksi teh kemasan ini terdapat beberapa komponen pendukung agar system berjalan dengan lancar, komponen-komponen tersebut ialah sebagai berikut

2.4.1 Mikrokontroler ATMega32

Mikrokontroller ATMEGA32 adalah mikrokontroler yang diproduksi oleh Atmel. Mikrokontroler ini memiliki *clock* dan kerjanya tinggi sampai 16 MHz, ukuran flash memorinya cukup besar, kapasitas SRAM sebesar 2 KiloByte, 32 buah port I/O yang sangat memadai untuk berinteraksi dengan LCD dan *keypad*.



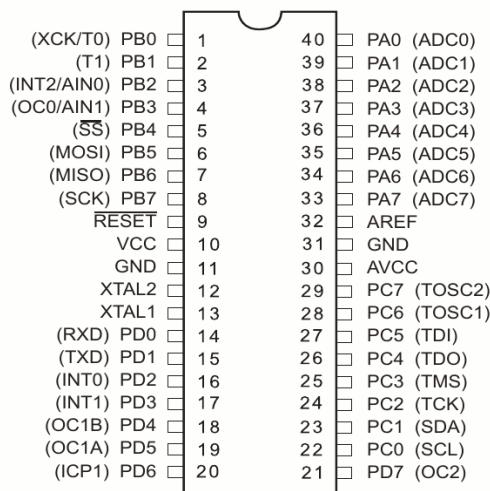
Gambar 2.6 Ic Chip Atmega32^[2]

Semua jenis AVR dilengkapi dengan flash memori sebagai memori program. Kapasitas dari flash memori ini berbeda antara *chip* yang satu dengan *chip* yang lain. Tergantung dari jenis IC yang digunakan. Untuk flash memori yang paling kecil adalah 1 kbytes (ATtiny11, ATtiny12, dan ATtiny15) dan paling besar adalah 128 Kbytes (AT-Mega128). Berikut ini adalah spesifikasi Mikrokontroler AVR ATMega-32 dan konfigurasi pin ATMEGA 128.

Spesifikasi :

- Saluran I/O ada 32 buah, yaitu Port A, Port B, Port C, dan Port D.
- ADC (*Analog to Digital Converter*) 10 bit sebanyak 8 *channel*.
- Tiga buah *Timer/Counter*.
- CPU yang terdiri dari 32 buah *register*.
- Memiliki 131 instruksi yang membutuhkan 1 siklus *clock*.
- *Watchdog timer* dengan osilator *internal*.
- Dua buah *timer/counter* 8 bit, satu buah *timer/counter* 16 bit.
- Tegangan operasi 2.7 V – 5.5 V pada ATmega 16L.
- Internal SRAM sebesar 1 KB.
- Memori *flash* sebesar 32KB dengan kemampuan *Read While Write*.
- Unit interupsi *internal* dan *external*.

- Port antarmuka SPI.
- EEPROM sebesar 512 byte yang dapat diprogram saat operasi.
- Antarmuka komparator analog.
- 4 channel PWM.
- 32x8 *general purpose register*.
- Hampir mencapai 16 MIPS pada Kristal 16 MHz.
- Port USART yang dapat diprogram untuk komunikasi



Gambar 2.7 Konfigurasi Pin pada Atmega32^[2]

Fungsi umum dari susunan pin *microcontroller* ATmega32 adalah sebagai berikut:

1. VCC merupakan catu daya positif.
2. GND sebagai pin *ground* catu daya negatif.
3. Port A (PA0-PA7) merupakan pin I/O dua arah dan dapat diprogram sebagai pin dari ADC.

4. Port B (PBO-PB7) merupakan pin I/O dua arah dan fungsinya sebagai, *timer/counter*, komparator analog, dan SPI.
5. Port C (PCO-PC7) merupakan pin I/O dua arah dan fungsinya sebagai, TWI, komparator analog, dan *timer osilator*.
6. Port D (PDO-PD7) merupakan pin I/O dua arah dan fungsinya sebagai, komparator analog, interupsi eksternal dan komunikasi serial.
7. Reset merupakan pin yang digunakan untuk me-*reset microcontroller*.
8. XTAL 1 dan XTAL 2 sebagai pin *clock* eksternal. Pada *microcontroller* membutuhkan sumber (*clock*) agar dapat mengeksekusi instruksi yang ada di memori. Semakin tinggi nilai kristalnya, maka semakin cepat *microcontroller* tersebut.
9. A VCC sebagai pin tegangan untuk ADC.
10. AREF sebagai pin tegangan referensi.

2.4.1 Code Vision AVR

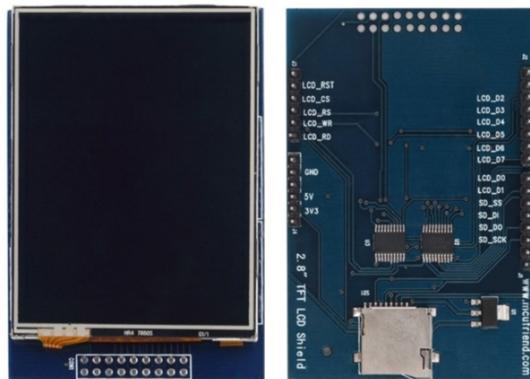
CodeVision AVR pada dasarnya merupakan perangkat lunak pemrograman mikrokontroler keluarga AVR berbasis bahasa C. Ada tiga komponen penting yang telah diintegrasikan dalam perangkat lunak ini: Compiler C, IDE, dan program generator. CodeVision AVR dilengkapi dengan source code editor, compiler, linker dan dapat memanggil Atmel AVR studio dengan debuggernya. Berdasarkan spesifikasi yang dikeluarkan oleh perusahaan pengembangnya, compiler C yang digunakan hampir mengimplementasikan semua komponen standar yang ada pada bahasa C standar ANSI (seperti struktur program, jenis tipe data, jenis operator, dan library fungsi standar berikut penamaannya). Tetapi walaupun demikian, dibandingkan bahasa C untuk aplikasi komputer, compiler C untuk mikrokontroler ini memiliki sedikit perbedaan yang

disesuaikan dengan arsitektur AVR tempat program C tersebut ditanamkan (*embedded*). Khusus untuk library fungsi, disamping library standar (seperti fungsi-fungsi matematik, manipulasi string, pengaksesan memori dan sebagainya), CodeVision AVR juga menyediakan fungsi-fungsi tambahan yang sangat bermanfaat dalam pemrograman antarmuka AVR dengan perangkat luar yang umum digunakan dalam aplikasi kontrol. Beberapa fungsi library yang penting diantaranya adalah fungsi-fungsi untuk pengaksesan LCD, komunikasi I2C, SPI (*Serial Peripheral Interface*) dan lain sebagainya. Untuk memudahkan pengembangan program aplikasi, CodeVision AVR juga dilengkapi IDE yang sangat user friendly. Selain menu-menu pilihan yang umum dijumpai pada setiap perangkat lunak berbasis Windows, CodeVision AVR ini telah mengintegrasikan perangkat lunak berbasis downloader yang bersifat In System Programmer yang dapat digunakan untuk mentransfer kode mesin hasil kompilasi ke dalam sistem memori mikrokontroler AVR yang sedang diprogram.

CodeVision AVR juga menyediakan sebuah fitur yang dinamakan dengan Code Generator atau CodeWizard AVR. Secara praktis, fitur ini sangat bermanfaat membentuk sebuah kerangka program (template), dan juga memberi kemudahan bagi programmer dalam penginisialisasi register-register yang terdapat pada mikrokontroler AVR yang sedang diprogram. Dinamakan Code Generator, karena perangkat lunas CodeVision ini akan membangkitkan kode-kode program secara otomatis setelah fase inisialisasi pada jendela CodeWizard AVR selesai dilakukan. Secara teknis, penggunaan fitur ini pada dasarnya hampir sama dengan application wizard pada bahasa-bahasa pemrograman visual untuk komputer (seperti Visual C, Borland Delphi, dan sebagainya). (Andrianto, 2013)

2.4.2 LCD TFT 2.8”

TFT LCD (*Thin Film Transistor Liquid Crystal Display*) adalah varian dari *Liquid Crystal Display* (LCD) yang menggunakan Transistor Film Tipis (TFT) teknologi untuk meningkatkan kualitas gambar. LCD TFT 2.8” memiliki library yang kompatibel dengan layar sentuh Adafruit, yang mudah digunakan. LCD TFT 2.8” memiliki resolusi lebih dari layar 128x64 hitam dan putih. Layar ini memiliki layar sentuh resistif yang melekat padanya, sehingga pengguna dapat mendeteksi penekanan jari di mana saja pada layar. Memiliki 240x320 piksel dengan kontrol piksel individual. Sementara itu, modul ini mendukung slot kartu mikro SD untuk memperluas penyimpanan.



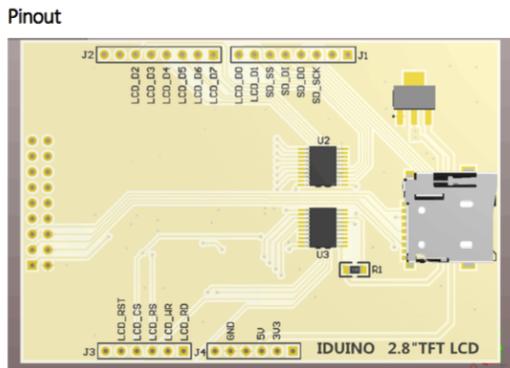
Gambar 2.8 LCD TFT 2.8”^[11]

Gambar 2.8 merupakan bentuk fisik LCD TFT 2.8” tampak depan dan belakang, seperti yang terlihat pada gambar bagian belakang terlihat LCD dilengkapi dengan *Slot SD Card* untuk dapat melakukan penyimpanan yang lebih luas.

Berikut ini adalah spesifikasi LCD TFT 2.8” dan konfigurasi pin LCD TFT 2.8”.

Spesifikasi :

- 2.8" tampilan diagonal LCD TFT
- Memiliki resolusi 240x320
- Memiliki slot Mini SD Card



Gambar 2.9 PinOut LCD TFT 2.8" [11]

Deskripsi dari susunan pin *LCD TFT 2.8"* adalah sebagai berikut:

1. 5V dan 3,3V (VCC) merupakan catu daya positif.
2. GND ialah pin *ground* catu daya negatif.
3. LCD_RST ialah Pin Reset pada LCD
4. LCD_CS ialah pin Chip select pada LCD
5. LCD_RS ialah pin Register select pada LCD
6. LCD_WR ialah pin Penulisan pada LCD
7. LCD_RD ialah pin Read pada LCD
8. LCD_02 – LCD_07 ialah LCD data bit 0-bit 7
9. SD_SS ialah SD *Card Slave Select*
10. SD_DI ialah SD Card Serial Data In
11. SD_DO ialah SD Card Serial Data Out
12. SD_SCK ialah SD Card Serial Clock

Halaman ini sengaja dikosongkan



BAB III

PERANCANGAN DAN PEMBUATAN ALAT

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

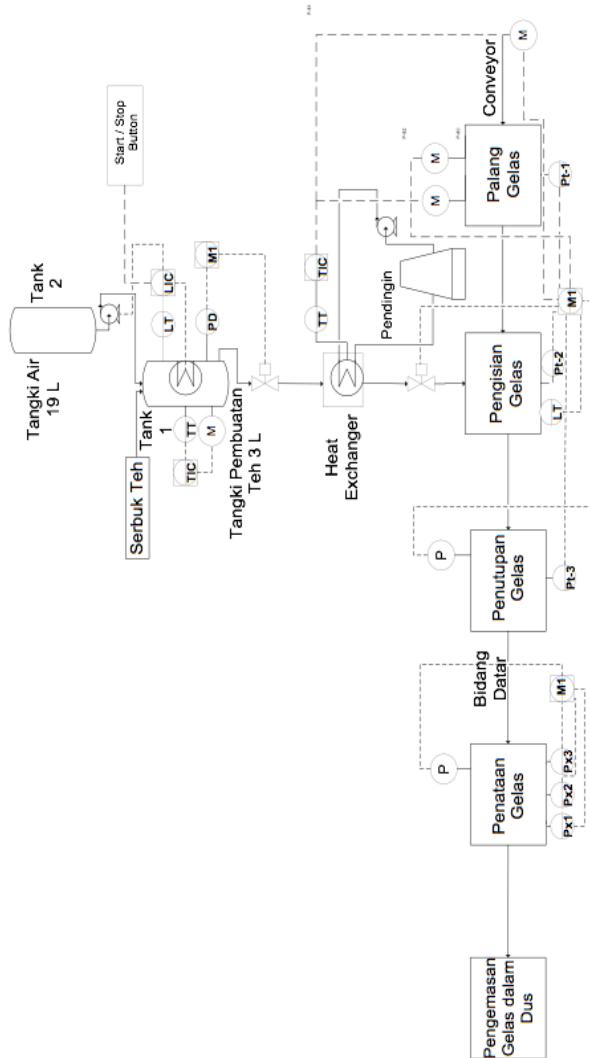
3.1 *Design Sistem Plant*

Pada Mesin Produksi Teh Tawar ini memiliki sejumlah sistem proses diantaranya yaitu tank pemasak teh yang berfungsi untuk memasak air panas dan teh. Air panas diproduksi oleh unit pemanas dengan menggunakan *heater*, sedangkan teh dimasukkan secara manual oleh *user*. Setelah tercampur, suhu teh yang telah dimasak dikendalikan hingga mencapai suhu 70 °C sebagai parameter tingkat masak teh tawar. Setelah dimasak dan mencapai suhu yang telah ditentukan teh akan dialirkan *heat exchanger* untuk didinginkan hingga setpoint 30 derajat *Celcius*. Ketika suhu teh tawar telah mencapai setpoint, maka akan mengaktifkan conveyor agar berjalan. Setelah conveyor berjalan akan dilakukan proses sealing pada gelas, pada proses ini juga terjadi penghitungan jumlah gelas melalui sensor proximity yang terletak pada system penutupan gelas teh. Proses terakhir ialah proses pengepakan yakni dilakukan dengan mengatur setpoint setiap 3 gelas akan langsung didorong dan dimasukkan kedalam kardus yang telah disediakan menggunakan piston pneumatic.

Pada plant ini, sekali produksi menghasilkan sejumlah 6 gelas teh tawar. Semua proses tersebut terutama pada proses pemasakan, pengisian dan pengepakan akan ditampilkan pada *Human Mechine Interface (HMI)* yakni berupa *LCD TFT*. Untuk menghubungkan antara mikrokontroler dengan HMI digunakan komunikasi data, sehingga terdapat bagian yang akan mengatur jalannya komunikasi dari antar mikrokontroler ke HMI dan mengatur HMI itu sendiri.

Pada Tugas Akhir ini dilakukan perancangan sistem monitoring suhu dan jumlah gelas pada mesin produksi teh tawar kemasan. Sistem monitoring yang terdapat pada plan berfungsi sebagai pemberitahuan dan juga untuk membantu agar user lebih mudah membaca kinerja pada proses pengendalian.

3.2 Piping And Instrumentation Diagram



Gambar 3.1 P&ID Mesin Produksi Teh Tawar Kemasan

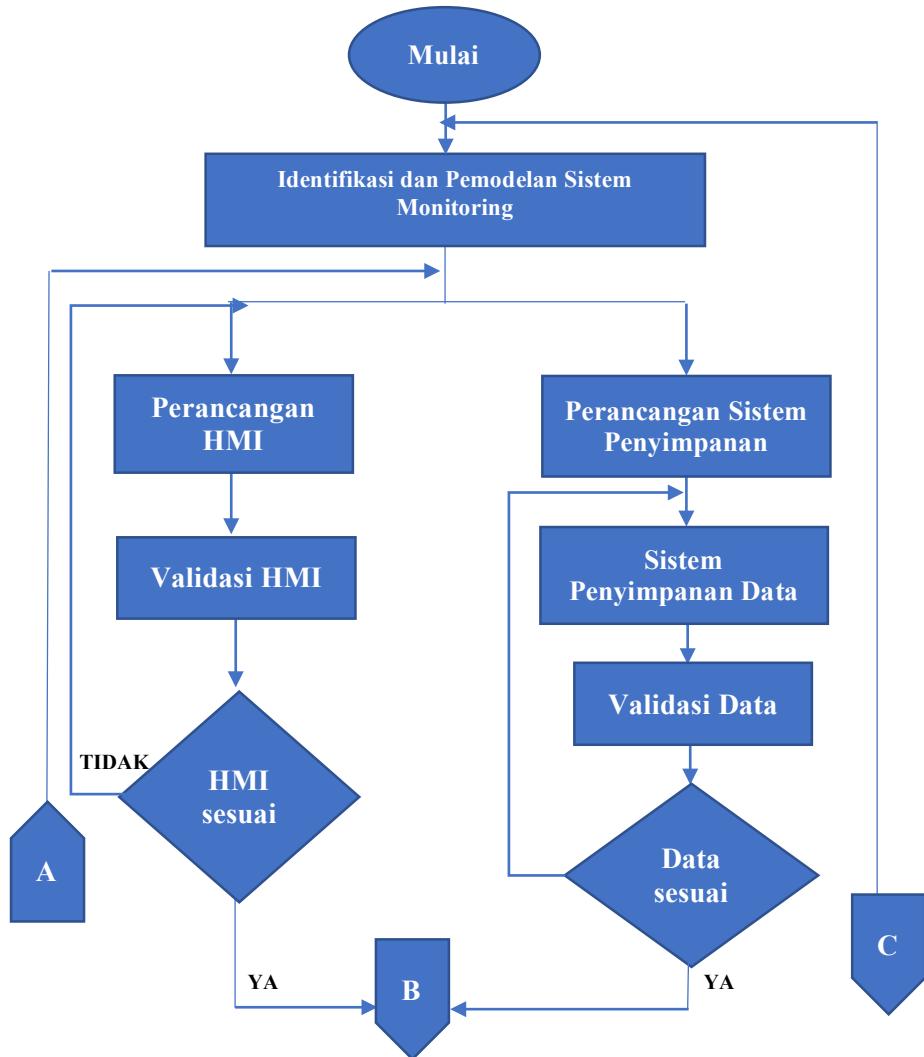
Pada Gambar 3.1 merupakan design P&ID Mesin Produksi Teh Tawar Kemasan. Telah tertera pada gambar mengenai proses pemasakan hingga pengepakan teh tawar kemasan. Pemasakan dilakukan oleh *tank* pertama yang didalamnya terdapat dua sensor yakni sensor *thermocouple* tipe K sebagai sensor pembaca suhu pemasakan teh dan juga phototransistor sebagai sensor pembaca tingkat kepekatan warna teh. Pada proses ini suhu dikendalikan hingga mencapai setpoint sebesar 70 °C untuk suhu dan menggunakan 3 parameter tingkat kekeruhan yakni tidak keruh, keruh, dan sangat keruh. Setelah melalui proses pemasakan, teh tawar yang telah dimasak akan memasuki proses pendinginan pada *heat exchanger* yang suhunya telah dikendalikan pada *setpoint* sebesar 28-30 °C, dengan tujuan agar suhu sesuai dengan kekuatan gelas dalam menahan panas. Ketika suhu teh telah memenuhi *setpoint* yang telah ditentukan maka akan dilakukan pengisian teh pada gelas yang akan disensing oleh sensor *proximity* mengenai isi dari gelas tersebut. Selanjutnya, setelah gelas terisi sesuai dengan *setpoint* yang telah ditentukan maka akan dilakukan penutupan gelas menggunakan *system pneumatic* yang terletak pada mesin *sealer* pada *plan* dengan bantuan kompresor sebagai penyedia tekanan. Setelah itu lanjut pada proses penataan gelas, disini juga akan langsung terjadi proses *counting* menggunakan sensor *proximity* yang terletak diantara proses pengisian sampai pengepakan. Dari keseluruhan system yang telah dijelaskan, ranah dari system monitoring ialah pada suhu pemasakan dan suhu pada *heat exchanger* serta *counter glass* pada proses pengepakan.

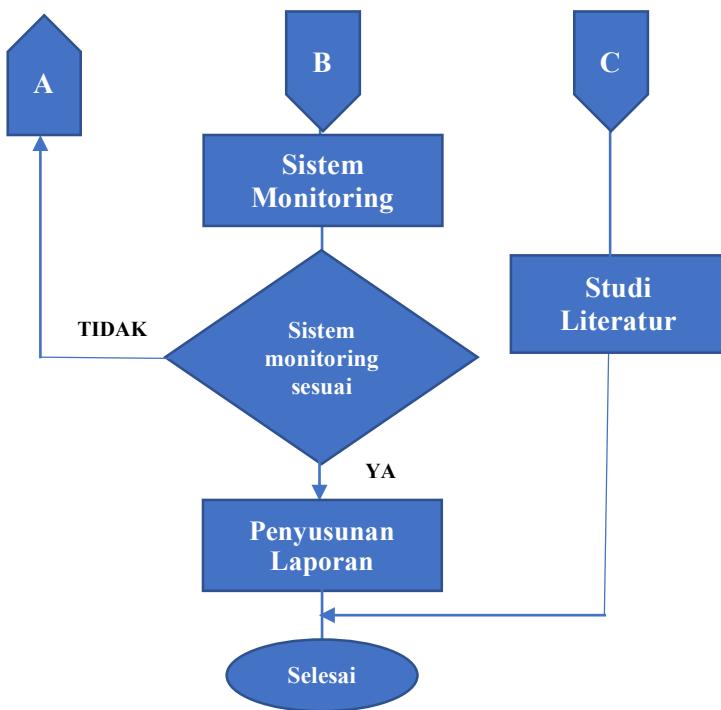
3.3 Perancangan Sistem Monitoring

Perancangan Sistem Monitoring akan dirasa mudah jika terdapat *design flowchart* secara runtut. Berikut akan dijelaskan mengenai flowchart system monitoring.

3.3.1 Flow chart Perancangan Sistem Monitoring

Pada sub-bab ini dijelaskan mengenai prosedur tahapan dalam penelitian tugas akhir yang dilakukan. Dengan *flow chart* dibawah ini





Gambar 3.2 *Flow Chart* Proses Perancangan Sistem Monitoring

Adapun keterangan Gambar 3.2 mengenai digram alir dapat dijelaskan sebagai berikut :

a. Studi Literatur

Dilakukan kajian terhadap metode-metode, konsep, atau teori yang terkait dengan penelitian yang dilakukan, baik yang bersumber dari jurnal, laporan penelitian, maupun buku-buku yang memiliki bahasan yang sesuai dengan tema penelitian. Studi literatur dilakukan selama proses, sedari akan memulai pekerjaan hingga perancangan selesai dengan tujuan agar segala proses perancangan memiliki acuan yang jelas.

b. Identifikasi dan Pemodelan Sistem Monitoring

Dilakukan identifikasi dan pemodelan sistem monitoring pada *plant* teh tawar kemasan, perancangan dilakukan dengan desain visualisasi data yang akan ditampilkan dan desain penyimpanan data yang akan disimpan.

c. Pembuatan Sistem Monitoring

Dilakukan pembuatan sistem monitoring pada *plant* teh tawar kemasan, pembuatan sistem monitoring dilakukan dengan membuat *Hardware* dan *software* dari sistem monitoring, pembuatan *hardware* meliputi pembuatan *board* pengganti *module* sebagai *triggered* agar LCD bisa menyala dengan semestinya dan pembuatan *software* meliputi pembuatan *visualisasi* data yang akan ditampilkan berdasarkan hasil baca daripada sensor juga membuat file penyimpanan data pada LCD.

d. Perancangan Human Machine Interface (HMI)

Perancangan HMI ini bertujuan untuk agar hasil data yang telah dibaca oleh sensor dapat tersampaikan pada LCD dan juga agar *display* daripada HMI mudah dimengerti oleh user, sehingga user dapat dengan mudah mengerti keadaan plan seperti apa dan dapat memudahkan mengidentifikasi terjadinya error hanya melalui HMI yang ada.

e. Validasi Human Machine Interface (HMI)

Validasi *Human Machine Interface* (HMI) dilakukan bertujuan untuk mengetahui kesesuaian bentuk *visualisasi* apakah hasil pembacaan sensor dengan data yang telah ditampilkan LCD, apabila tidak sesuai maka akan dilakukan evaluasi terhadap perancangan *Human Machine Interface* (HMI)

f. Perancangan Sistem Penyimpanan

Pada proses ini bertujuan agar data yang direkam runtut sehingga memudahkan user mengidentifikasi sistem yang error dari data yang telah direkam. Tujuan lain dari adanya sistem penyimpanan ini ialah sebagai acuan ketika sedang dilakukan maintanance pada plan.

g. Sistem Penyimpanan Data

Sistem penyimpanan data yang dimaksud perancangan mulai dari format data hingga memori yang digunakan. Seperti misalnya akan menggunakan data logger melalui *SD Card* ataukah *database* yang dapat diakses secara *online*.

h. Validasi Sistem Penyimpanan Data

Pada proses validasi sistem penyimpanan akan dilakukan suatu tindakan yang membuktikan bahwa data yang telah ditampilkan pada display telah sesuai dengan data yang masuk dan disimpan oleh sistem penyimpanan data.

i. Sistem Monitoring

Sistem Monitoring dilakukan jika validasi pada proses perancangan *Human Machine Interface* (HMI) dan validasi pada proses perancangan sistem penyimpanan telah sesuai dan berhasil. Jika masih terdapat salah satu proses yang mengalami error ataupun ketidaksesuaian maka perlu dilakukan pemeriksaan pada identifikasi dan pemodelan sistem monitoring.

j. Pembuatan Laporan

Setelah semua dirasa sesuai dengan runtutan *flowchart* yang telah dibuat, mulai dari desain hingga memeriksa hasil validasi daripada hasil desain akhir maka dapat dilanjutkan dengan penulisan laporan sebagai bentuk hasil kerja yang juga dapat digunakan sebagai sumber penelitian dimasa mendatang

3.4 Pengerjaan Sistem Monitoring

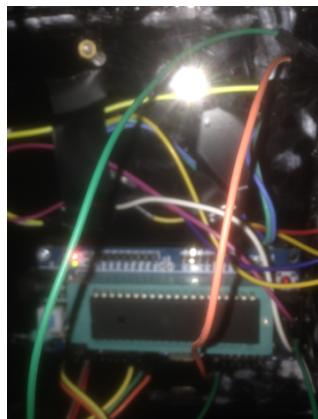
Dalam perencanaan kerja sistem dibagi menjadi dua bagian, yaitu perancangan perangkat keras dan perangkat lunak. Pada perancangan perangkat keras berikut adalah langkah-langkahnya

3.4.1 Pengerjaan Perangkat Lunak

Dalam pengerjaan perangkat lunak yaitu untuk penyimpanan data yang telah terekam dan menampilkan pada pusat kontrol panel HMI dengan menggunakan mikrokontroler AVR ATmega32. Serta perancangan *data logger* sehingga dapat

menemukan elemen yang mengalami *error* untuk menormalkan kembali kondisi *error* pada plant.

Pada perancangan sistem monitoring ini menggunakan Atmega32. Hal ini digunakan karena sesuai dengan kebutuhan pengguna dalam menggunakan fungsi-fungsi mikrokontroller tersebut yang berfungsi sebagai tempat pengidentifikasi *input* dan *output* data. Pada mikrokontroller Atmega ini diaktifkan tiap port pada *power*, ADC, dan digital. Sebagai aktivasi mikrokontroller Atmega32 ini mendapatkan sumber tegangan 5 volt. Untuk koneksi dari tiap pin terhadap instrumen digunakan kabel. Dapat dilihat pada gambar berikut.



Gambar 3.3 Konfigurasi Pin ATMega32

Pada gambar 3.3 merupakan gambar konfigurasi Pin ATMega 32. Untuk port yang digunakan pada ATMega hanya *port B* dan *D*. Dimana *port B* yang digunakan ialah pada Pin B4, Pin B6, Pin dan B7 sebagai jalur komunikasi dengan *Heat Exchanger*. Sedangkan Port D yang digunakan ialah Pin D0, dan Pin D1. Dimana pin D1 sebagai Tx dan Pin D0 sebagai Rx artinya kedua pin ini merupakan jalur komunikasi antar mikrokontroler.

3.4.2 Penggerjaan Design Tampilan LCD

Perihal *User Interface*, LCD digunakan untuk menampilkan data yang telah terbaca oleh sensor sehingga dapat ditampilkan data berupa data *digital* yang menunjukkan nilai *variable* yang diukur. Sebelum bisa tampil melalui LCD, hasil pembacaan sensor yang awalnya berupa data ADC diolah terlebih dahulu oleh sistem pengendalian dengan sedemikian rupa sehingga langsung dapat disimpan dan disampaikan langsung oleh mikrokontroler ATMega32 yang mana, mikrokontroler disini berfungsi sebagai terminal penghubung antar panel.

Untuk *design* tampilan, dipilih gambar thermometer sebagai indikator suhu pada kedua proses yakni *heat exchanger*, Gradiasi warna sebagai indikator untuk melihat tingkat kekeruhan, dan gambar gelas sebagai indikator *counter*. Berikut sub-program yang bertugas menggambar ketiga gambar yang mewakili variabel yang diukur:

1. Thermometer sebagai indikator proses pemasakan

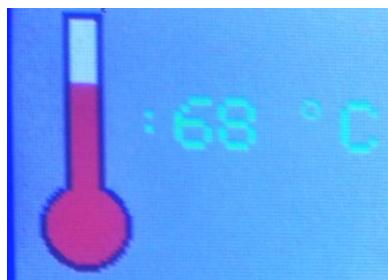
```

1. tft.setTextColor(TFT_GREEN);
2. tft.drawRect(16, 8, 11, 50,TFT_BLACK);
3. tft.drawRect(17, 9, 9, 48,TFT_BLACK);
4. tft.fillCircle(21, 66, 12, TFT_BLACK);
5. tft.fillCircle(21, 66, 10, TFT_RED);
6. tft.fillRect(18, 12, 7, 50,TFT_RED);
7. tft.fillRect(18, 10, 7, 45-
a/2.5,TFT_WHITE);
8. tft.print(":"); tft.print(a);
9. tft.setCursor(80, 30);tft.print(" C");
10.tft.drawCircle(83, 32, 2, TFT_GREEN);

```

Sub program diatas merupakan perintah untuk menggambar *layer-layer* yang akan membentuk thermometer. Dapat dijelaskan pada baris pertama ialah perintah untuk warna font daripada nilai hasil pembacaan sensor suhu. Baris kedua merupakan perintah untuk menggambar kotak yang akan membentuk tangkai daripada thermometer dengan ukuran lebar 11

pixels, tinggi 50 *pixels* dan kotak ini letak titik koordinat paling atas sebalah kiri berada pada titik 16,8 dan diatur agar berwarna hitam. Sama halnya dengan baris kedua, yakni menggambar kotak namun ditaris ketiga ini untuk bingkai dari kotak pada baris kedua. Hal tersebut dapat kita ketahui melalui titik koordinat terluar yang lebih besar daripada baris kedua yakni 17,9 sedangkan ukuran lebar dan tingginya yakni 9 *pixels* dan 50 *pixels*. Untuk baris keempat dan kelima sama persis perintah yang dilakukan dengan baris kedua dan ketiga hanya saja bentuk dari baris keempat dan kelima ini adalah lingkaran. Hal tersebut dapat kita ketahui dari perintah *tft.fillCircle*. Untuk baris keenam, merupakan perintah untuk menggambar kotak berwarna merah yang berperan sebagai indikator perubahan suhu pada termometer. Pada baris ketujuh merupakan perintah untuk menggambar kotak berwarna putih yang nantinya akan berubah naik turun agar kotak berwarna merah seakan yang berubah seperti awamnya indikator pada termometer. Baris kedelapan adalah perintah untuk menuliskan tanda ":" pada LCD, sama halnya baris kedelapan untuk baris kesembilan merupakan perintah untuk menuliskan huruf "C". Terakhir baris kesepuluh ialah perintah untuk menggambar lingkaran, dalam hal ini adalah karakter "°" dan diatur agar berwarna hijau sama dengan angka dan huruf yang mengindikasikan nilai suhu yang diukur. Berikut merupakan visualisasi daripada perancangan *display* suhu pada proses pemasakan teh.



Gambar 3.4 Design Tampilan Suhu Pada Proses Pemasakan

Pada gambar 3.4 merupakan hasil design daripada display suhu pada proses pemasakan teh di tank pertama. Pada gambar terlihat suhu sedang pada posisi baca 68°C yang artinya mendekati setpoint yang telah ditentukan yakni 70°C

2. Thermometer sebagai indikator suhu pada *heat exchanger*

```
1. tft.setTextColor(TFT_BLUE);
2. tft.drawRect(16+gx, 8, 11, 50, TFT_BLACK);
3. tft.drawRect(17+gx, 9, 9, 48, TFT_BLACK);
4. tft.fillCircle(21+gx, 66, 12, TFT_BLACK);
5. tft.fillCircle(21+gx, 66, 10, TFT_RED);
6. tft.fillRect(18+gx, 12, 7, 50, TFT_RED);
7. tft.fillRect(40+gx, 30 , 46, 30
    ,tft.color565(220,220,220));
8. tft.print(":");
9. tft.setCursor(80+gx, 30);tft.print(" C");
10. tft.drawCircle(83+gx, 32, 2, TFT_BLUE);
```

Sub program diatas merupakan perintah untuk menggambar layer-layer yang akan membentuk thermometer. Dapat dijelaskan pada baris pertama ialah perintah untuk warna font daripada nilai hasil pembacaan sensor suhu yakni warna biru. Baris kedua merupakan perintah untuk menggambar kotak yang akan membentuk tangkai daripada thermometer dengan ukuran lebar 11 pixels, tinggi 50 pixels dan kotak ini letak titik koordinat paling atas sebalah kiri berada pada titik $16+gx$, 8 dan diatur agar berwarna hitam. Sama halnya dengan baris kedua, yakni menggambar kotak namun dibaris ketiga ini untuk bingkai dari kotak pada baris kedua. Hal tersebut dapat kita ketahui melalui titik koordinat terluar yang lebih besar daripada baris kedua yakni $17+gx$, 9 sedangkan ukuran lebar dan tingginya yakni 9 pixels dan 48 pixels . Untuk baris keempat dan kelima sama persis perintah yang dilakukan dengan baris kedua dan ketiga hanya saja bentuk dari baris keempat dan kelima ini adalah lingkaran. Hal tersebut

dapat kita ketahui dari perintah *tft.fillCircle*. Untuk baris keenam, merupakan perintah untuk menggambar kotak berwarna merah yang berperan sebagai indikator perubahan suhu pada termometer. Pada baris ketujuh merupakan perintah untuk menggambar kotak berwarna putih yang nantinya akan berubah naik turun agar kotak berwarna merah seakan yang berubah seperti awamnya indikator pada termometer. Baris kedelapan adalah perintah untuk menuliskan tanda “.” pada LCD, sama halnya baris kedelapan untuk baris kesembilan merupakan perintah untuk menuliskan huruf “C”. Terakhir baris kesepuluh ialah perintah untuk menggambar lingkaran, dalam hal ini adalah karakter “o” dan diatur agar berwarna hijau sama dengan angka dan huruf yang mengindikasikan nilai suhu yang diukur. Berikut merupakan visualisasi daripada perancangan *display* suhu pada *heat exchanger*



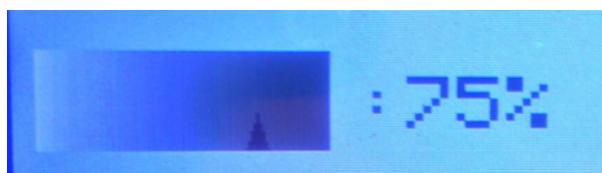
Gambar 3.5 Design Tampilan Suhu Pada Heat Exchanger

Pada gambar 3.5 merupakan hasil design daripada display suhu pada proses pendinginan di *heat exchanger* setelah proses pemasakan di tank pertama. Pada gambar terlihat suhu sedang pada posisi baca 26°C yang artinya mendekati setpoint yang telah ditentukan yakni $28^{\circ}\text{C}-30^{\circ}\text{C}$.

3. Gradasi Warna sebagai indikator tingkat kepekatan teh

```
1. for(int n=10 ; n<=110 ;n+=1)
2. tft.fillTriangle(10+a, 135, 6+a,
   149, 14+a, 149 , TFT_BLACK);
3. tft.setTextSize(3);
4. tft.setCursor(48, 120);
   tft.print(":"); tft.print(a);
   tft.print("%");
```

Terdapat 4 baris utama untuk merancang display tingkat kepekatan warna teh. Baris pertama ialah perintah untuk menggambar perpindahan warna pada gradasi warna dengan range 110 sebagai range maksimal dalam pembacaan nilai kepekatan warna. Sedangkan baris kedua ialah untuk menggambar bentuk segetiga sebagai indikator letak warna kepekatan berada pada tingkat berapa dan telah diatur warnanya yakni warna hitam. Baris ketiga menjelaskan mengenai penulisan nilai hasil pembacaan sensor kepekatan dan nilai 3 mengindikasikan angka yang akan keluar pada tampilan sejumlah 3 digit. Baris keempat dan atau terakhir ini merupakan perintah untuk menuliskan karakter seperti “:”, dan karakter “%” untuk melengkapi nilai hasil pembacaan telah mencapai tingkat kepekatan yang telah ditentukan.



Gambar 3.6 Design Tampilan Kekeruhan

Pada gambar 3.6 merupakan hasil design daripada *display* tingkat kepekatan warna teh pada proses pemasakan teh di *tank*

pertama. Pada gambar terlihat kepekatan sedang pada posisi baca 75% yang artinya nilai kepekatan warna sudah mencapai hampir maksimal sehingga harus melakukan *filtering* dengan mengganti dengan kantong teh yang baru.

4. Counter Glass

```
1. void out_gelas(int a)
2. tft.fillRect(5,170 ,304, 66,
   tft.color565(220,220,220));
3. for(int n=0;n<a;n++)
4. tft.drawRoundRect(5+n*45, 180, 35,
   10,7, TFT_BLACK);
   tft.drawRoundRect(6+n*45, 181, 33, 8,4,
   TFT_BLACK);
5. tft.drawRoundRect(11+n*45, 230, 23,
   6,3, TFT_BLACK);
   tft.drawRoundRect(12+n*45, 231, 21,
   4,1, TFT_BLACK);
6. tft.drawLine(5+n*45,183,11+n*45,232,
   TFT_BLACK)
7. tft.drawLine(39+n*45,183,33+n*45,232,
   TFT_BLACK);
   tft.drawLine(38+n*45,184,33+n*45,230,
   TFT_BLACK);
8. tft.fillRect(270,200 ,45, 35
   ,tft.color565(220,220,220));
9. tft.setTextSize(2); tft.setCursor(270,
  200); tft.print(":"); tft.print(a);
```

Pada sub program diatas merupakan proses pembuatan program gambar gelas, dapat dilihat pada baris pertama yakni perintah utama untuk proses penggambaran gelas. Pada baris kedua ialah untuk menghapus gelas yang telah tampak untuk ke perhitungan selanjutnya dan baris ketiga ialah penggambaran ulang setelah terjadi penghapusan. Untuk baris ke empat dan kelima merupakan proses pembuatan gelas yang tersusun atas *layer-layer*

berbentuk kotak dan bulat, hal tersebut dapat kita ketahui melalui isi dari sub program yang menyebutkan “*tft.drawRoundRect*”. Selanjutnya baris keenam dan ketujuh ialah perintah sub-program untuk menggambar sebuah garis agar mempertebal gambar sehingga ketika dilakukan penghapusan dan penggambaran ulang , gambar gelas akan tetap terlihat jelas dan memiliki kualitas bagus. Untuk baris kedelapan merupakan sub program yang digunakan untuk menghapus nilai sebelumnya, fungsi ini digunakan dalam masa transisi antara gelas pertama dan gelas kedua begitu hingga mencapai gelas keenam. Terakhir ialah baris kesembilan yang memiliki tugas untuk menuliskan nilai yang barusaja dihitung dengan mencetak karakter “:” dan (a) adalah nilai data hasil pembacaan sensor.

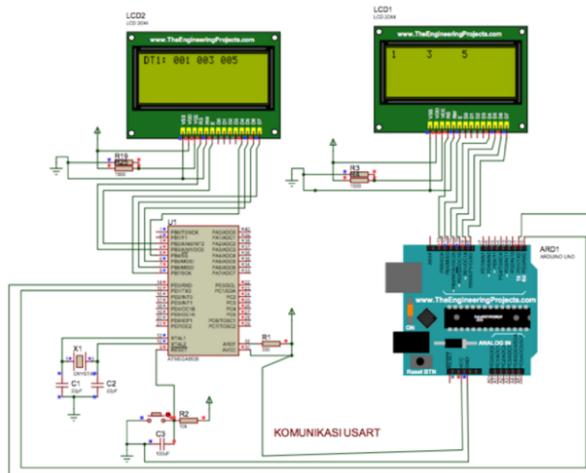


Gambar 3.7 Design Tampilan *Counter Glass*

Pada gambar 3.7 merupakan hasil design daripada display *counter glass* pada proses pengepakan teh tawar. Pada gambar terlihat pembacaan *counter glass* oleh sensor *proximity* sedang pada posisi baca 2 gelas yang artinya sensor *proximity* telah berhasil membaca gelas sejumlah yang ditampilkan, dan pada gambar 3.7 merupakan gambar yang menunjukkan *setpoint* maksimal yakni 6 gelas.

Setelah dilakukannya *design* terhadap *display*, selanjutnya dilakukan perancangan *hardware* melalui simulasi. Dalam perancangan perangkat lunak digunakan simulasi percobaan menggunakan *Proteus 7 Professional* sebagai *software* simulator dan *CodeVisionAVR* sebagai *software* IDE program dari ATmega 32. *Proteus 7 Professional* digunakan sebagai perangkat simulator

dikarenakan dalam aplikasi tersebut terdapat berbagai elemen seperti ATMega 32, LCD, dan sebagainya. Sehingga, pelaksanaan simulasi pada perangkat lunak bisa berlangsung sesuai dengan kebutuhan pada perangkat kerasnya. Sedangkan *CodeVisionAVR* digunakan sebagai aplikasi untuk mengkoding program dikarenakan aplikasi tersebut mudah digunakan dan tidak memakan memori banyak, sehingga bagi komputer yang memiliki spesifikasi memori rendah pun masih bisa menggunakannya, dan dari sekian banyak aplikasi IDE untuk membuat program pada AVR, *CodeVisionAVR* mempunyai beberapa fitur yang bisa mempermudah pengguna untuk mengkoding sebuah program. Berikut hasil dari simulasi pada perancangan perangkat lunak



Gambar 3.8 Wiring Diagram Design Hardware

Pada Gambar 3.8 di atas adalah simulasi pertama yang dilakukan dengan menggunakan LCD Karakter 16x2 dikarenakan pada software proteus tidak terdapat jenis LCD TFT yang akan diaplikasikan langsung pada plan. LCD ini digunakan untuk validasi data yang dibaca dengan potensiometer, serta ATMega 32 sebagai kontroler. Dalam simulasi tersebut, digunakan 3

potensiometer sesuai dengan variabel yang telah ditetapkan yaitu *temperature1*, *temperature2*, dan *counter*.

3.4.3 Penggerjaan *Design* Penyimpanan Data

Dalam proses penyimpanan data, digunakan fasilitas pada LCD TFT yang memiliki slot sd card sebagai media *data logger*. Sehingga, pengguna dapat dengan mudah melakukan proses penyimpanan data yang telah didapat. Berikut merupakan *sub-program* yang bertugas membaca dan menampilkan data yang telah dikirim oleh sistem pengendalian

```

1. void update_history() {
2. SD.remove("laporan.txt");
3. myFile = SD.open("laporan.txt",
FILE_WRITE);
4. if (myFile)
5. {
6. Serial.print("Writingtofile...");
Serial.println("laporan.txt");
7. myFile.print(char(line_A));myFile.print(
char(line_B));
myFile.print(char(line_C));
myFile.print(char(line_D));
8. myFile.print(page_n[0]);myFile.print(pag
e_n[1]); myFile.print(page_n[2]);
myFile.print(page_n[2]);
9. myFile.close();
10.Serial.println("Done.");
11. }

```

Pada sub-program diatas merupakan proses perancangan *system* penyimpanan ataupun *data logger* yang akan digunakan pada *system* monitoring ini. Terdapat 10 baris kumpulan fungsi yang masing-masing memiliki peran yang berbeda. Pada baris pertama merupakan inisialisasi proses penyimpanan yang akan dilakukan. Update history yang dimaksud ialah sistem pembacaan yang dilakukan sensor selalu mengalami perubahan sehingga

terjadi *system refresh* atau pembaruan pada data yang dibaca. Pada baris kedua dan ketiga merupakan proses pemasangan dan proses pembacaan SD *Card* pada LCD. Pada baris keempat hingga keenam merupakan proses penulisan data yang akan disimpan pada *data logger*. Pada baris ketujuh merupakan *sub-program* yang akan menuliskan huruf “A” yang datanya akan dituliskan berupa *line* atau baris sebanyak data yang akan dibaca. Huruf “A” disini merupakan *indicator* suhu pada proses pemasakan, sedangkan huruf “B” merupakan *indicator* kepekatan pada proses pemasakan, untuk huruf “C” ialah *indicator counter glass* pada proses pengepakan, dan yang terakhir adalah huruf “D” merupakan *indicator* suhu pada proses pendinginan pada *heat exchanger*. Untuk baris kedelapan merupakan design halaman jika data yang telah disimpan melabih satu halaman pertama dan akan berlanjut ke halaman selanjutnya jika dilakukan *running* alat secara berkala. Baris kesembilan dan kesepuluh merupakan perintah untuk menutup file yang telah terbaca.

```
1. void simpan(char x,char n,int l,int
   data)
2. char namaf[]="Xn.txt";
3. namaf[0]=x; namaf[1]=n;
4. myFile = SD.open(namaf, FILE_WRITE);
5. if (myFile)
6. {
7. Serial.print("Writing to
   file...");Serial.println(namaf);
8. myFile.print(l);myFile.print(":");myFil
   e.println(data);
9. myFile.close();
10. Serial.println("Done.");
```

```

11.      }
12.else {
13.Serial.print("error opening");
Serial.print(nama);Serial.println("...");
```

Pada sub program diatas merupakan sub-program yang bertugas untuk menyimpan data yang telah diterima dan langsung menuliskannya pada tampilan dengan format *txt*. Hal tersebut dapat diketahui dengan melihat baris pertama dan kedua pada sub-program. Sama halnya seperti yang dijelaskan, proses penulisan terdapat pada baris keempat dan kelima. Pada baris ketujuh merupakan fungsi untuk mengirimkan data dan baris kedelapan merupakan fungsi penulisannya.



Gambar 3.9 Design DataLogger

Pada gambar 3.9 diatas merupakan salah satu *design* tampilan *datalogger* untuk *variable* suhu proses pemasakan. Dapat dilihat pada gambar terdapat tulisan *display* di pojok kanan atas mewakili perintah untuk kembali ke display awal yakni *display* yang menunjukkan *variable* yang dimonitoring, halaman 1 di pojok kanan bawah mewakili perintah bahwa pengguna sedang berada pada halaman tersebut dan tulisan *file* di pojok kiri atas yang mewakili fungsi daripada variable yang diukur.

Halaman ini sengaja dikosongkan

BAB IV

HASIL DATA DAN

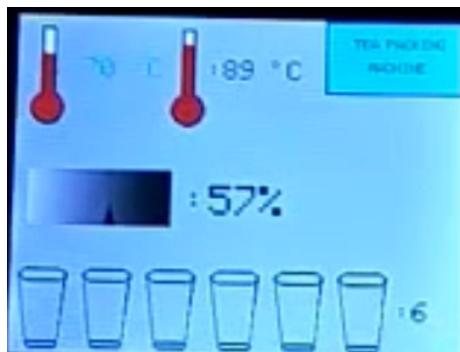
PEMBAHASAN

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil Rancang Bangun Alat

Berikut ini adalah hasil perancangan sistem monitoring suhu dan jumlah gelas pada mesin produksi teh tawar kemasan yang terintegrasi dengan *Human Machine Interface* (HMI)



Gambar 4.1 Hasil Tampilan Panel Monitoring

Pada gambar 4.1 merupakan hasil perancangan display *human machine interface* (HMI) pada mesin produksi teh tawar kemasan. Dapat dilihat pada gambar, terdapat dua buah gambar *thermometer* yang mewakili pembacaan suhu sebelah kiri dengan warna *text* data hijau dan sebelah kanan dengan warna *text* data biru berturut-turut ialah merepresentasikan pembacaan sensor pada proses pemasakan dan *cooling* pada *heat exchanger*. Sedangkan terdapat sebuah persegipanjang yang merepresentasikan tingkat kepekatan warna teh pada proses pemasakan. Selanjutnya gambar paling bawah ialah terdapat gambar gelas yang merepresentasikan *counter glass* yang akan dilakukan oleh sensor *proximity*.

4.1.1. Validasi *Display*

Untuk mengetahui apakah data yang ditampilkan sesuai atau tidak dengan *setpoint* yang telah ditentukan, maka perlu dilakukan

uji validasi pada variabel yang diukur. Uji validasi tersebut dapat dilakukan dengan dua cara yakni dengan menganalisa kesesuaian data yang keluar pada setiap variabel terwakili yang ditampilkan LCD pada panel monitoring, dan juga menyesuaikan dengan hasil pembacaan yang ditampilkan oleh panel sistem pengendalian.

Berikut salah satu hasil validasi suhu yang telah dilakukan dengan menganalisa kesesuaian data yang keluar pada setiap variabel terwakili yang ditampilkan LCD pada panel monitoring.

Tabel 4.1 Validasi Tampilan LCD pada variable suhu HE

Suhu yang dibaca	Hasil Tampilan	Keterangan
25°C		Proses pembacaan suhu pada proses <i>cooling</i> di <i>heat exchanger</i> pada suhu mencapai 25°C
26°C		Proses pembacaan suhu pada proses <i>cooling</i> di <i>heat exchanger</i> pada suhu mencapai 26°C

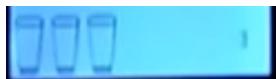
Tabel 4.1 diatas menjelaskan tampilan pada proses *cooling* yang terjadi di *heat exchanger*, pada tabel juga dapat dijelaskan mengenai perbedaan tampilan yang dihasilkan pada suhu 25°C dan suhu 26°C. Dari tabel diatas juga dapat dilihat bahwa *indicator* penunjuk kenaikan suhu mengalami perubahan, sehingga dapat dikatakan bahwa secara *system* program telah sesuai.

Tabel 4.2 Validasi Tampilan LCD pada variable suhu Pemasakan

Suhu yang dibaca	Hasil Tampilan	Keterangan
58°C		Proses pembacaan suhu pada proses pemasakan di tank pemasakan pada suhu mencapai 58°C
60°C		Proses pembacaan suhu pada proses pemasakan di tank pemasakan pada suhu mencapai 60°C
70°C		Proses pembacaan suhu pada proses pemasakan di tank pemasakan pada suhu mencapai 70°C

Tabel 4.2 diatas menjelaskan tampilan pada proses pemasakan teh yang terjadi di tank pemasakan, pada tabel juga dapat dijelaskan mengenai perbedaan tampilan yang dihasilkan pada suhu 58°C, 60°C dan *setpoint* yakni suhu 70°C. Dari tabel diatas juga dapat dilihat bahwa *indicator* penunjuk kenaikan suhu mengalami perubahan, jadi semakin tinggi suhu yang dibaca sebanding dengan semakin tinggi *indicator* kenaikan suhu tampil pada *display* sehingga dapat dikatakan bahwa secara *system* program telah sesuai.

Tabel 4.3 Validasi tampilan *counter glass*

Jumlah gelas yang dibaca	Hasil Tampilan	Keterangan
1		Hasil tampilan ketika gelas pertama telah melewati sensor <i>proximity</i>
2		Hasil tampilan ketika gelas kedua telah melewati sensor <i>proximity</i>
3		Hasil tampilan ketika gelas ketiga telah melewati sensor <i>proximity</i>
4		Hasil tampilan ketika gelas keempat telah melewati sensor <i>proximity</i>
5		Hasil tampilan ketika gelas kelima telah melewati sensor <i>proximity</i>
6		Hasil tampilan ketika gelas keenam telah melewati sensor <i>proximity</i>

Tabel diatas merupakan hasil tampilan daripada proses counter glass. Seperti yang dijelaskan pada bab sebelumnya bahwa dalam menampilkan hasil pembacaan dari gelas pertama ke gelas berikutnya dalam artian tergantung gelas yang telah melewati sensor proximity ialah dengan cara menampilkan dan menghapus data awal. Jadi ketika gelas pertama selesai dihitung, gambar gelas pada hitungan pertama akan dihapus dan ketika menuju ke perhitungan gelas kedua sistem akan langsung menghitung dan menampilkan dua gambar gelas sekaligus. Begitupun pada gelas-gelas berikutnya hingga mencapai *setpoint* yang telah ditentukan yakni sejumlah enam gelas.

Cara kedua untuk melakukan validasi data yang ditampilkan oleh monitoring ialah dengan cara menyesuaikan dengan hasil pembacaan yang ditampilkan oleh panel sistem pengendalian. Berikut merupakan perbandingan tampilan data pada panel monitoring dan panel pengendalian yang disajikan pada tabel dibawah ini

Tabel 4.4 Validasi tampilan lcd dengan panel *system* pengendalian

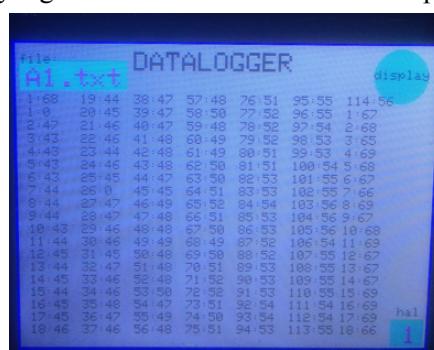
Display LCD TFT	Display pada panel pengendalian
	SUHU = 59, 3 °C LEVEL = HIGH KEPEKATAN=T. PEKAT PEMASAKAN TEH - 11

Pada Tabel 4.3 di atas adalah hasil uji validasi yang telah dilakukan dengan membandingkan kesesuaian antara panel pengendalian dan panel monitoring. Terlihat bahwa pada tampilan *variable* suhu mengalami sedikit perbedaan dikarenakan tipe data

yang digunakan ketika memrogram pada system pengendalian menggunakan jenis atau tipe data *float* yang memungkinkan pembacaan data dengan nilai desimal, namun untuk komunikasi data serial dalam menggunakan jenis data tersebut justru akan memungkinkan banyaknya terjadi kesalahan pengiriman data sehingga ketika akan dikirim, maka data yang dikirim diubah jenisnya menjadi data integer yang bertujuan untuk meminimalisir terjadinya kesalahan dalam pembacaan data dan menghindari terjadinya data yang hilang.

4.1.2 Validasi Proses Penyimpanan Data

Untuk mengetahui apakah data yang telah dibaca tersebut telah terekam oleh *system* penyimpanan, maka perlu dilakukan validasi pada system dengan tujuan memastikan bahwa data yang ditampilkan pada LCD merupakan data yang sesuai dengan data yang telah diolah ketika berada pada system pengendalian. Sama halnya dengan ketika melakukan validasi tampilan yakni dilakukan pengecekan pada tampilan *data logger* yang terdapat di LCD. Berikut data yang berhasil terekam dan tersimpan pada *data logger*.



Gambar 4.2 Hasil penyimpanan data suhu pada *datalogger*

Gambar 4.2 merupakan tampilan daripada hasil penyimpanan data untuk *variable* suhu pada proses pemasakan teh. Terlihat pada gambar sering terjadi pengulangan pembacaan daripada nilai baca suhu. Hal tersebut dikarenakan pada program proses pengendalian

tidak diatur untuk keluaran data agar menerima data sekali saja, dalam artian jika sensor membaca data suhu yang sama selang beberapa detik maka yang akan ditampilkan sesuai dengan hasil yang sama seharusnya bisa diatur sedemikian rupa agar hanya mengirimkan data sekali saja jika data tersebut menunjukan nilai yang sama.

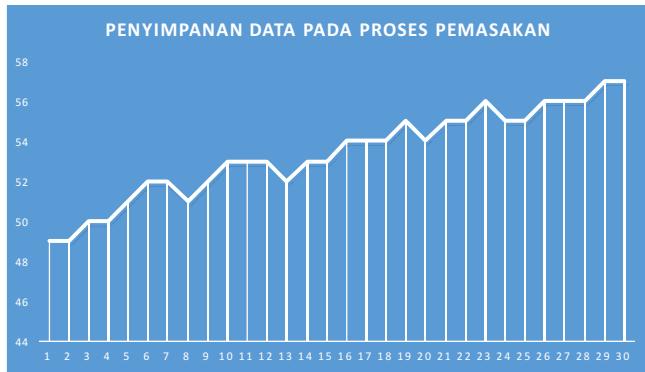
Berikut data hasil penyimpanan data suhu yang telah dikonversi dari format .txt ke excel

Tabel 4.5 Penyimpanan data suhu pemasakan

Data ke-n	Penyimpanan data pada proses pemasakan
1	49
2	49
3	50
4	50
5	51
6	52
7	52
8	51
9	52
10	53
11	53
12	53
13	52
14	53
15	53
16	54
17	54
18	54
19	55
20	54
21	55
22	55
23	56
24	55
25	55
26	56
27	56
28	56
29	57
30	57

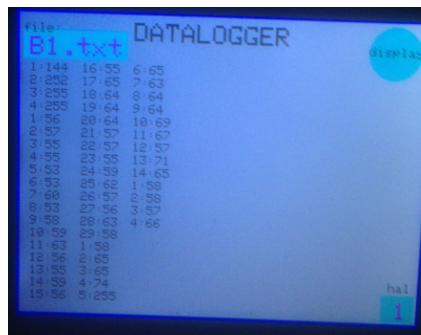
Pada tabel 4.5 merupakan 30 data yang diambil pada proses penyimpanan data yang akan disimpan dalam *datalogger*. Secara kasat mata, data diatas mengalami kenaikan yang stabil

dalam artian tidak terlalu fluktuatif. Sehingga bisa juga dikatakan bahwa pembacaan daripada sensor telah mendekati setpoint yang telah ditentukan. Lebih jelasnya dapat dilihat pada grafik berikut



Gambar 4.3 Grafik penyimpanan data suhu pemasakan teh

Pada gambar diatas dapat kita lihat bahwa dalam 30 sampel data yang diambil menghasilkan pembacaan yang naik turun. Data yang diambil dari ketiga puluh data ialah pada *range* suhu 49-57 mengalami naik turun meskipun tidak terlalu signifikan namun juga tidak bisa dibilang linear sempurna. Namun dari *range* yang ada, data nilai penyimpanan suhu memiliki kecenderungan naik.



Gambar 4.4 Hasil penyimpanan kekeruhan pada *datalogger*

Gambar 4.3 merupakan tampilan daripada hasil penyimpanan data untuk *variable kekeruhan* pada proses pemasakan teh. Seperti yang dijelaskan pada bab sebelumnya untuk system kepekatan teh, range pembacaan akan dibagi menjadi 3 yakni tidak pekat, pekat, dan sangat pekat. Dalam artian, dari 3 range tersebut jika diimplementasikan dalam bentuk persen maka akan menjadi 0-33% untuk merepresentasikan nilai tidak pekat, 34-67% merepresentasikan nilai pekat dan 68-100% untuk nilai sangat pekat. Namun, data ini tidak seberapa dibutuhkan dikarenakan sensor yang digunakan mengalami kerusakan.

1	2	3	4
1.1	20.41	39.1	12.6
2.1	21.1	40.2	
3.0	22.2	1.1	
4.3	23.1	4.8	
5.4	24.2	3.3	
6.5	25.1	4.1	
7.6	26.1	5.65	
8.1	27.1	6.1	
9.1	28.2	1.1	
10.1	29.3	2.2	
11.2	30.1	3.3	
12.3	31.2	4.4	
13.1	32.1	5.1	
14.2	33.2	6.1	
15.1	34.3	7.1	
16.2	35.4	8.2	
17.3	36.5	9.5	
18.1	37.1	10.4	
19.1	38.3	11.5	

Gambar 4.5 Hasil penyimpanan data *counter*

Gambar 4.5 merupakan tampilan daripada hasil penyimpanan data untuk *variable counter glass* pada proses akhir yakni pengepakan. *Setpoint* daripada proses ini ialah dapat menghitung gelas sebanyak 6 gelas yang mana pada setiap 3 gelasnya akan dilakukan *delay* selama beberapa detik karena ada nada proses pendorongan gelas kedalam kardus. Namun, dapat dilihat pada gambar telah terjadi beberapa kali *error* yang mengakibatkan sensor membaca nilai yang melampaui *setpoint*. Hal tersebut bisa terjadi dikarenakan terdapat *bug* pada program sehingga harus menata ulang program agar bisa bekerja sesuai dengan *setpoint* yang telah ditentukan.

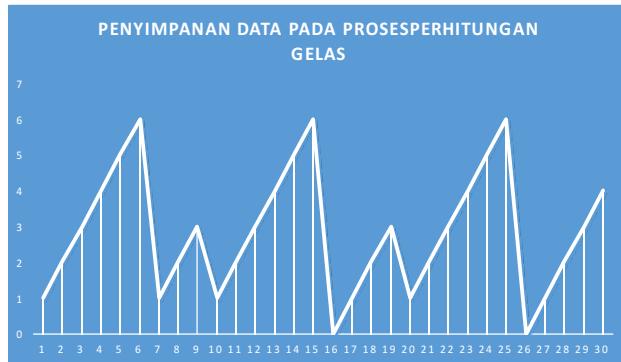
Berikut data hasil penyimpanan data suhu yang telah dikonversi dari format .txt ke excel.

Tabel 4.6 Penyimpanan data perhitungan gelas

Data ke-n	Simpanan data pada proses perhitungan gelas
1	1
2	2
3	3
4	4
5	5
6	6
7	1
8	2
9	3
10	1
11	2
12	3
13	4
14	5
15	6
16	0
17	1
18	2
19	3
20	1
21	2
22	3
23	4
24	5
25	6
26	0
27	1
28	2
29	3
30	4

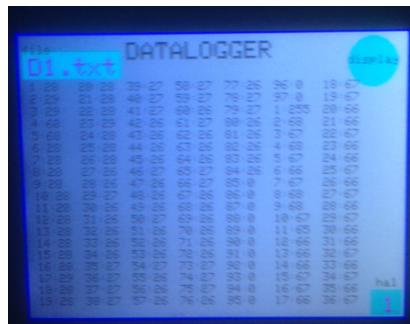
Pada tabel 4.6 dapat diketahui bahwa proses perhitungan gelas sempat mengalami error dikarenakan sensor yang cukup sensitive jika mendekripsi adanya benda yang menutupinya dapat juga karena berhentinya gelas yang menutupi sensor berhenti terlalu lama sehingga sensor menghitungnya termasuk *double counting*. Berikut juga ditampilkan hasil analisis data penyimpanan dalam bentuk grafik untuk *variable counter glass*. Terdapat nilai 0

pada tabel dikarenakan jika pembacaan telah memenuhi *setpoint*, maka *counter* akan mengalami reset pada proses awal.



Gambar 4.6 Grafik data *counter* pada *data logger*

Gambar 4.6 menjelaskan bahwa hasil pembacaan yang disimpan sudah mendekati dengan *setpoint* yang telah ditentukan, bahkan dalam *running* alat sebenarnya yang telah dilakukan sudah berhasil mencapai *setpoint* yang ditentukan yaitu menghitung sebanyak 6 gelas. Namun, masih terdapat beberapa titik *error* karena mengalami *reset* yang disebabkan *human error* ketika melukau pengujian alat.



Gambar 4.7 Hasil penyimpanan data suhu

Gambar 4.7 merupakan tampilan daripada hasil penyimpanan data untuk *variable* suhu pada proses pendinginan teh oleh *heat*

exchanger. Terlihat pada gambar sering terjadi pengulangan pembacaan daripada nilai baca suhu. Hal tersebut dikarenakan pada program proses pengendalian tidak diatur untuk keluaran data agar menerima data sekali saja, dalam artian jika sensor membaca data suhu yang sama selang beberapa detik maka yang akan ditampilkan sesuai dengan hasil yang sama seharusnya bisa diatur sedemikian rupa agar hanya mengirimkan data sekali saja jika data tersebut menunjukkan nilai yang sama.

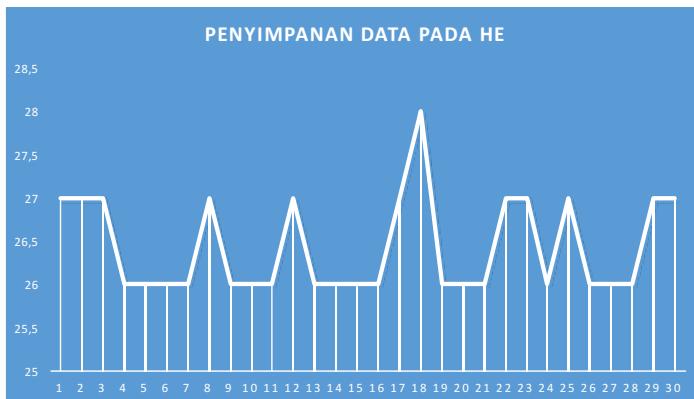
Berikut data hasil penyimpanan data suhu yang telah dikonversi dari format .txt ke excel.

Tabel 4.7 Penyimpanan data suhu pada *heat exchanger*

Data ke-n	Penyimpanan data pada HE
1	27
2	27
3	27
4	26
5	26
6	26
7	26
8	27
9	26
10	26
11	26
12	27
13	26
14	26
15	26
16	26
17	27
18	28
19	26
20	26
21	26
22	27
23	27
24	26
25	27
26	26
27	26
28	26
29	27
30	27

Pada tabel 4.7 merupakan 30 data yang diambil pada proses penyimpanan data suhu pada proses pendinginan di *heat exchanger* yang akan disimpan dalam *datalogger*. Secara kasat mata, data diatas mengalami kenaikan yang cukup stabil dalam

artian tidak terlalu fluktuatif. Sehingga bisa juga dikatakan bahwa pembacaan daripada sensor telah mendekati *setpoint* yang telah ditentukan. Lebih jelasnya dapat dilihat pada grafik berikut



Gambar 4.6 Grafik data suhu HE pada *data logger*

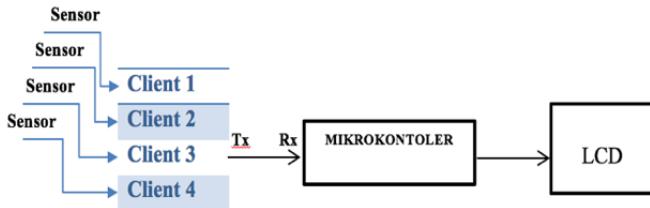
Pada gambar 4.6 diatas dapat kita lihat bahwa dalam 30 sampel data yang diambil menghasilkan pembacaan yang naik turun. Data yang diambil dari ketiga puluh data ialah pada *range* suhu 26°C-28°C seperti pada gambar data suhu mengalami naik turun meskipun tidak terlalu signifikan namun juga tidak bisa dibilang linear sempurna. Namun dari *range* yang ada, data nilai penyimpanan suhu telah sesuai dengan pembacaan ketika menggunakan *thermometer digital*.

4.2. Komunikasi Data

4.2.1 Penerimaan data dari sensor ke *display*

Data yang diterima oleh sensor ialah berupa sinyal analog, ketika akan dikirimkan kepada mikrokontroler maka harus dikonversikan terlebih dahulu menggunakan *Analog Digital Converter* (ADC) agar bisa terbaca oleh mikrokontroler. Untuk suhu misalnya, dikarenakan output suhu berupa tegangan sehingga membutuhkan ADC berupa *max6675* agar bisa terbaca oleh

mikrokontroler. Selanjutnya, mikrokontroler akan mengolah data dari sensor dan akan ditampilkan serta disimpan pada LCD TFT.



Gambar 4.8 Diagram Blok Sistem Komunikasi Data^[14]

Pada gambar 4.6 dapat dijelaskan bahwa maksud daripada *client* disini ialah pihak yang menerima hasil pembacaan sensor. Dalam plan ini, terdapat 4 *client* yang akan diproses datanya agar bisa ditampilkan pada *display* dan disimpan pada *datalogger*. Keempat client tersebut ialah suhu proses pemasakan, tingkat kepekatan warna teh, *counter glass*, dan suhu pada proses pendinginan di *heat exchanger*. Dari keempat *client* tersebut, akan terhubung dengan 1 mikrokontroler master yakni mikrokontroler ATMega 32 dengan cara menyambungkan pin Tx dan Rx yang ada pada masing-masing mikrokontroler yang digunakan.

4.2.2 Jenis Komunikasi Data

Pada system monitoring mesin produksi teh kemasan ini menggunakan komunasi *Serial Peripheral Interface* (SPI) agar mikrokontroler dapat berkomunikasi dengan baik. SPI merupakan salah satu metode pengiriman data dari suatu *device* ke *device* lainnya. Metode ini merupakan metode yang bekerja pada metode *full duplex* dan merupakan standar sinkronasi *serial data*. Komunikasi serial yang dimaksud disini adalah dengan menyambungkan port Tx dan Rx antar mikrokontroler yang ada pada mesin produksi teh tawar kemasan. Pada SPI, *device* dibagi menjadi dua bagian *master* dan *slave* dengan *master* sebagai *device* yang menginisiasi pengiriman data. Dalam aplikasinya,

sebuah *master* dapat digunakan untuk mengatur pengiriman data dari atau ke beberapa *slave* sekaligus. Alasan mengapa menggunakan jenis komunikasi SPI adalah karena SPI dapat digunakan untuk pengaturan beberapa *peripheral* oleh suatu *master* (mikrokontroler atau PC). Berikut adalah beberapa *peripheral* yang dapat menggunakan komunikasi data SPI:

- Sensor: *temperature*, tekanan, ADC, *touchscreen*
- *Control device*: *audio codecs*, *digital potentiometers*, DAC
- Akses memori: *flash* and EEPROM
- LCD *display*
- Beberapa MMC atau SD *card*

Halaman ini sengaja dikosongkan

BAB V **PENUTUP**

BAB V PENUTUP

5.1. Kesimpulan

Berdasarkan hasil perancangan sistem monitoring pada *plant* teh tawar kemasan maka dapat didapatkan kesimpulan sebagai berikut :

1. Telah berhasil dirancang sistem monitoring pada *plant* teh tawar kemasan. Pada sistem monitoring *temperature* menggunakan alat ukur sensor termokopel baut tipe K dengan pengkondisian sinyal MAX6675 dan monitoring *counter* gelas dengan menggunakan sensor proximity yang keduanya telah dilakukan kalibrasi.
2. Validasi nilai pembacaan sensor dilakukan dengan 2 cara yakni membandingkan hasil baca antara panel pengendalian dan panel monitoring dan pada hasil daripada design tampilan yang terdapat pada LCD. Untuk variabel suhu terdapat perbedaan rata-rata sekitar $0,4^{\circ}\text{C}$ yang disebabkan tipe data yang digunakan oleh system pengendalian dan system monitoring berbeda.
3. Telah berhasil dirancang sistem monitoring *temperature* dan *counter* yang terintegrasi HMI menggunakan LCD TFT sebagai *interface*. Sistem monitoring pada *plant* teh tawar ini dilengkapi dengan hasil yang dapat di-*record* dan disimpan langsung pada slot sd card yang telah disediakan oleh TFT dengan memory penyimpanan sesuai dengan kapasitas memory pada sd card yakni 8GB.

5.2. Saran

Adapun saran yang diperlukan untuk keberlangsungan Tugas Akhir ini apabila ada orang lain yang ingin mengembangkan Tugas Akhir ini di kemudian hari:

1. Perlu dilakukannya pengambilan data terhadap hasil monitoring dengan minimal 10 kali pengambilan data guna menguji kesesuaian terhadap hasil pengukuran

pembacaan sensor yang ditampilkan pada panel pengendalian dan panel monitoring.

2. Peletakan/ *tapping point* pada instrument alat ukur perlu diperhatikan karena nilai rambat suatu element berbeda sehingga tidak memberikan nilai yang seharusnya.
3. Setiap kegiatan uji yang dilakukan perlu dilakukan dokumentasi agar setiap sistem yang berjalan dapat diketahui ketika terjadi sistem *error* secara tiba-tiba

DAFTAR PUSTAKA

DAFTAR PUSTAKA

1. Supriyadi, Mukh Yasir Nur. 2012. *Pengenalan Mikrokontroller AVR Atmega32* . Bandung : Laboratorium Hardware Program Study Teknik Komputer FMIPA UNPAD.
2. Yulistianto, Deni. 2013. *Pengertian CodeVision AVR* . Bandung
3. Hadimi, Supandi, & Rohermantti, A. (2006). Rancang Bangun Model Mesin Produksi Teh Kemasan. Jurnal Ilmiah Semesta Teknika Vol 9 No.1, 16-24
4. Hidayat, Rohmat. 2013. *Perancangan HMI pada Hitcut Machine dengan PLC Omron Sysmac CPIL*. Surabaya : Institut Teknologi Sepuluh Nopember
5. Marcus Zakaria, Teddy. 2003. *Pemrograman untuk Pemula : IDE dan Struktur Pemrograman*. Bandung
6. Supriyadi, Mukh Yasir Nur. 2012. Pengenalan Mikrokontroller AVR Atmega32 . Bandung : Laboratorium Hardware Program Study Teknik Komputer FMIPA UNPAD.
7. Engineering, Omega. 1999. *Transactions in Measurement and Control: Force-Related Measurements*, 2nd Ed. CT Putnam Publishing and Omega Press: Stamford
8. Gulo, Aktuallitas. (2017). MIKROKONTROLER AVR ATMega32
9. DCC, Deka. 2016. *Membangun Sistem Akuisisi Data Berbasis HM**I**i*. Jakarta
10. Setyawan, Angga. Tanpa tahun. *Perancangan Sitem Monitoring Temperature pada Superheater Steam di PLTU Unit Pembangkit Gresik*. Surabaya : Institut Teknologi Sepuluh Nopember
11. Atmel. (2011). Atmega32. San Jose, USA
12. <https://datasheetspdf.com/pdf/729350/Himax/HX8347-D/1/> diakses pada tanggal 29 Mei 2018
13. Sukamta, Sri & Kusmanto, Adhi. (2013). Perencanaan Mesin Produksi Air Minum Dalam Kemasan. Jurnak Teknik Elektro Vol.5 No.2

14. Purbawati,Anike.2010,"*Perancangan Sistem Monitoring Suhu Keluaran Steam Separator dalam Upaya Peningkatan Kualitas Output Steam di PT Pertamina Gheothermal Energy Area Kamojang*",Fakultas Teknologi Industri: Institut Teknologi Sepuluh Nopember .
15. Forest M Mims III. 1989. *Engineer's Notebook "Communications Project"*. Texas, USA. Archer
16. Putra, Agfianto Eko and Nugraha, Dhani. 2010. Tutorial Pemrograman Mikrokontroler AVR
17. Yogyakarta, U. A. (2017). *Jurnal Ilmiah Data Manajemen dan Teknologi Informasi*. Yogyakarta: Universitas Amikom Yogyakarta.
18. Soulié, Juan. 2007 .C++ Language Tutorial

LAMPIRAN

LAMPIRAN ***LISTING PROGRAM PADA CAVAVR***

This program was created by the CodeWizardAVR V3.32a
Automatic Program Generator

◆ Copyright 1998-2017 Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfotech.com>

Project :

Version :

Date : 5/30/2018

Author :

Company :

Comments:

Chip type : ATmega32A

Program type : Application

AVR Core Clock frequency: 16.000000 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 512

*****/

```
#include <32a.h>

#include <delay.h>

// SPI functions
#include <spi.h>

// Alphanumeric LCD functions
#include <alcd.h>

#ifndef RXB8
#define RXB8 1
```

```
#endif

#ifndef TXB8
#define TXB8 0
#endif

#ifndef UPE
#define UPE 2
#endif

#ifndef DOR
#define DOR 3
#endif

#ifndef FE
#define FE 4
#endif

#ifndef UDRE
#define UDRE 5
#endif

#ifndef RXC
#define RXC 7
#endif

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

// USART Receiver buffer
#define RX_BUFFER_SIZE 8
char rx_buffer[RX_BUFFER_SIZE];

#if RX_BUFFER_SIZE <= 256
unsigned char rx_wr_index,rx_rd_index,rx_counter;
#else
unsigned int rx_wr_index,rx_rd_index,rx_counter;
```

```

#endif

// This flag is set on USART Receiver buffer
overflow
bit rx_buffer_overflow;

// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
char status,data;
status=UCSRA;
data=UDR;
if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
{
    rx_buffer[rx_wr_index++]=data;
#if RX_BUFFER_SIZE == 256
    // special case for receiver buffer size=256
    if (++rx_counter == 0) rx_buffer_overflow=1;
#else
    if (rx_wr_index == RX_BUFFER_SIZE)
        rx_wr_index=0;
    if (++rx_counter == RX_BUFFER_SIZE)
    {
        rx_counter=0;
        rx_buffer_overflow=1;
    }
#endif
}
}

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;

```

```

while (rx_counter==0);
data=rx_buffer[rx_rd_index++];
#if RX_BUFFER_SIZE != 256
if (rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
#endif
#asm("cli")
--rx_counter;
#asm("sei")
return data;
}
#pragma used-
#endif

// Standard Input/Output functions
//#include <stdio.h>

// Declare your global variables here

// Voltage Reference: AREF pin
#define ADC_VREF_TYPE ((0<<REFS1) | (0<<REFS0) |
(0<<ADLAR))

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input | ADC_VREF_TYPE;
// Delay needed for the stabilization of the ADC
input voltage
delay_us(10);
// Start the AD conversion
ADCSRA|=(1<<ADSC);
// Wait for the AD conversion to complete
while ((ADCSRA & (1<<ADIF))==0);
ADCSRA|=(1<<ADIF);
return ADCW;
}

int jumlah_gelas=0;  int buckgelas=0;
// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)

```

```

{
// Place your code here
jumlah_gelas++;
if (jumlah_gelas>5) {jumlah_gelas=0; }

}

// SPI functions
#include <spi.h>
#include <stdio.h>
unsigned result;
/* LCD display buffer */
char lcd_buffer[33];

void kirim_suhu(int suhu) {
if(suhu<255) {
putchar('D');
putchar(suhu); }
else{putcha,r('D'); putchar(255); }

}

int dat;

void main(void)
{
// Port B initialization
// Function: Bit7=Out Bit6=In Bit5=Out Bit4=Out
Bit3=In Bit2=In Bit1=In Bit0=In
DDRB=(1<<DDB7) | (0<<DDB6) | (1<<DDB5) | (1<<DDB4)
| (0<<DDB3) | (0<<DDB2) | (0<<DDB1) | (0<<DDB0);
// State: Bit7=0 Bit6=T Bit5=0 Bit4=0 Bit3=T
Bit2=T Bit1=T Bit0=T
PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) |
(0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) |
(0<<PORTB1) | (0<<PORTB0);
// SPI initialization
// SPI Type: Master
// SPI Clock Rate: 2764,800 kHz
}

```

```

// SPI Clock Phase: Cycle Start
// SPI Clock Polarity: Low
// SPI Data Order: MSB First
PORTD=0x03;
DDRD=0x02;

SPCR=(0<<SPIE) | (1<<SPE) | (0<<DORD) | (1<<MSTR)
| (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);
SPSR=(0<<SPI2X);
// Alphanumeric LCD initialization
// Connections are specified in the
//                                         Project|Configure|C
Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTA Bit 0
// RD - PORTA Bit 1
// EN - PORTA Bit 2
// D4 - PORTA Bit 4
// D5 - PORTA Bit 5
// D6 - PORTA Bit 6
// D7 - PORTA Bit 7
// Characters/line: 16

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Rising Edge
// INT1: Off
// INT2: Off
GICR|=0x40;
MCUCR=0x03;
MCUCSR=0x00;
GIFR=0x40;

//           Timer(s)/Counter(s)           Interrupt(s)
initialization
TIMSK=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No
Parity

```

```

// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x98;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x4D;

// Global enable interrupts
#asm("sei")
lcd_init(16);
delay_ms(100);
lcd_puts("komunikasi ok");
puts("komunikasi ok"); delay_ms(500);
PORTC.4 = 1;

while (1)
{
    /* if (buckgelas!=jumlah_gelas){
        putchar('C');putchar(jumlah_gelas);
        buckgelas=jumlah_gelas;
    }
        while(rx_counter){
            dat=getchar();
            putchar(dat);
            lcd_gotoxy(13,0);lcd_putchar(dat);
    delay_ms(100);
    }*/
    /* read the MSB using SPI */
    PORTB.4=0;
    result=(unsigned) spi(0)<<8;
    /* read the LSB using SPI and combine with MSB */
    result|=spi(0);
    /* calculate the voltage in [mV] */
}

```

```
result=(unsigned)          (((unsigned)      long)
result*5000)/4096L);

lcd_clear();
lcd_gotoxy(0,0);lcd_puts(lcd_buffer);
sprintf(lcd_buffer,"Uadc=%4u.%u
%cC",result/40,(result%40),0xDF);
kirim_suhu(result/40);

lcd_gotoxy(0,1);lcd_puts(lcd_buffer);
delay_ms(100);
    if(result > 35)
{
    PORTC.4 = 0;
}
else if(result > 30 && result <= 35)
{
    PORTC.4 = 1;
}
}

}
```

LAMPIRAN

LISTING PROGRAM UNTUK TAMPILAN LCD

```
void desktop() {
    tft.setTextColor(TFT_BLUE);
    //tft.setTextSize(3);
    tft.setCursor(160-140, 100);
    tft.print("HAVE A NICE DAY"); delay(3000);
    tft.fillScreen(tft.color565(220,220,220));
    tft.setTextColor(TFT_BLACK);

    tft.fillRect(160+60, 0, 100, 54,TFT_BLUE);
    tft.fillRect(160+62, 2, 96, 50,TFT_CYAN);
    tft.setCursor(160+120, 4); //posisi kursor
    //tft.setTextSize(1);
    tft.print("Erche");
    tft.setTextSize(1);
    tft.setCursor(160+80, 13);
    tft.print("TEA PACKING"); delay(400);
    tft.setCursor(160+90,30);
    tft.print("MACHINE"); delay(400);

    out_suhu(0);
    out_suhu2(0);
    out_kekeruhan(0);
    out_gelas(0);
}

void out_suhu(int a)
{
    //DRAWING TERMOMETER
    tft.setTextColor(TFT_GREEN);
    tft.drawRect(16, 8, 11, 50,TFT_BLACK);
    tft.drawRect(17, 9, 9, 48,TFT_BLACK);
    tft.fillCircle(21, 66, 12, TFT_BLACK);
    tft.fillCircle(21, 66, 10, TFT_RED);
    tft.fillRect(18, 12, 7, 50,TFT_RED);
}
```

```

tft.fillRect(18, 12, 7, 50,TFT_RED);
tft.fillRect(18, 10, 7, 45-a/2.5,TFT_WHITE);
tft.fillRect(40,30,46,30,
tft.color565(220,220,220));
tft.setTextSize(2);
tft.setCursor(34, 30);
tft.print(":"); tft.print(a);
tft.setCursor(80, 30);tft.print(" C");
tft.drawCircle(83, 32, 2, TFT_GREEN);
tft.setTextColor(TFT_BLACK);
}

void out_suhu2(int d)
{
    //DRAWING TERMOMETER
    tft.setTextColor(TFT_BLUE);
    int gx=100;
    tft.drawRect(16+gx, 8, 11, 50,TFT_BLACK);
    tft.drawRect(17+gx, 9, 9, 48,TFT_BLACK);
    tft.fillCircle(21+gx, 66, 12, TFT_BLACK);
    tft.fillCircle(21+gx, 66, 10, TFT_RED);
    tft.fillRect(18+gx, 12, 7, 50,TFT_RED);

    tft.fillRect(18+gx, 12, 7, 50,TFT_RED);
    tft.fillRect(18+gx, 10, 7, 45-d/2.5,TFT_WHITE);

    tft.fillRect(40+gx,30,46,30
,tft.color565(220,220,220));
    tft.setTextSize(2);
    tft.setCursor(34+gx, 30);
    tft.print(":"); tft.print(d);
    tft.setCursor(80+gx, 30);tft.print(" C");
    tft.drawCircle(83+gx, 32, 2, TFT_BLUE);
    tft.setTextColor(TFT_BLACK);
}
void out_kekeruhan(int a)
{
    tft.fillRect(6,110 ,210,
,tft.color565(220,220,220));
    for(int n=10 ; n<=110 ;n+=1)

```

```

{
    tft.fillRect(n,110 ,1, 40 ,tft.color565(255-
n*1.5,255-n*1.7,255-n*2));
}

tft.fillTriangle(10+a, 135, 6+a, 149, 14+a, 149
, TFT_BLACK);

tft.setTextSize(3);
tft.setCursor(48, 120);
tft.print(":");tft.print(a); tft.print("%");

}

void out_gelas(int a)
{
    tft.fillRect(5,170 ,304, 66
,tft.color565(220,220,220));
    for(int n=0;n<a;n++)
    {

        tft.drawRoundRect(5+n*45, 180, 35, 10,7,
TFT_BLACK); tft.drawRoundRect(6+n*45, 181, 33,
8,4, TFT_BLACK);
        tft.drawRoundRect(11+n*45, 230, 23, 6,3,
TFT_BLACK); tft.drawRoundRect(12+n*45, 231, 21,
4,1, TFT_BLACK);
        tft.drawLine(5+n*45,183,11+n*45,232,
TFT_BLACK);
        tft.drawLine(39+n*45,183,33+n*45,232,
TFT_BLACK);
        tft.drawLine(38+n*45,184,33+n*45,230,
TFT_BLACK);
    }
    tft.fillRect(270,200 ,45, 35
,tft.color565(220,220,220));

    tft.setTextSize(2);    tft.setCursor(270, 200);
tft.print(":");    tft.print(a);
}

```

LAMPIRAN

LISTING PROGRAM UNTUK PENYIMPANAN

```
void simpan(char x,char n,int l,int data){  
    char namaf[]="Xn.txt";  
    namaf[0]=x; namaf[1]=n;  
    myFile = SD.open(namaf, FILE_WRITE);  
    if (myFile) {  
        Serial.print("Writingtofile...");  
        Serial.println(namaf);  
  
        myFile.print(l);myFile.print(":");myFile.println  
(data);  
        myFile.close();  
        Serial.println("Done.");  
    }  
    // if the file didn't open, print an error:  
    else {  
        Serial.print("erroropening")  
;Serial.print(namaf); Serial.println("...");  
    }  
  
}  
  
void tampil(char x, char n){  
    char data[46]={};  
    char namafile[]={"Xn.txt"};  
    namafile[0]=x; namafile[1]=n;  
  
    begin:  
    int index=0;  
    int bar=0; int col=0;  
    tft.fillRect(tft.color565(220,220,220));  
    tft.drawRect(0, 0, 320, 240,TFT_BLUE);  
    tft.drawRect(1, 1, 318, 238,TFT_BLUE);
```

```
tft.setCursor(90,8);
tft.println("DATALOgger");
Serial.println(namafile);
tft.setCursor(290,220); bar; tft.println(hal);
myFile = SD.open(namafile);
if (myFile) {
    Serial.println("Reading...");
    tft.setCursor(10,20); bar;
tft.print(namafile); tft.setTextSize(1);
    Serial.println(namafile[0]);

    while (myFile.available()) {
        data[idx]=myFile.read();
        Serial.write(data[idx]);
        idx++;

        if(data[idx-1]=='\n')
        {
            tft.setCursor(10+col,40+bar); bar+=10;
            tft.println(data); idx=0;
            if(bar==190){bar=0;
            if(col<240)col+=40;
            else{break;} //hal[1]++;
            tft.fillScreen(tft.color565(220,220,220));
            tampil(x,n+1);
        }
        myFile.close();
    }
    else {
        Serial.print("error opening ");
        Serial.println(namafile);
    }
}
```

LAMPIRAN

DATA SHEET SYSTEM ATMEGA 32

Features

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
 - 32Kbytes of In-System Self-programmable Flash program memory
 - 1024Bytes EEPROM
 - 2Kbyte Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four PWM Channels
 - 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels in TQFP Package Only
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
 - Byte-oriented Two-wire Serial Interface
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
 - 32 Programmable I/O Lines
 - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
 - 2.7V - 5.5V for ATmega32L
 - 4.5V - 5.5V for ATmega32
- Speed Grades
 - 0 - 8MHz for ATmega32L
 - 0 - 16MHz for ATmega32
- Power Consumption at 1 MHz, 3V, 25°C
 - Active: 1.1mA
 - Idle Mode: 0.35mA
 - Power-down Mode: < 1µA



8-bit **AVR**[®]
Microcontroller
with 32KBytes
In-System
Programmable
Flash

ATmega32
ATmega32L

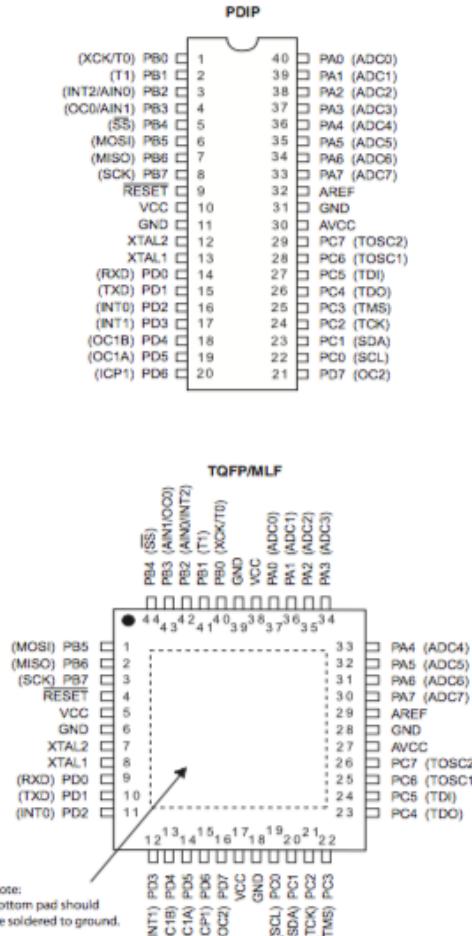
Summary



ATmega32(L)

Pin Configurations

Figure 1. Pinout ATmega32



ATmega32(L)

The Atmel® AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega32 provides the following features: 32Kbytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 1024bytes EEPROM, 2Kbyte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire interface, A/D Converter, SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega32 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The Atmel AVR ATmega32 is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

Pin Descriptions

VCC	Digital supply voltage.
GND	Ground.
Port A (PA7..PA0)	Port A serves as the analog inputs to the A/D Converter. Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.



ATmega32(L)

Port B (PB7..PB0)	Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port B also serves the functions of various special features of the ATmega32 as listed on page 57 .
Port C (PC7..PC0)	Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs. The TDO pin is tri-stated unless TAP states that shift out data are entered. Port C also serves the functions of the JTAG interface and other special features of the ATmega32 as listed on page 60 .
Port D (PD7..PD0)	Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port D also serves the functions of various special features of the ATmega32 as listed on page 62 .
RESET	Reset Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 37 . Shorter pulses are not guaranteed to generate a reset.
XTAL1	Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
XTAL2	Output from the inverting Oscillator amplifier.
AVCC	AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to V_{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.
AREF	AREF is the analog reference pin for the A/D Converter.

ATmega32(L)

Resources	A comprehensive set of development tools, application notes and datasheets are available for download on http://www.atmel.com/avr .
Data Retention	Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.
About Code Examples	This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C Compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C Compiler documentation for more details.

BIODATA PENULIS

BIODATA PENULIS



Penulis yang bernama Erche Berlian Fajarindah dilahirkan di Mojokerto pada tanggal 24 Februari 1997. Penulis merupakan anak kedua dari dua bersaudara dari orang tua Bapak Edy Soekamto dan Ibu Wahjoe Sri Redjeki. Penulis menempuh pendidikan formal di TK Yaminna, SDN Wates III, SMPN 1 Mojokerto, dan SMAN 1 Sooko Mojokerto. Kemudian penulis melanjutkan ke jenjang yang lebih tinggi yaitu di Institut Teknologi Sepuluh Nopember Surabaya, tepatnya di Departemen

Teknik Instrumentasi Fakultas Vokasi di tahun 2015 dan terdaftar dengan NRP 10 51 15 000 00 025.

Di Departemen Teknik Instrumentasi ini penulis aktif di organisasi kemahasiswaan dan kepanitiaan seperti di SPE ITS SC sebagai Staff Eksternal periode 2016/2017. Pada laporan ini, penulis telah melaksanakan Kerja Praktik di PT. Geodipa Energi, Dieng.

Selesainya Tugas Akhir ini diajukan penulis untuk memenuhi syarat memperoleh gelar Ahli Madya di Departemen Teknik Instrumentasi Fakultas Vokasi Institut Teknologi Sepuluh Nopember Surabaya.

efajarindah@gmail.com
[linkedin.com/erchefajarindah](https://www.linkedin.com/in/erchefajarindah)
[Instagram.com/erchefajarindah](https://www.instagram.com/erchefajarindah)