

**PEMBUATAN ALAT BANTU MONITORING KADAR CO₂ DAN O₂
PADA PROSES FERMENTASI BIJI KAKAO**

**SKRIPSI
KONSENTRASI TEKNIK ELEKTRONIKA**

*Diajukan Untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Teknik*

UNIVERSITAS BRAWIJAYA



Disusun oleh :

AHMAD HADI

NIM. 0410630007-63

DEPARTEMEN PENDIDIKAN NASIONAL

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

JURUSAN TEKNIK ELEKTRO

MALANG

2009

**PEMBUATAN ALAT BANTU MONITORING KADAR CO₂ DAN O₂
PADA PROSES FERMENTASI BIJI KAKAO**

SKRIPSI

*Diajukan Untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Teknik*



Disusun oleh :
AHMAD HADI
NIM. 0410630007-63

Telah diperiksa dan disetujui oleh:
DOSEN PEMBIMBING

Dosen Pembimbing

Dosen Pembimbing

Ir. M. Julius St., MS.
NIP. 19540720 198203 1 002

Panca Mudjirahardjo, ST. MT.
NIP. 19700329 200012 1 002



PEMBUATAN ALAT BANTU MONITORING KADAR CO₂ DAN O₂ PADA PROSES FERMENTASI BIJI KAKAO

Skripsi ini telah di uji dan dinyatakan lulus
Pada Tanggal 11 Agustus 2009

Disusun oleh :

AHMAD HADI
NIM. 0410630007-63

Majelis Penguji :

Ir. Ponco Siwindarto, MS
NIP. 19590304 198903 1 001

Waru Djuriatno, ST. MT.
NIP. 19690725 199702 1 001

Mochammad Rif'an, ST. MT.
NIP. 19710301 200012 1001

Mengetahui,
Ketua Jurusan Teknik Elektro

Ir. Heru Nurwarsito, M.Kom
NIP. 19650402 199002 1 001

PENGANTAR

Segala pujian hanya milik Allah, Pencipta Manusia, alam semesta dan kehidupan. Tiada Illah yang patut untuk disembah di taati aturannya kecuali hanya kepada Allah semata.

Sholawat dan salam juga senantiasa tetap tercurah limpahkan kepada Nabi Muhammad, para sahabat, keluarganya serta umatnya yang senantiasa memegang teguh risalah Islam dan juga mengamalkan serta memperjuangkan islam dalam sistem kehidupan secara kaffah sampai hari kiamat.

Hanya karena pertolongan Allah semata penulis mampu melewati segala kendala yang ada selama penyusunan skripsi ini sehingga penyusunan skripsi ini dapat terselesaikan. Skripsi berjudul “Pembuatan Alat Bantu Monitoring Kadar CO₂ dan O₂ Pada Proses Fermentasi Biji Kakao” ini disusun sebagai salah satu syarat untuk mendapatkan gelar Sarjana Teknik di Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.

Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, dengan kesungguhan dan kerendahan hati penulis menyampaikan terima kasih kepada:

- ◆ Bapak dan Ibu yang aku cintai karena-Nya, *jazakumullah khairan katsir*. Semoga Allah SWT membalas kasih sayang dan kebaikan kalian selama ini dengan kebaikan yang jauh lebih banyak,
- ◆ Bapak Ir. Heru Nurwarsito, M.Kom selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya dan Bapak Rudy Yuwono, ST, M.Sc selaku Sekretaris Jurusan Teknik Elektro.
- ◆ Bapak Ir. M. Julius St., MS selaku KKDK Elektronika Jurusan Teknik Elektro dan sekaligus dosen pembimbing yang telah banyak memberikan motivasi untuk segera menyelesaikan skripsi.
- ◆ Bapak Panca Mudjirahardjo, ST. MT. selaku dosen pembimbing skripsi yang telah banyak memberikan pengarahan dan bimbingan dalam penyelesaian skripsi ini serta atas segala bentuk bantuan dan saran yang membangun.
- ◆ Bapak, Ibu dosen serta segenap staf dan karyawan Jurusan Teknik Elektro yang telah banyak membantu dan memberikan ilmunya dengan ikhlas, Semoga Allah membalas dengan sesuatu yang lebih baik.

- ◆ Saudara-saudaraku, Mas Rizal, dik Lini dan adik adi yang saya sayangi dan menyayangiku dan semua keluargaku, *jazakumullah khairan katsir* atas dukungannya selama ini,
- ◆ Dr. Sri Mulato, Pak Edy, mbak ifa, pak kaswan, wahid serta semua karyawan Puslit Biji Kopi dan Kakao yang di jember yang telah memberikan banyak membantu penyelesaian skripsi ini, semoga alat ini bermanfaat dalam kemajuan Puslit ke depan.
- ◆ Mas burhan, mas riki, mas agus dan munir yang senantiasa yang telah banyak memberikan bantuan, motifasi, ilmu, sarana dan prasarana yang telah diberikan, semoga bantuan kalian di balas Allah dengan kebaikan.
- ◆ Temen-temen angkatan 2004, saya mohon maaf atas kesalahan yang tidak berkenan selama ini dan terima kasih untuk semuanya dan semoga kalian senantiasa menjadi orang-orang yang sukses di dunia dan juga di akhirat kelak
- ◆ Semua temen-temenku aktivis dakwah, semoga kita selalu istiqomah dalam perjuangan dakwah Islam untuk diterapkan secara kaffah ini, dan semoga kita senantiasa dikumpulkan dalam surgaNya
- ◆ Teman-teman elektro angkatan 2002, 2003, 2004, 2005 dan teman-temanku yang lain yang tidak mungkin untuk saya tulis semua namanya, terima kasih banyak atas bantuan dan dukungannya.

Akhir kata, penulis menyadari bahwa skripsi ini masih belum sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan. Penulis sangat berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi, terutama bagi Pusat Penelitian Kopi dan Kakao Indonesia, PTPN XII, Jember.

Malang, Agustus 2009

Penulis

DAFTAR ISI

PENGANTAR.....	i
DAFTAR ISI.....	iii
RINGKASAN.....	viii
BAB I.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Ruang Lingkup.....	2
1.4 Tujuan.....	2
1.5 Sistematika Penulisan.....	2
BAB II.....	4
2.1 Fermentasi Biji Kakao.....	4
2.2 Sensor TGS4161.....	7
2.3 Sensor KE-50.....	8
2.4 Operasional Amplifier.....	10
2.4.1 Penguat Penyangga (Buffer).....	11
2.4.2 Penguat Tak Membalik (Non-Inverting Amplifier).....	11
2.5 Mikrokontroler ATmega8P.....	12
2.6 Seven Segment.....	19
BAB III.....	20
3.1 Studi Literatur.....	20
3.2 Perancangan Alat.....	21
3.3 Pembuatan Alat.....	21
3.4 Pengujian Alat.....	21
3.5 Pengambilan Kesimpulan.....	22
BAB IV.....	23
4.1 Spesifikasi Alat.....	23
4.2 Blok Diagram Sistem.....	23
4.3 Prinsip Kerja Alat.....	25
4.4 Perancangan Perangkat Keras.....	26
i. Perancangan Rangkaian Sensor.....	26
4.4.1.1 Perancangan Rangkaian Sensor CO2.....	26

4.4.1.2 Perancangan Rangkaian Sensor O2.....	27
ii. Perancangan Rangkaian Pengkondisi Sinyal.....	28
1. Perancangan RPS Sensor CO2.....	28
2. Perancangan RPS Sensor O2.....	29
iii. Mikrokontroler ATmega8P.....	30
iv. Perancangan Display.....	32
4.5 Perancangan Perangkat Lunak.....	33
BAB V.....	37
5.1 Pengujian Sensor.....	37
5.1.1 Tujuan.....	37
5.1.2 Metode Pengujian.....	37
5.1.3 Prosedur Pengujian.....	38
5.1.4 Analisis dan Hasil Pengujian.....	38
5.2 Pengujian Rangkaian Pengkondisi Sinyal.....	41
5.2.1 Tujuan.....	41
5.2.2 Prosedur Pengujian.....	41
5.2.3 Hasil Pengujian.....	41
5.3 Pengujian ADC Internal.....	43
5.3.1 Tujuan.....	43
5.3.2 Prosedur Pengujian.....	43
5.3.3 Hasil Pengujian.....	43
5.4 Pengujian Display.....	44
5.4.1 Tujuan.....	44
5.4.2 Prosedur pengujian.....	44
5.4.3 Hasil pengujian.....	45
5.5 Pengujian keseluruhan.....	45
5.5.1 Tujuan.....	45
5.5.2 Prosedur pengujian.....	45
5.5.3 Hasil dan Analisa pengujian.....	46
BAB VI.....	50
6.1 Kesimpulan.....	50
6.2 Saran.....	51
DAFTAR PUSTAKA.....	52



DAFTAR TABEL

Tabel 2.1	Komposisi bahan pada pulpa Biji Kakao.....	5
Tabel 2.2	Perubahan Selama Proses fermentasi.....	6
Tabel 2.3	Parameter pengoperasian sensor Gas CO ₂	8
Tabel 2.4	Parameter pengoperasian sensor Gas O ₂	9
Tabel 5.1	Data Hasil Pengujian Sensor TGS4161.....	38
Tabel 5.2	Data Hasil Pengujian Sensor KE-50.....	40
Tabel 5.3	Hasil Pengujian RPS untuk Sensor TGS4161.....	41
Tabel 5.4	Hasil Pengujian RPS untuk Sensor KE-50.....	42
Tabel 5.5	Hasil Pengujian ADC internal Mikrokontroler Atmega8P.....	43
Tabel 5.6	Hasil Pengujian Kemampuan Alat.....	46
Tabel 5.7	Hasil Pengukuran untuk O ₂	47
Tabel 5.8	Hasil Pengukuran untuk CO ₂	48



DAFTAR GAMBAR

Gambar 2.1	Perubahan Warna Pada Biji Kakao.....	5
Gambar 2.2	Grafik karakteristik dari sensor TGS4161.....	7
Gambar 2.3	Grafik karakteristik dari sensor KE-50.....	9
Gambar 2.4	Penguat Penyangga.....	11
Gambar 2.5	Penguat Non-Inverting.....	12
Gambar 2.6	Arsitektur AVR.....	13
Gambar 2.7	Peta memori AVR.....	14
Gambar 2.8	Blok Diagram ADC.....	15
Gambar 2.9	EEPROM <i>Address Register</i>	17
Gambar 2.10	EEPROM <i>Data Register</i>	17
Gambar 2.11	EEPROM <i>Control Register</i>	18
Gambar 2.12	Led Seven Segmen.....	19
Gambar 4.1	Blok Diagram Alat.....	24
Gambar 4.2	Perancangan Rangkaian Sensor CO ₂	25
Gambar 4.3	Perancangan Rangkaian Sensor O ₂	26
Gambar 4.4	Perancangan RPS Sensor CO ₂	28
Gambar 4.5	Perancangan RPS Sensor O ₂	29
Gambar 4.6	Perancangan Mikrokontroler ATmega8 sebagai kontrol utama.....	31
Gambar 4.7	Blok Perancangan Display.....	32
Gambar 4.8	Diagram Alir Fungsi Utama.....	33
Gambar 4.9	Diagram Alir Baca ADC.....	34
Gambar 4.10	Diagram Interrupt Timer.....	35
Gambar 5.1	Grafik Karakteristik Sensor Gas CO ₂ dalam datasheet.....	37
Gambar 5.2	Grafik Karakteristik Sensor Gas O ₂ dalam datasheet.....	39
Gambar 5.3	Hasil Pengujian Display	44
Gambar 5.4	Blok Diagram pengujian Sistem secara Keseluruhan	45

RINGKASAN

Ahmad Hadi, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, 2008, *Pembuatan Alat Bantu Monitoring Kadar CO₂ dan O₂ Pada Proses Fermentasi Biji Kakao*. Dosen Pembimbing : Ir. M. Julius St., MS, Panca Mudjiraharjo, ST., MT.

Fermentasi merupakan bagian penting dari proses pengolahan biji kakao. Proses ini tidak hanya bertujuan untuk membebaskan biji kakao dari pulpanya namun juga untuk membentuk citarasa coklat yang enak dan menyenangkan serta mengurangi rasa sepat dan pahit pada biji Kakao. Pada awalnya, kondisi dalam massa biji sangat cocok untuk pertumbuhan khamir. Khamir akan merubah sebagian besar gula dalam pulpa menjadi alkohol. Reaksi pembentukan alkohol ini akan menghasilkan sejumlah besar karbondioksida (CO₂). Proses fermentasi ini berlangsung secara alami oleh mikroba dengan bantuan oksigen (O₂) dari udara. Mikroba memanfaatkan senyawa gula yang ada di dalam pulpa sebagai media tumbuh sehingga lapisan pulpa terurai menjadi encer dan keluar lewat lubang-lubang di dasar dan dinding peti fermentasi (Rohan, 1963).

Reaksi kimia yang terjadi pada saat proses fermentasi tersebut menghasilkan kadar CO₂ dengan jumlah tertentu melalui bantuan O₂ dari udara. Besaran fisis yang terjadi tersebut di butuhkan para peneliti Biji kakao untuk membuat pengembangan-pengembangan baru sehingga menghasilkan cita rasa coklat yang sempurna dan waktu fermentasi yang semakin singkat. Salah satunya bagaimana mengamati besaran Gas yang terjadi saat fermentasi dengan menggunakan alat yang berbasis digital. Solusi untuk masalah tersebut adalah memanfaatkan sensor CO₂ dan O₂ sebagai Alat Bantu Monitoring dalam Proses Fermentasi Biji Kakao.

Alat ini tersusun atas mikrokontroller Atmega8 sebagai kontrol utamanya, sensor TGS 4161 sebagai detektor Gas CO₂, Sensor KE-50 sebagai detektor kadar O₂, dilengkapi dengan OpAmp TLC271 sebagai komponen Utama rangkaian Pengkondisi Sinyalnya dan Seven Segment untuk menampilkan nilai ukur yang terdeteksi. Dari hasil pengujian diperoleh bahwa alat mampu mengukur nilai kadar Oksigen antara 5-97% dan Karbondioksida antara 350-4965ppm.

Kata kunci: CO₂, O₂, Monitoring, Fermentasi

BAB I PENDAHULUAN

1.1 Latar Belakang

Biji Kakao dari Perkebunan Biji Kakao di Indonesia cenderung masih bermutu rendah. Rendahnya mutu biji kakao, terutama disebabkan oleh cara pengolahan yang kurang baik, seperti biji kakao yang tidak difermentasi atau proses fermentasi yang kurang baik. Fermentasi merupakan inti dari proses pengolahan biji kakao. Proses ini tidak hanya bertujuan untuk membebaskan biji kakao dari pulpanya namun juga untuk membentuk citarasa coklat yang enak dan menyenangkan serta mengurangi rasa sepat dan pahit pada biji Kakao (Widyotomo, dkk, 2004).

Pada awalnya, kondisi biji kakao sangat cocok untuk pertumbuhan khamir. Khamir akan merubah sebagian besar gula dalam pulpa menjadi alkohol. Reaksi pembentukan alkohol ini akan menghasilkan sejumlah besar karbondioksida (CO_2). Proses fermentasi ini berlangsung secara alami oleh mikroba dengan bantuan oksigen (O_2) dari udara. Mikroba memanfaatkan senyawa gula yang ada di dalam pulpa sebagai media tumbuh sehingga lapisan pulpa terurai menjadi encer dan keluar lewat lubang-lubang di dasar dan dinding peti fermentasi (Rohan, 1963).

Faktor yang berpengaruh terhadap keberhasilan fermentasi adalah wadah fermentasi, waktu, aerasi, pembalikan, aktivitas mikroba dan penguraian kandungan pulpa. Penguraian kandungan pulpa ditentukan dengan lamanya pemeraman buah kakao setelah dipetik.

Reaksi kimia yang terjadi pada saat proses fermentasi tersebut menghasilkan kadar CO_2 dengan jumlah tertentu melalui bantuan O_2 dari udara. Besaran fisis yang terjadi tersebut di butuhkan para peneliti Biji kakao untuk membuat pengembangan-pengembangan baru sehingga menghasilkan cita rasa coklat yang sempurna dan waktu fermentasi yang semakin singkat. Salah satunya bagaimana mengamati besaran Gas yang terjadi saat fermentasi dengan menggunakan alat yang berbasis digital. Solusi untuk masalah tersebut adalah memanfaatkan sensor CO_2 dan O_2 sebagai Alat Bantu Monitoring dalam Proses Fermentasi Biji Kakao.

1.2 Rumusan Masalah

Dalam perancangan ini rumusan masalah ditekankan pada :

- Bagaimana merancang sensor yang berfungsi untuk memonitor Gas CO₂ dan O₂ yang dihasilkan pada saat proses fermentasi
- Bagaimana membuat program yang dapat menjalankan semua rangkaian dengan bekerja baik menggunakan mikrokontroler ATmega8P
- Bagaimana membuat dan merancang antarmuka rangkaian dengan mikrokontroler

1.3 Ruang Lingkup

Mengacu pada permasalahan yang ada maka pada tugas akhir ini dibatasi pada:

- Sensor yang dipakai yaitu TGS4161 dan KE-50
- Tidak membahas power supply
- Alat ini hanya memonitor saja tanpa memberikan perlakuan pada sistem
- Parameter keberhasilan alat adalah sesuai dengan spesifikasi alat yang diinginkan.

1.4 Tujuan

Tujuan penyusunan skripsi ini adalah merencanakan dan membuat sebuah alat yang dapat memonitor proses fermentasi biji kakao di Pusat Penelitian Kopi dan Kakao Indonesia.

1.5 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penyusunan tugas akhir ini adalah sebagai berikut:

BAB I Pendahuluan

Memuat tentang latar belakang permasalahan, rumusan masalah, ruang lingkup pembahasan, tujuan penulisan, serta sistematika penulisan.

Bab II Tinjauan Pustaka

Membahas teori-teori dasar dari beberapa piranti dan metode yang digunakan untuk menunjang perancangan dan pembuatan alat.

BAB III Metode Penelitian

Menjelaskan tahap-tahap dan metode yang dilakukan dalam perencanaan, pembuatan sampai dengan pengujian alat.

BAB IV Perancangan dan Pembuatan Alat

Memuat spesifikasi, diagram blok, prinsip kerja, perancangan dan pembuatan alat.

BAB V Hasil dan Pembahasan

Memuat hasil pengujian alat dan analisis terhadap data hasil pengujian menggunakan teori yang ada.

BAB VI Kesimpulan dan Saran

Memuat kesimpulan dan saran-saran untuk pengembangan lebih lanjut dari alat yang telah dibuat.



BAB II

TINJAUAN PUSTAKA

2.1 Fermentasi Biji Kakao

Biji kakao adalah bahan mentah untuk membuat cokelat. Untuk menjadi sebuah batangan coklat, biji kakao harus melewati beberapa proses pengolahan. Salah satu tahapan penting dalam penanganan pascapanen kakao adalah proses fermentasi. Penanganan pascapanen kakao dimulai sejak pemetikan buah, fermentasi sampai pengeringan dan pengemasan. Tahapan ini sangat penting dilalui untuk mempersiapkan biji kakao basah menjadi biji kakao kering bermutu tinggi dan layak dikonsumsi.

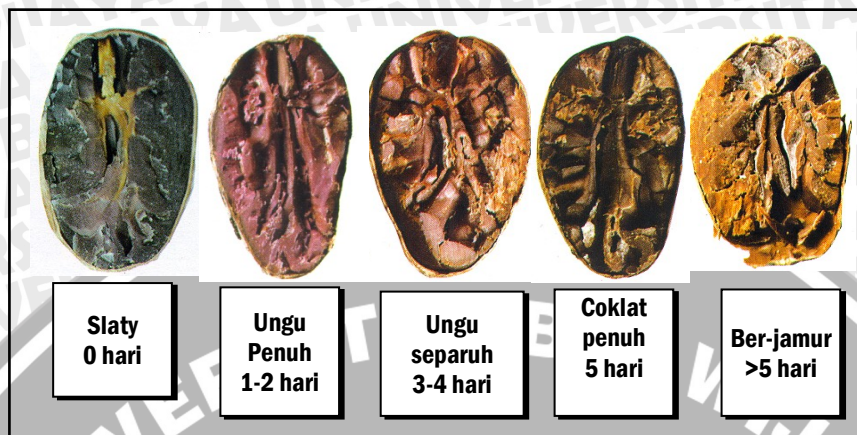
Kakao yang banyak di budidayakan oleh perkebunan rakyat adalah jenis *Forastero*. Dalam dunia perdagangan kakao jenis ini sering disebut kakao lindak atau *bulk cacao*. Buah kakao terdiri dari 3 komponen penyusun utama yaitu kulit buah, biji kakao dan plasenta. Kulit buah kakao merupakan komponen terbesar yaitu sekitar 70% dari berat buah kakao yang telah masak, 27%-29% dan sisanya adalah plasenta yang mengikat sekitar 30 biji sampai 40 biji.

Permukaan biji di selimuti oleh lapisan pulpa atau lendir yang berwarna putih. Biji kakao yang berasal dari buah yang matang memiliki plasenta yang lunak dan manis. Pulpa sangat penting agar proses fermentasi berlangsung dengan baik. Biji kakao yang telah difermentasi memiliki kegunaan, misalnya bahan baku pembuatan cokelat dan nata de coco.

Fermentasi biji kakao akan menumbuhkan citarasa, aroma dan warna cokelat, jenis biji kakao inilah yang banyak di cari oleh perusahaan Cokelat sebagai bahan dasarnya. Citarasa, aroma dan warna pada biji kakao mengalami perubahan selama fermentasi. Ini di sebabkan karena terjadi perubahan fisik, kimiawi dan biologi didalam biji kakao tersebut. Faktor yang berpengaruh terhadap keberhasilan fermentasi adalah wadah fermentasi, waktu, aerasi, pembalikan, aktivitas mikroba dan penguraian kandungan pulpa. Penguraian kandungan pulpa ditentukan dengan lamanya pemeraman buah kakao setelah dipetik.

Dalam Gambar 2.1. menjelaskan tentang proses perubahan warna yang terjadi selama fermentasi dalam selang waktu hari per hari. Biji kakao yang tidak difermentasi ditandai dengan ciri-ciri bertekstur pejal, berwarna slaty (keabu-abuan), memiliki rasa sangat pahit dan sepat. Biji kakao yang difermentasi dengan baik akan bertekstur agak

enyah atau mudah pecah, warna keping biji cokelat dengan sedikit warna ungu, memiliki citarasa coklat enak tanpa rasa pahit dan sepat. Biji kakao yang kurang fermentasi ditandai dengan ciri-ciri berwarna ungu, bertekstur pejal, didominasi oleh rasa pahit dan sepat, serta sedikit citarasa coklat.



Gambar 2.1 Perubahan warna pada Biji Kakao

Pusat Penelitian Biji kopi dan Kakao Indonesia yang berpusat di jember, adalah pusat yang aktivitasnya utamanya adalah penelitian. Para peneliti disana melakukan terobosan-terobosan baru untuk perkembangan kemajuan Biji Kopi dan Kakaonya dengan mutu dan kualitas yang baik. Beberapa penelitian di lakukan oleh para peneliti dalam hal mengembangkan Biji Kakao dengan kualitas yang baik dan proses pengolahan yang relative singkat adalah melakukan penelitian pada proses fermentasi.

Dari penelitian tersebut didapatkan bahwa proses fermentasi Biji Kakao dilakukan selama 5 hari terjadi perubahan kimia, fisika, dan biologi. Untuk mengetahui proses-proses yang terjadi selama fermentasi, di bawah ini akan di jelaskan tentang hal tersebut. Saat biji kakao di dikeluarkan dari buah, biji kakao di selimuti oleh lendir putih atau pulpa. Komposisi pulpa kakao segar di sajikan dalam Tabel 2.1

Tabel 2.1 Komposisi bahan pada pulpa Biji Kakao

No	Bahan	Kadar (%)
1	Air	84.5
2	Pentosan	2.7
3	Sukrosa	0.7
4	Glukosa, Fruktosa	10.0
5	Protein	0.6
6	Asam-asam	0.7
7	Garam-garam organik	0.8

Sumber : Hardy (1960) dalam (Wood & Lass, 1985)

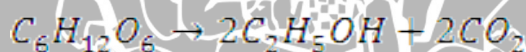
Pulpa tersebut pada mulanya steril, tetapi dengan adanya gula dan keasaman yang tinggi (pH 3,5) karena mengandung asam sitrat dan juga suhu semakin naik antara 27-42°C. Kondisi ini ideal untuk pertumbuhan mikroorganisme dan menyebabkan perubahan warna pada biji kakao.

Pada hari pertama sampai hari kedua (0-48jam), awalnya kotak fermentasi dalam kondisi anaerob (tanpa Oksigen) karena lapisan pulpa masih rapat dan tebal sehingga oksigen dari luar tidak dimungkinkan bisa masuk. Kondisi ini sangat cocok untuk pertumbuhan khamir yang akan merubah sebagian besar gula menjadi alkohol dan menghasilkan sejumlah besar CO₂ atau Karbondioksida. Spesies khamir yang sering ada adalah: *Caccharomyces cerevisiae*, *Candida rugosa* dan *Kluyveromyces marxianus*. Untuk lebih memudahkan, berikut ini di gambarkan tentang Perubahan yang terjadi selama 2 hari proses fermentasi yang di perlihatkan dalam Tabel 2.2.

Tabel 2.2 Perubahan Selama 2 Hari Proses Fermentasi

Time	Kondisi	Proses Kimia	Mikroba	Suhu
0 – 48 jam	Anaerobik [pulpa pekat]	Gula menjadi Alkohol + Karbondioksida	Yeast [saccaromyces]	27 – 42 °C

Dari tabel di atas dapat di tulis rumus Kimianya :

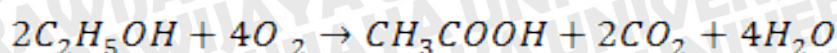


Setelah 2 hari, oksigen yang semula terhalang lapisan pulpa, dapat masuk ke dalam tumpukan biji Kakao. Kondisi aerob (kaya Oksigen-O₂) ini dimanfaatkan oleh aseto-bakteri untuk mengubah alkohol menjadi asam asetat dengan mengeluarkan bau khas yang menyengat. Tabel 2.3. menunjukkan proses perubahan yang terjadi setelah 2 hari pertama fermentasi.

Tabel 2.3. Perubahan Selama 3 Hari Proses Akhir Fermentasi

Time	Kondisi	Proses Kimia	Mikroba	Suhu
48 – 120 jam	Aerobik [pulpa encer]	Alkohol menjadi Asam Asetat + Karbondioksida	Bakteri [acetobacter]	45 – 50 °C

Dari tabel di atas dapat di tulis rumus Kimianya Rumus Kimianya :



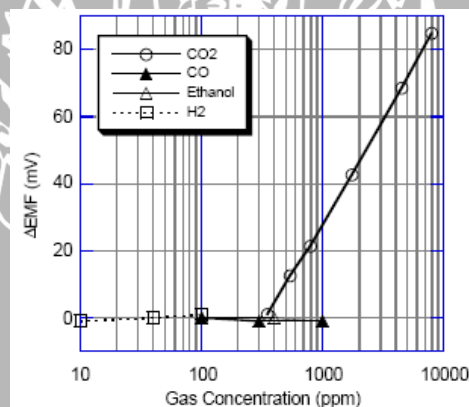
Jika fermentasi dilanjutkan terus akan muncul bau yang tidak diharapkan karena pertumbuhan bacillus dan kapang (*Aspergillus*, *Penicillium*, dan *Mucor*) yang

menghidrolisis lipid dalam biji dan menghasilkan asam lemak rantai pendek. pH menjadi sekitar 7 dan tumbuhlah *Pseudomonas*, *Enterobacter* atau *Escherichia coli* yang menghasilkan bau dan rasa tidak enak.

Dari teori di atas dapat disimpulkan bahwa kadar Oksigen, pada awalnya menurun dan semakin naik pada saat hari ke-3 sampai hari ke-5. sedangkan untuk kadar Gas CO₂, pada mulanya tidak ada gas CO₂ yang dihasilkan. Namun pada saat adanya perubahan gula menjadi alcohol, maka akan di hasilkan Gas karbondioksida yang semakin besar.

2.2 Sensor TGS4161

Sensor Gas CO₂ yang merupakan salah satu jenis transduser. Transduser ini digunakan sebagai komponen yang langsung mengadakan kontak dengan suatu media yang diukur dan berfungsi untuk mengubah konsentrasi gas CO₂ menjadi besaran gaya gerak listrik (ggl/EMF). Karakteristik dari sensor TGS 4161 di tunjukkan dalam Gambar 2.2



Gambar 2.2 Grafik Karakteristik dari sensor TGS 4161

Dari Grafik di atas, di jelaskan bahwa : $\Delta EMF = EMF1 - EMF2$

Dimana, $EMF1$ adalah EMF pada 350ppm CO₂

$EMF2$ adalah EMF yang terukur dari sensor

Untuk mendeteksi konsentrasi gas CO₂ yang akan di monitor di dalam wadah fermentasi Biji Kakao, di gunakan sensor TGS4161. Sensor TGS 4161 ini dapat bekerja dengan baik bila kondisi pengoperasiannya di penuh. Dalam Tabel 2.3 di jelaskan tentang Parameter pengoperasiannya Sensor TGS4161

Tabel 2.3 Parameter pengoperasian sensor Gas CO₂

Model number		TGS 4161	
Sensing element type		Solid electrolyte	
Target gases		Carbon dioxide	
Typical detection range		350 ~ 8000 ppm	
Electrical characteristics	Heater resistance	RH	70 ± 7Ω at room temp.
	Heater current	IH	approx. 50mA
	Heater power consumption	PH	approx. 250mW
	Electromotive force	EMF	220~490mV in 350ppm CO ₂
	Sensitivity	ΔEMF	44~72mV EMF(350ppm CO ₂)- EMF(3500ppm CO ₂)
	Heater voltage	VH	5.0 ± 0.2V (DC)
Sensor characteristics	Response time	approx. 1.5 min. (to 90% of final value)	
	Measurement accuracy	approx. ±20% at 1,000ppm CO ₂	
Operating conditions		-10~50°C, 5~95%RH	
Storage conditions		-20~60°C, 5~90%RH (store in moisture proof bag with silica gel)	
Standard test conditions	Test gas condition	CO ₂ in air at 20±2°C, 65±5%RH	
	Circuit condition	VH = 5.0±0.05V DC	
	Conditioning period before test	2 days	

Sumber: datasheet

Informasi tambahan mengenai sensor TGS 4161, yaitu :

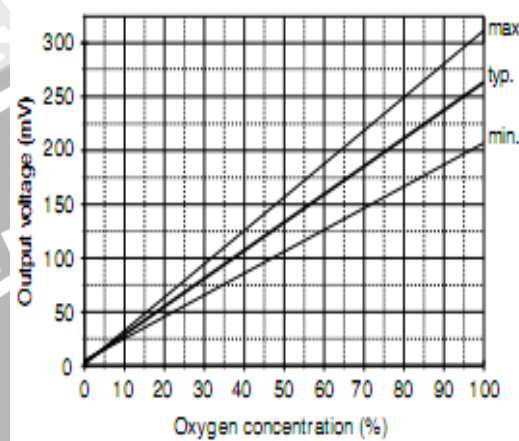
1. Sebelum di rangkai dengan rangkaian sensor ini memerlukan pemanasan awal selama 12 jam.
2. Op amp yang digunakan dalam pengukuran harus memiliki impedansi tinggi (>100GOhm) dan arus bias rendah (<1pA). Rekomendasinya Tlc271
3. Kecepatan Waktu respon pada perubahan gas yaitu sekitar 2.5 menit
4. Jika lama tidak digunakan dan tidak dialiri tenaga respons sensor membutuhkan waktu sekitar 15 menit untuk mencapai performa.
5. Mampu untuk dihidupkan terus menerus hingga mencapai lebih dari 10 tahun pada suhu kamar.
6. Hal yang perlu diperhatikan terhadap sensor ini adalah tidak boleh ada guncangan atau ditekan karena menyebabkan nilai keluaran tidak stabil, Kondensasi (pengembunan) air jangan sampai menyentuh permukaan sensor karena bisa merusak sensor.

2.3 Sensor KE-50

Sensor Gas Oksigen dengan Seri KE-50 adalah sensor yang dikembangkan di Jepang pada tahun 1985 oleh perusahaan Figaro. Fitur-fitur andalan dari sensor ini adalah ketahanan (KE-25 mempunyai kemampuan beroperasi sampai 5 tahun sedangkan KE-50 mempunyai kemampuan beroperasi sampai 10 tahun). Sensor KE-50

ini dilengkapi dengan thermistor yang berfungsi sebagai rangkaian kompensasi suhu, sehingga output sensor tidak berpengaruh pada suhu. Outputnya stabil dan tidak membutuhkan power supply luar sehingga tidak perlu adanya pemanasan.

Dalam Gambar 2.3 dijelaskan Karakteristik Sensor Ke-50. Karakteristik tersebut dapat disimpulkan bahwa Nilai Tegangan keluaran dari sensor KE-50 dipengaruhi konsentrasi Gas O₂ dengan nilai berbanding lurus.



Gambar 2.3 Karakteristik sensor KE-50

Sumber: Datasheet

Sensor KE-50 ini dapat bekerja dengan baik bila kondisi pengoperasiannya dipenuhi. Parameter pengoperasiannya ditunjukkan dalam Tabel.2.4

Tabel 2.4 Parameter pengoperasian sensor Gas O₂

Item		Model	
		KE-25	KE-50
Measurement range		0~100% O ₂	
Accuracy (Note 1)		±1% (full scale)	±2% (full scale)
Operating conditions	Atmospheric pressure	811hPa ~1216hPa	
	Temperature	5~40°C	
	Relative humidity	10 ~ 90%R.H. (no condensation)	
Response time (90%) (Note 4)		14±2 seconds	60±5 seconds
Initial output voltage under standard test conditions		10.0~15.5mV	47~65mV
Standard test conditions	Test gas	21% O ₂	
	Atmospheric pressure	1013±5hPa	
	Temperature	25°C±1°C	
	Relative humidity	60±5%	
Linearity	(V _a -V ₀)/(V ₁₀₀ -V ₀) (Note 2)	0.21±0.02	
Offset voltage	V ₀	≤0.5mV	≤6.0mV
Temperature characteristics (Note 3)	V _H /V _a	0.91~1.09	
	V _L /V _a	0.91~1.09	

Sumber : Datasheet

Sensor ini juga tidak berpengaruh terhadap Gas CO₂ dan berbagai gas yang lain seperti CO, NO, NO₂, H₂S, H₂ dan metana. Disamping itu juga tidak berpengaruh

terhadap suhu dan kelembaban. Aplikasi sensor ini banyak digunakan untuk monitoring gas oksigen misal monitoring pembakaran gas, aplikasi kedokteran, dan lain-lain.

Informasi tambahan mengenai sensor KE-50, yaitu :

1. Sensor ini bekerja dengan baik pada operasi tekanan dan suhu tertentu
 - a) Jika kondisi tekanan kurang dari 507hPa, maka ketahanan sensor tidak tahan lama
 - b) Jika tekanan yang di berikan melebihi dari 1520hPa, maka output sensor menjadi tidak stabil
 - c) Jika suhu sekitar sensor kurang dari -20°C , maka sensor tidak bisa berfungsi
 - d) Jika suhu sekitar sensor melebihi dari 60°C , maka casing ABS resin (bodinya) menjadi rusak yang akhirnya menyebabkan sensor juga akan rusak.
2. Kondensasi (pengembunan) akan menyebabkan output sensor dan kecepatan respon dari sensor menurun sehingga menyebabkan pengukuran tidak teliti. Namun, setelah pengembunan menghilang, sensor akan segera mengembalikan keadaan kembali seperti semula
3. Polaritas elektroda sensor tidak boleh terbalik, tegangan output tidak stabil dan kecepatan respon dari sensor menurun
4. Jika ada larutan elektrolit yang bocor akan menyebabkan sensor akan rusak sehingga untuk merawatnya sensor harus ditempatkan dalam plastik
5. Sensor ini harus di hubungkan dengan rangkaian yang mempunyai impedansi $>1\text{M}\Omega$, karena jika tidak dihubungkan akan menyebabkan kompensasi mengalami gangguan.

2.4 Operasional Amplifier

Op-amp adalah rangkaian elektronik serbaguna yang dirancang dan di kemas khusus, sehingga dengan menambahkan komponen luar sedikit saja, sudah dapat dipakai untuk berbagai keperluan.

Karakteristik terpenting dari sebuah op-amp yang ideal adalah:

- Penguatan loop terbuka amat tinggi
- Impedansi masukan yang sangat tinggi sehingga arus masukan dapat diabaikan
- Impedansi keluaran sangat rendah sehingga keluaran penguat tidak terpengaruh oleh pembeban.

Pada op-amp terdapat satu terminal keluaran, dan dua terminal masukan. Terminal masukan yang diberi tanda (-) dinamakan terminal masukan pembalik

(inverting), sedangkan terminal masukan yang diberi (+) dinamakan terminal masukan bukan pembalik (noninverting).

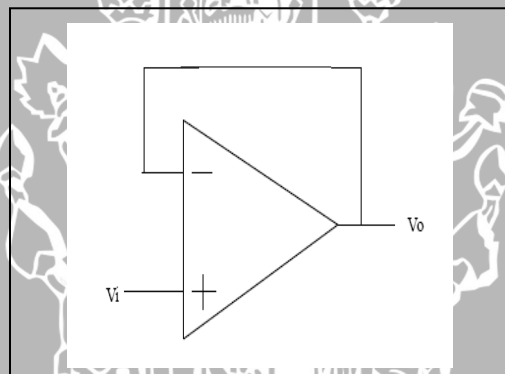
2.4.1 Penguat Penyangga (Buffer)

Penguat Penyangga merupakan penguat yang kegunaannya untuk penguatan tanpa pembalikan fasa. Artinya sinyal output penguat sefasa dengan sinyal input penguat yang diberikan dan merupakan umpan balik negatif 180° . Seperti yang ditunjukkan dalam Gambar 2.4 bahwa tahanan umpan baliknya tidak ada sehingga seluruh tegangan keluaran akan diumpan balikkan ke masukan. Penguatan tegangan dari penguat penyangga ini sama dengan satu, dimana :

$$A = \frac{V_o}{V_i} = 1 \quad \dots\dots\dots(2.1)$$

V_i : Tegangan Masukan

V_o : Tegangan Keluaran



Gambar 2.4 Penguat Penyangga
Sumber : National semiconductor, 1995

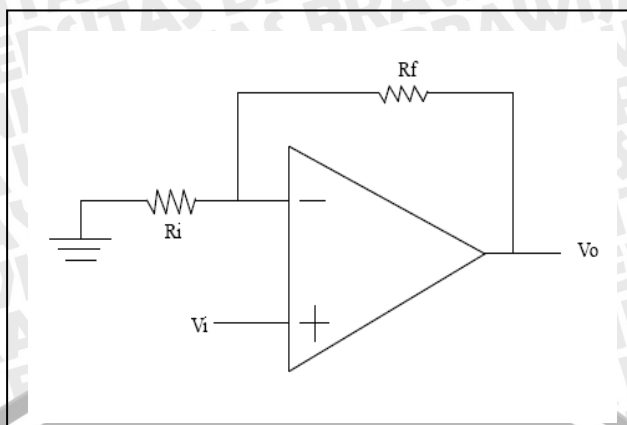
2.4.2 Penguat Tak Membalik (Non-Inverting Amplifier)

Penguat tak membalik (non-inverting amplifier) adalah penguat yang sinyal inputnya di berikan ke masukan non inverting (terminal positif). Tegangan keluaran diumpan balikkan ke masukan inverting. Rangkaian dari penguat non-inverting ditunjukkan dalam Gambar 2.5.

Besarnya penguatan untuk penguat tak membalik adalah sebagai berikut :

$$V_o = \frac{R_f + R_i}{R_i} V_i \quad \dots\dots\dots(2.2)$$

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



Gambar 2.5. Penguat Non-Inverting

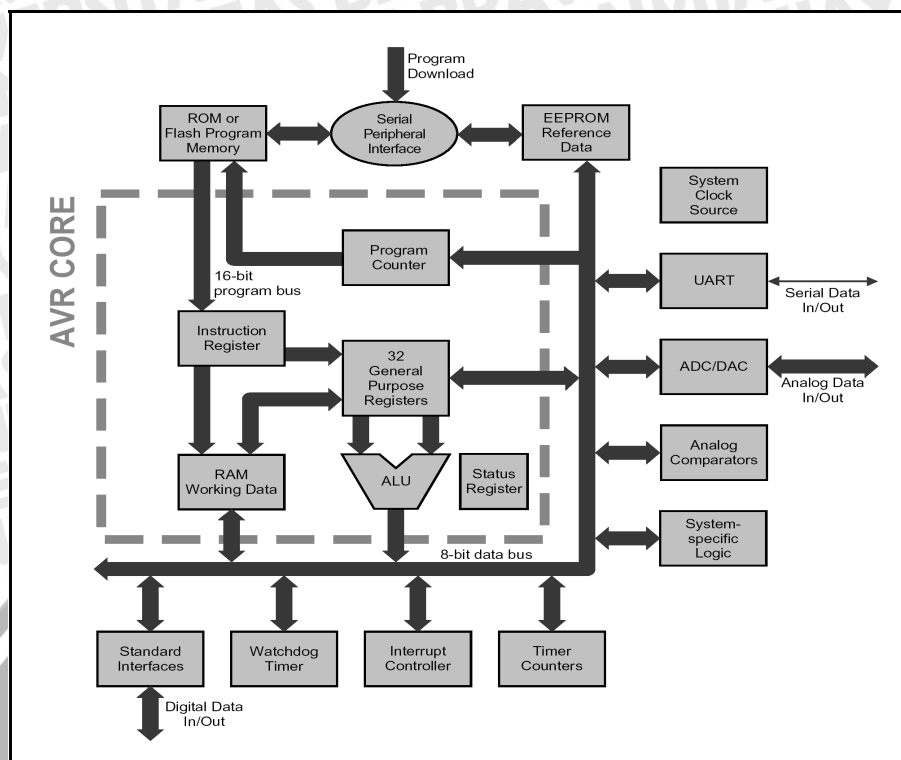
Sumber : Malvino, 1987 : 86

2.5 Mikrokontroler ATmega8P

Mikrokontroler Atmega8P merupakan salah satu bagian dari keluarga AVR (At and Vegard's Risc prosessor) yang diproduksi Atmel dengan menggunakan arsitektur RISC (Reduced Instruction Set Computing) 8 bit. AVR yang pertama kali diperkenalkan pada tahun 1996 ini didesain dengan mengkombinasikan arsitektur RISC, memori flash internal dan jumlah register serbaguna yang besar (32 buah) untuk memperoleh ukuran kode program, kinerja, dan konsumsi daya yang optimal. Beberapa kelebihan yang dimiliki Mikrokontroler keluarga AVR adalah kemampuan eksekusi dalam satu siklus clock serta arsitektur AVR yang dirancang untuk bekerja secara efisien menggunakan bahasa tingkat tinggi C.

2.5.1 Arsitektur AVR

AVR menggunakan konsep arsitektur Harvard dengan memori dan bus terpisah untuk data dan program. Lebar bus program pada AVR adalah 16 bit sedangkan lebar bus data adalah 8 bit. Memori program dieksekusi dengan pipeline satu tingkat, dimana saat instruksi sedang dieksekusi, instruksi yang berikutnya dibaca dari memori program. Konsep ini memungkinkan instruksi untuk dieksekusi dalam tiap-tiap siklus clock. Arsitektur AVR seperti terlihat pada Gambar 2.6.



Gambar 2.6. Arsitektur AVR

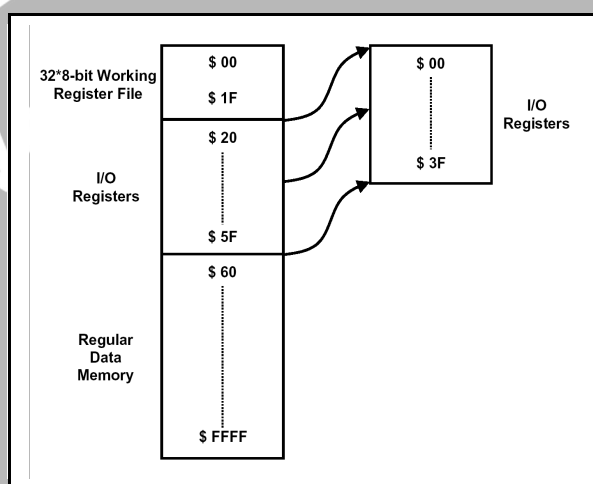
Sumber : Atmel, 2002 : 5

AVR memiliki jumlah register yang relatif besar untuk ukuran Mikrokontroler 8 bit, yaitu 32 buah general purpose registers. AVR tidak memiliki accumulator seperti yang dimiliki sebagian besar mikroprosesor/Mikrokontroler. Seluruh general purpose register terhubung ke ALU (Arithmetic Logic Unit) sehingga operasi ALU dapat dilaksanakan dengan menggunakan general purpose registers sebagai operand. Hal ini yang membuat AVR begitu cepat dalam mengeksekusi instruksi. Dalam satu siklus clock, terdapat dua register independen yang dapat diakses oleh suatu instruksi. Teknik yang digunakan adalah fetch during execution atau memegang sambil mengerjakan. Hal ini berarti, dua operand dibaca dari dua register, dilakukan eksekusi operasi dan hasilnya disimpan kembali ke dalam salah satu register, semuanya dilakukan dalam satu siklus clock (Pratomo A., 2005). Dalam skala operasi yang relatif besar, operasi ALU dapat dijalankan dengan lebih cepat. Semua general purpose registers juga dapat digunakan untuk mengakses data dari dan ke memori dengan instruksi load dan store.

Enam dari 32 register dapat digunakan sebagai pasangan register 16 bit. Pasangan register 16 bit dapat digunakan sebagai pointer register untuk pengalamatan area data sehingga memungkinkan perhitungan alamat menjadi lebih efisien. Ketiga

register tersebut adalah register X (R26 dan R27), Y (R28 dan R29), dan Z (R30 dan R31).

Peta memori data AVR dapat dilihat pada Gambar 2.7. Memori data dibagi menjadi 4 bagian. 32 alamat paling bawah (0000 – 001F) ditempati oleh general purpose registers. 64 alamat berikutnya (0020 – 005F) ditempati oleh register I/O yang mengatur piranti CPU seperti register kontrol, timer/counter, ADC, dan fungsi I/O lainnya. Alamat berikutnya digunakan oleh SRAM internal dan eksternal. Perlu dicatat, ukuran SRAM internal tidak sama untuk masing-masing tipe AVR dan tidak semua tipe memiliki SRAM internal atau SRAM eksternal.



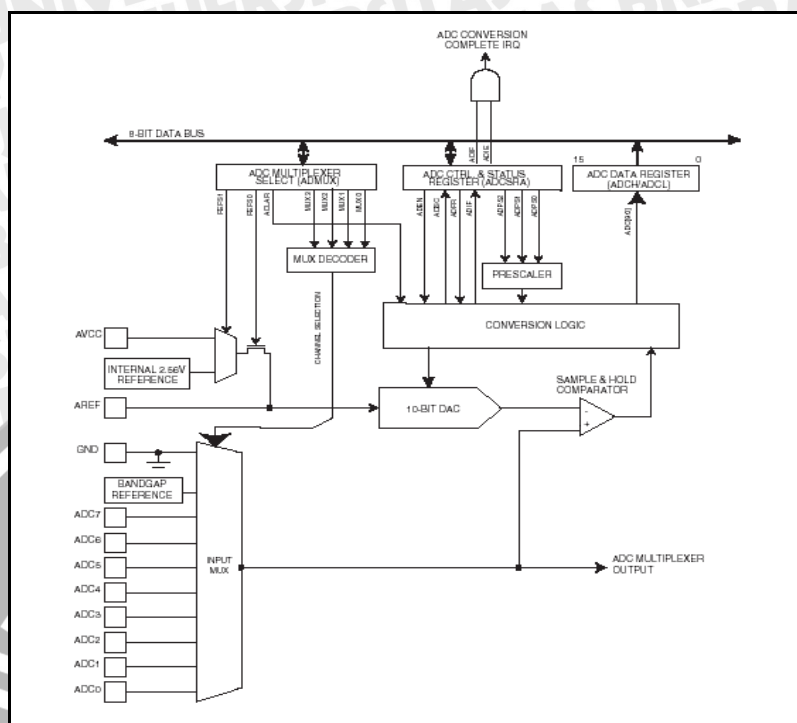
Gambar 2.7. Peta Memori AVR
Sumber : Atmel, 2002 : 6

Selama interupsi dan pemanggilan subroutine, alamat pada Program Counter (PC) disimpan pada stack. Pada tipe AVR yang memiliki SRAM internal atau eksternal, stack dialokasikan di SRAM sehingga ukuran stack hanya terbatas oleh total ukuran dan pemakaian SRAM. Semua program harus menginisialisasi Stack Pointer (SP) di dalam rutin reset, sebelum subroutines atau interrupt dieksekusi. Sedangkan tipe AVR yang tidak memiliki SRAM, alamat pada PC disimpan di stack hardware yang memiliki ukuran terbatas.

2.5.2 Analog to Digital Converter (ADC)

AVR memiliki beberapa jenis peripheral internal, diantaranya komparator analog, timer/counter, UART, watchdog timer, EEPROM dan ADC/DAC. Dalam pembuatan alat ukur, ADC/DAC adalah peripheral yang sering di gunakan. AVR Atmega8P memiliki fasilitas ADC 10 bit yang terhubung dengan 6 saluran multiplekser analog yang membolehkan setiap pin pada port C untuk digunakan sebagai input untuk ADC. ADC terdiri dari satu Sample/Hold Amplifier yang memastikan bahwa tegangan input

ke ADC ditahan pada level konstan selama konversi. Blok diagram dari ADC seperti yang ditunjukkan pada Gambar 2.8.



Gambar 2.8. Blok Diagram ADC
Sumber: Atmel, 2002: 197

ADC mempunyai 2 pin tegangan analog yang terpisah yaitu AVCC dan AGND. AGND harus dihubungkan ke GND dan tegangan pada AVCC tidak boleh beda lebih dari $\pm 0.3V$ dari VCC. Tegangan referensi eksternal harus dikenakan ke pin AREF. Tegangan ini harus berada dalam range AGND – AVCC. ADC bisa beroperasi dalam 2 mode yaitu Single Conversion dan Free Run Mode. Pada Single Conversion Mode, setiap konversi harus diinisialisasi oleh pengguna. Pada Free Run Mode, ADC secara konstan menyampling dan mengupdate ADC Data Register.

ADC aktif dengan memberikan logika 1 ke bit ADC Enable, yaitu ADEN pada ADCSR. Konversi pertama dimulai setelah mengaktifkan ADC, dimana waktu yang diperlukan untuk melakukan konversi sebesar 13 siklus clock ADC. Bit ini tetap high selama konversi berlangsung dan akan diset nol oleh hardware ketika konversi lengkap. Jika suatu saluran data berbeda terpilih sedangkan suatu konversi masih dalam proses, ADC akan menyelesaikan konversi yang sekarang sebelum melakukan perubahan saluran. Hasil konversi ADC akan menghasilkan 10-bit data pada dua register data, yaitu ADCH dan ADCL, dimana data hasil konversi harus dibaca untuk mendapatkan hasil ketika konversi lengkap. Logika perlindungan khusus digunakan untuk

memastikan bahwa isi dari register data ADCH dan ADCL telah diakses secara tepat ketika konversi telah selesai.

Mekanisme pengaksesan hasil konversi adalah sebagai berikut: ketika pembacaan data, ADCL harus dibaca pertama kali. Sekali ADCL dibaca, akses ke register data diblok. Ini berarti bahwa jika ADCL telah dibaca dan konversi selesai sebelum ADCH dibaca, tidak ada satu register pun yang diupdate dan hasil dari konversi hilang. Ketika ADCH dibaca, akses ADC ke register ADCH dan ADCL di enable ulang. ADC pada Atmega8P memiliki interrupt sendiri, yaitu ADIF, yang bisa dipicu ketika konversi selesai. Akses ADC ke register data dilarang untuk melakukan pembacaan dari ADCH menuju ADCL, apabila hal ini dilakukan maka data hasil konversi menjadi tidak valid lagi. ADC memiliki fasilitas prescaler, yang berfungsi membagi sistem clock menjadi frekuensi clock ADC yang mudah diterima. ADC Atmega8P dapat menerima frekuensi clock input dalam range 50 kHz - 200 kHz.

Penggunaan frekuensi input yang semakin tinggi akan menghasilkan tingkat akurasi yang semakin rendah. Bit ADPS0 - ADPS2 pada ADCSR digunakan untuk menghasilkan frekuensi clock input ADC yang diperlukan dari frekuensi XTAL di bawah 100 kHz. Prescaler mulai menghitung saat ADC diaktifkan dengan mengeset bit ADEN pada ADCSR. Prescaler dijaga tetap bekerja selama bit ADEN diset dan secara kontinyu direset ketika ADEN rendah. Ketika penginisialisasian konversi dengan mengeset bit ADSC pada ADCSR, konversi mulai pada tepi naik dari siklus clock ADC.

Pada kenyataannya sample/hold memerlukan 1.5 siklus clock ADC setelah konversi dimulai. Hasil konversi siap dan ditulis pada register hasil ADC setelah 13 siklus clock ADC. Pada single conversion mode, ADC memerlukan lebih dari satu siklus clock sebelum konversi yang baru bisa dimulai lagi. Jika ADSC diset high pada periode ini, ADC akan mulai konversi baru dengan seketika. Pada free run mode, konversi yang baru akan dimulai dengan tiba-tiba setelah hasil konversi ditulis pada register hasil ADC. Hasil penggunaan free run mode dan frekuensi clock ADC 200 kHz memberikan konversi waktu terendah, 65 ms, ekuivalen dengan 15.4 kSPS.

2.5.3 EEPROM

ATmega8L mempunyai EEPROM dengan kapasitas 512 *byte* yang dapat dibaca dan ditulis sampai 100.000 kali. Pengaksesan EEPROM dapat dilakukan dengan melibatkan *EEPROM Address Register (EEAR)*, *EEPROM Data Register (EEDR)* dan *EEPROM Control Register (EECR)*.

EEPROM Address Register terdiri dari 2 buah register, yaitu *EEARL* dan *EEARH* yang berfungsi untuk pengalamatan data yang akan dibaca/ditulis dari/ke *EEPROM*. Konfigurasi *EEPROM Address Register* seperti ditunjukkan pada Gambar 2.9.

Bit	15	14	13	12	11	10	9	8	
	—	—	—	—	—	—	—	EEAR8	EEARH
	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
Read/Write	R	R	R	R	R	R	R	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	X	
	X	X	X	X	X	X	X	X	

Gambar 2.9 *EEPROM Address Register*

Sumber : Atmel, 2004: 18

Bit 15 .. 9 - *Reserved bit*, pada bit-bit ini merupakan *invalid* bit dan akan selalu terbaca logika rendah(0). Bit 8 .. 0 - *EEPROM Address*, pada bit-bit ini, register berfungsi untuk pengalamatan *EEPROM* dengan kapasitas 512 *byte* yang dapat dialamati dari alamat 0 sampai 511.

EEPROM Data Register terdiri dari 8 bit yang berfungsi sebagai register data sementara yang dibaca dari *EEPROM* atau data yang akan ditulis ke *EEPROM* dengan alamat tertentu. *EEPROM data register* berkapasitas 8 bit seperti ditunjukkan dalam Gambar 2.10.

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.10 *EEPROM Data Register*

Sumber : Atmel, 2004: 18

Bit 7 .. 0 – *EEDR7..0* : *EEPROM Data Register*, pada mode penulisan, *Register EEDR* berisikan data yang akan dituliskan pada *EEPROM* dengan alamat yang terdapat *EEPROM address register*. Sedangkan pada mode pembacaan, hasil dari mode pembacaan pada alamat yang berada pada *EEPROM address register* akan terdapat pada register ini.

EEPROM Control Register seperti yang ditunjukkan pada Gambar 2.11 diantaranya berfungsi untuk menentukan *access* yang akan dilakukan terhadap EEPROM, seperti proses baca dan proses tulis dari/ke EEPROM.

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	EERIE	EEMWE	EEWE	EERE	EECR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	X	0	

Gambar 2.11 EEPROM Control Register

Sumber : Atmel, 2004: 18

Bit 7 .. 4 : *Reserved bit*, pada bit-bit ini merupakan invalid bit dan akan selalu terbaca logika rendah(0). Bit 3 : *EEPROM Ready Interrupt Enable*, jika pada bit ini diset 1, maka mode *Ready Interrupt Enable* akan aktif. Bit 2 : *EEMWE : EEPROM Master Write Enable*, bit ini digunakan untuk menentukan mode penulisan selain *EEWE* pada EEPROM. Jika diset 1 maka dengan menyeting *EEWE* dengan 4 *clock* akan melakukan penulisan pada alamat yang dituju. Jika diset 0 maka konfigurasi *EEWE* tidak berfungsi. Bit 1 : *EEWE : EEPROM Write Enable*, jika diset 1 maka EEPROM berada pada mode penulisan. Bit 0 : *EERE : EEPROM Read Enable*, jika diset 1 maka EEPROM berada pada mode pembacaan.

Proses penulisan data ke EEPROM dapat dilakukan dengan prosedur sebagai berikut:

1. Pastikan *EEWE* berlogika nol.
2. Pastikan *SPMEN* pada *SPMCR* berlogika nol.
3. Tuliskan data alamat baru ke *EEAR* jika diinginkan.
4. Tuliskan data baru yang akan disimpan ke EEPROM pada *EEDR* jika diinginkan.
5. Tuliskan logika satu pada *EEMWE* pada saat *EEWE* berlogika nol.
6. Tuliskan logika satu pada *EEWE* setelah empat *clock cycle* dari penulisan logika satu pada *EEMWE*.

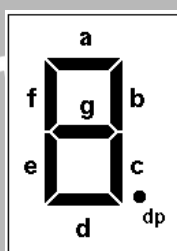
Sedangkan proses pembacaan data dari EEPROM dapat dilakukan dengan prosedur sebagai berikut:

1. Pastikan *EEWE* berlogika nol.
2. Pastikan *SPMEN* pada *SPMCR* berlogika nol.
3. Tuliskan data alamat baru ke *EEAR* jika diinginkan.
4. Tuliskan logika satu ke *EERE* pada *EECR*.
5. Ambil data yang ada pada *EEDR*.

Penggunaan *memory non-volatile* pada mikrokontroler AVR ini bertujuan untuk menyimpan informasi yang berupa kadar Gas Oksigen dan Karbondioksida sekaligus menampilkannya kembali pada saat dibutuhkan.

2.6 Seven Segment

Peralatan keluaran yang sangat umum digunakan untuk menampilkan bilangan desimal adalah peraga Senen Segmen. Suatu peraga tujuh segmen terdiri dari tujuh buah LED (Light Emitting Dioda) yang tersusun dari a sampai g seperti yang ditunjukkan dalam Gambar 2.12.



Gambar 2.12 LED 7 Segmen
Sumber: Elektuur, 1985:190

Ada dua macam hubungan pada display 7-segment, yaitu :

1. Common anoda

Pada hubungan ini semua anoda dihubungkan menjadi satu ke positif catu daya dan katodanya difungsikan sebagai masukan.

2. Common katoda

Pada hubungan ini semua katoda dihubungkan menjadi satu ke ground dan anodanya sebagai masukan.

Komponen Seven segmen yang digunakan dalam perancangan ini merupakan common katoda.

BAB III

METODOLOGI

Penyusunan skripsi ini didasarkan pada masalah yang bersifat aplikatif, yaitu perancangan dan perealisasiannya alat agar dapat menampilkan unjuk kerja sesuai dengan yang direncanakan dengan mengacu pada rumusan masalah. Data dan spesifikasi komponen yang digunakan dalam perancangan merupakan data sekunder yang diambil dari buku data komponen elektronika. Pemilihan komponen berdasarkan perancangan dan disesuaikan dengan komponen yang ada di pasaran.

Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang akan dibuat adalah sebagai berikut:

3.1 Studi Literatur

Studi literatur dalam penulisan ini dimaksudkan untuk mempelajari teori komponen dan atau rangkaian yang digunakan dalam penulisan ini. Selain itu juga mempelajari secara global beberapa teori pendukung yang menunjang penyusunan Tugas Akhir ini. Beberapa hal yang dipelajari dalam studi literatur adalah sebagai berikut:

1. Mencari studi literatur secara global tentang proses fermentasi Biji kakao, perubahan apa saja yang terjadi, dan hal-hal yang mempengaruhi proses fermentasinya
2. Melakukan studi literatur tentang sensor yang dapat mendeteksi gas Karbondioksida dan Oksigen, mempelajari datasheetnya dan bagaimana cara pengoperasiannya
3. Melakukan studi literatur tentang rangkaian pengkondisi Sinyal dengan menggunakan op-Amp yang bertujuan, dan mengetahui karakteristik dan penggunaannya
4. Melakukan studi literatur tentang sistem mikrokontroler ATmega8P serta beberapa peripheral yang akan digunakan dalam perealisasiannya alat agar dapat mengetahui cara kerja mikrokontroler ATmega8P serta bahasa pemrograman yang akan dipakai
5. Melakukan studi literatur tentang Seven Ssegment untuk mempelajari bagaimana cara menampilkan data dari mikrokontroler ke *display* peraga.

3.2 Perancangan Alat

Agar perancangan dan perealisasiian alat berjalan secara sistematis maka perlu dirancang diagram blok yang menjelaskan sistem yang dirancang secara garis besar. Pada setiap perancangan, pertama kali dilakukan penentuan spesifikasi rangkaian yang akan dibuat, kemudian dibuat blok diagram fungsional. Perancangan rangkaian dilakukan pada setiap blok diagram untuk mempermudah perancangan dan dilakukan perhitungan dalam menentukan besarnya komponen dan jenis komponen yang digunakan. Selanjutnya masing-masing blok dihimpun menjadi suatu sistem lengkap. Perancangan alat didasarkan pada teori yang ada dan data-data komponen yang dikeluarkan oleh industri pembuat komponen yang bersangkutan. Pemilihan komponen berdasarkan perancangan dan disesuaikan dengan komponen yang ada di pasaran.

3.3 Pembuatan Alat

Pembuatan alat dibagi menjadi dua yaitu pembuatan perangkat keras dan pembuatan perangkat lunak. Pembuatan perangkat keras meliputi pembuatan PCB dan pemasangan serta penyolderan komponen agar menjadi satu rangkaian. Pembuatan perangkat lunak hanya meliputi *firmware* untuk mikrokontroler yang digunakan.

3.4 Pengujian Alat

Untuk mengetahui kinerja piranti apakah sesuai dengan yang direncanakan maka dilakukan pengujian rangkaian. Pengujian dilakukan pada masing-masing blok dan secara keseluruhan. Pengujian alat meliputi:

1. Pengujian Sensor

Pengujian ini dilakukan untuk mengetahui unjuk kerja sensor TGS4161 dan KE-50. Pengujian sensor dilakukan pada saat mengukur kadar Gas Oksigen dan Karbondioksida di udara bebas yaitu 20-21% oksigen dan 370-380ppm untuk Gas Karbondioksida. Tegangan keluaran dari sensor ini di bandingkan dengan nilai tegangan output sensor yang di jelaskan dari datasheet. Dari perbandingan tersebut, dapat diketahui berapa besar penyimpangannya.

2. Pengujian Rangkaian Pengkondisi Sinyal

Pengujian ini dilakukan untuk menghubungkan antara rangkaian sensor dengan port ADC Mikrokontroler ATmega8P sehingga mudah di olah dan diproses.

3. Pengujian ADC internal mikrokontroler ATmega8P

Pengujian ini dilakukan agar output dari Rangkaian Pengkondisi sinyal yang masih berupa besaran analog, dapat dikondisikan menjadi besaran digital yang mudah diproses oleh Mikrokontroler.

4. Pengujian Display

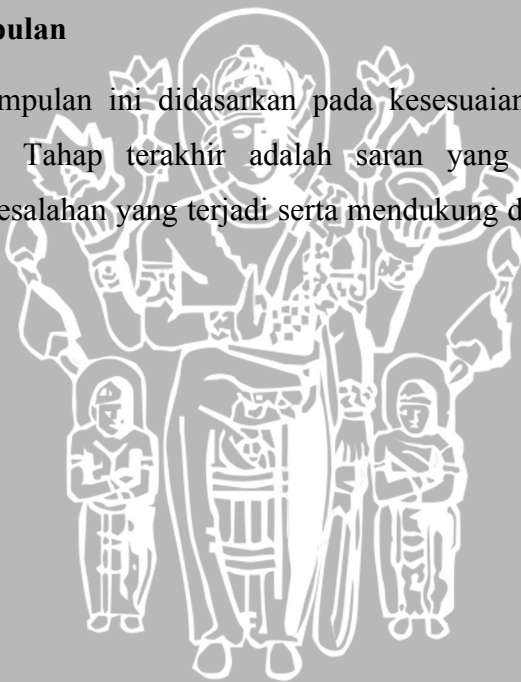
Pengujian ini dilakukan untuk mengetahui unjuk kerja dari display berupa seven segment. Pengujian dilakukan dengan cara memberinya program sederhana untuk menampilkan nilai tertentu.

5. Pengujian keseluruhan sistem

Pengujian keseluruhan sistem dimaksudkan untuk memastikan apakah alat dapat bekerja dengan baik. Pengujian alat dilakukan di Pusat Penelitian Kopi dan Kakao Indonesia, Jember, Indonesia.

3.5 Pengambilan Kesimpulan

Pengambilan kesimpulan ini didasarkan pada kesesuaian antara perancangan dengan hasil pengujian. Tahap terakhir adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi serta mendukung dalam penyempurnaan penelitian ke depan.



BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

Bab ini membahas tentang perancangan dan pembuatan Alat Monitoring Kadar Gas O₂ dan CO₂ sebagai Alat Bantu Proses Fermentasi Biji Kakao yang meliputi perancangan perangkat keras dan perangkat lunak. Perancangan perangkat keras terdiri dari: rangkaian sensor, rangkaian penguat diferensial dan Penguat Tak Membalik, sistem mikrokontroler yang diantarmukakan dengan rangkaian pengkondisi sinyal sensor dan peripheral pendukung yang berupa tiga buah tombol *pushbutton* serta Display yang berupa seven segment. Sedangkan perancangan perangkat lunak akan dibahas tentang pembuatan *flowchat* dan program pada ATmega8P.

4.1 Spesifikasi Alat

Hal pertama yang perlu dilakukan sebelum melakukan perancangan dan pembuatan alat adalah penentuan spesifikasi alat yang akan dibuat. Spesifikasi alat yang akan direncanakan adalah sebagai berikut:

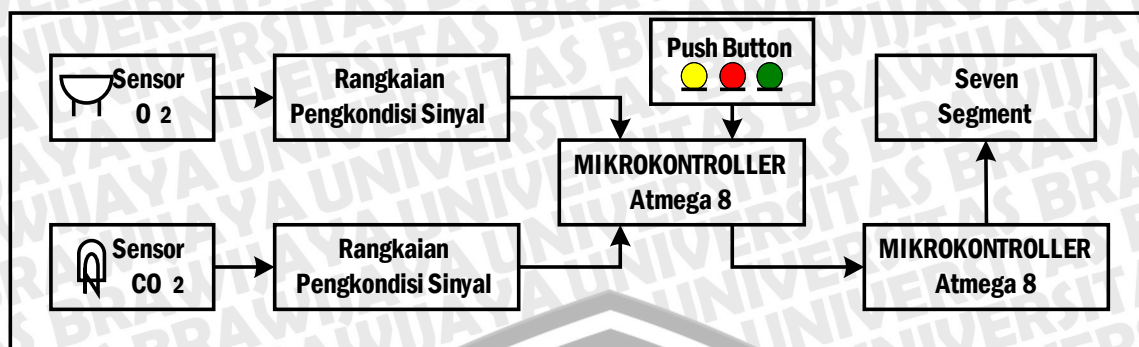
1. Alat menggunakan 2 buah mikrokontroler ATmega8 yang berfungsi sebagai pengontrol utama dan untuk memproses data ke display.
2. Kemampuan Sensor CO₂ dapat mendeteksi Gas antara CO₂ antara 370 – 5000ppm dan Sensor O₂ dapat mendeteksi 0 – 100 % oksigen. Ini dikarenakan dalam proses fermentasi biji kakao menghasilkan gas karbondioksida maksimal 5000ppm, sedangkan Oksigen yang di butuhkan dari kadar gas Oksigen di udara sampai 80%
3. Untuk pengopeasian alat menggunakan 3 buah tombol *push button* sebagai power, tombol Up dan tombol down
4. Data yang di ukur akan di tampilkan ke Seven-Segment dalam bentuk angka desimal.
5. Catu daya yang digunakan adalah 5V DC dengan menggunakan adaptor 12V
6. Resolusi Alat, untuk pengukuran Gas Karbondioksida 4,5ppm sedangkan untuk pengukuran Gas Oksigen, resolusinya 0,1 %

4.2 Blok Digram Sistem

Sebelum perancangan alat, terlebih dahulu harus di buat blok diagram yang memudahkan dalam pembuatan alat. Blok digram ini di buat bertujuan untuk

mengetahui perangkat keras yang digunakan dan cara kerja dari sistem. Blok Diagram Alat di tunjukkan dalam Gambar 4.1.





Gambar 4.1 Blok Diagram Alat
Sumber : Perancangan

4.3 Prinsip Kerja Alat

Alat ini di buat untuk memonitoring Kadar Gas Oksigen yang di butuhkan dan Karbondioksida yang di hasilkan selama proses fermentasi yang dilakukan 5 hari. Kadar gas tersebut akan di deteksi oleh sensor yang berfungsi merubah besaran Gas menjadi besaran elektrik berupa tegangan. Karena tegangan yang dikeluarkan sensor relative kecil, maka diperlukan Rangkain pengkondisi Sinyal dengan nilai penguatan tertentu sesuai tegangan batas tegangan referensi ADC internal dari Mikrokontroller ATmega8. Kemudian tegangan tersebut di ubah menjadi sinyal digital agar mudah diolah oleh Mikrokontroller ATmega8. setelah itu, ditampilkan ke seven segment dalam bentuk desimal setelah di olah dengan Mikrokontroller ATmega8 yang ke-dua. Agar nilai yang ditampilkan sesuai permintaan, maka diperlukan tombol push button yang berfungsi untuk mengatur nilai yang ingin di lihat. Nilai pengukuran untuk menampilkan Oksigen dalam satuan persen, sedangkan untuk Karbondioksida dalam satuan ppm.

Cara pengoperasian Alat adalah sebagai berikut :

1. Tekan tombol power untuk menghidupkan dan mematikan Alat
2. Alat ini dapat menampilkan 2 Besaran fisis yaitu, Kadar Oksigen (O_2) dalam satuan % dan Kadar Karbon dioksida (CO_2) dalam satuan ppm (part per million)
3. Cara menampilkan Nilai besaran fisis tersebut :
 - a) Tombol Hijau berfungsi untuk menampilkan Nilai-nya (Tombol Enter)
 - b) Tombol Kuning berfungsi untuk memilih pilihan besaran yang akan ditampilkan ke arah atas yaitu : All, O_2 , CO_2
 - c) Tombol Merah berfungsi untuk memilih pilihan besaran yang akan ditampilkan ke bawah atas yaitu : CO_2 , O_2 , All

4.4 Perancangan Perangkat Keras

Bagian ini menguraikan rincian perancangan perangkat keras yang meliputi:

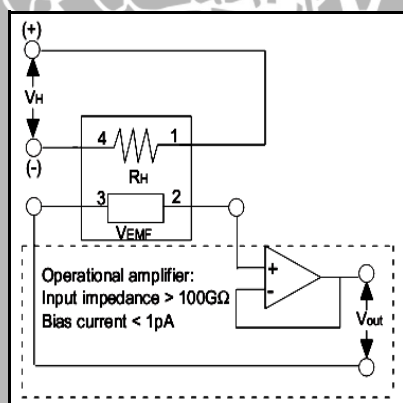
1. Perancangan Rangkaian sensor CO₂ dan Sensor O₂
2. Perancangan rangkaian pengkondisi sinyal.
3. Perancangan sistem mikrokontroler Atmega8P.
4. Perancangan rangkaian Display

i. Perancangan Rangkaian Sensor

Perancangan ini sensor ini bertujuan untuk mengoperasikan sensor agar dapat di gunakan dalam pengukuran besaran yang terdeteksi. Perancangan ini mengacu pada datasheet sensor sesuai pengoperasian sensor.

4.4.1.1 Perancangan Rangkaian Sensor CO₂

Sensor CO₂ yang di gunakan adalah TGS4161. keluaran dari sensor ini berupa tegangan. Datasheet menjelaskan bahwa sensor ini membutuhkan rangkaian buffer yang mempunyai nilai impedansi masukan >100Gohm, dan arus bias yang rendah yaitu < 1pA agar nilai Outputnya stabil. Yang direkomendasikan adalah Op-Amp Tlc271 sebagai Rangkaian buffernya. Gambar 4.2 menjelaskan tentang Perancangan Rangkaian Sensor CO₂.



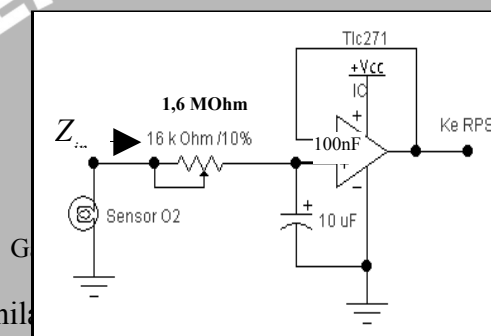
Gambar 4.2 Perancangan Rangkaian Sensor CO₂
Sumber : Datasheet

Keterangan :

1. Pin 1 : Heater (+) dihubungkan dengan catu daya dengan nilai $V_H=5\pm 0,05$ VDC, pemanasan awal dilakukan selama 12 jam
2. Pin 2 : Sensor Elektroda (+) di hubungkan dengan input Rangkaian Pengkondisi Sinyal
3. Pin 3 : Sensor Elektroda (-) di hubungkan dengan Ground
4. Pin 4 : Heater (-) di hubungkan dengan Ground

4.4.1.2 Perancangan Rangkaian Sensor O₂

Untuk Perancangan Rangkaian Sensor O₂, yang di gunakan adalah KE-50. ari Datasheet di jelaskan bahwa sensor akan menghasilkan tegangan Output yang tidak stabil apabila sensor terkena bahan pelarut organic seperti toluene, benzene, sylen, acetone, methyl ethyl keton, methyl chloride, kerosin, gasoline, dan gas oli. Untuk mengatasinya diperlukan rangkaian filter. Yang direkomendasikan adalah rangkaian Low pass filter dengan frekuensi kritisnya 1Hz. Disamping itu, sensor juga harus di hubungkan dengan rangkaian yang mempunyai impedansi masukan >1MΩ. Karena jika tidak, maka rangkaian kompensasi suhu yang berupa thermistor tidak berfungsi. Dalam hal ini Op Amp yang digunakan untuk rangkaian buffer dengan Op-Amp Tlc271 dikarenakan impedansinya 10¹² Ω. Gambar 4.3 menjelaskan tentang Perancangan Rangkaian Sensor O₂.



Untuk Mencari nilai f_c digunakan persamaan :

$$f_c = \frac{1}{2\pi RC} \quad (4.1)$$

Diketahui : $f_c = 1\text{Hz}$ (direkomendasikan dari datasheet)

Dalam perancangan ini, nilai C yang dipilih = 100nF, dengan alasan :

1. Agar tidak menarik arus yang terlalu besar
2. Karena sensor harus dihubungkan dengan impedansi masukan >1MΩ

Maka nilai R didapatkan :

$$1 = \frac{1}{2 \times 3,14 \times 1 \times 10^{-7} \times R}; \text{ sehingga } R = \frac{1}{0,000000628} = 1592356,688\Omega \approx 1,6\text{M}\Omega$$

Untuk mencari Impedansi Masukan Z_{in} (frekuensi pada arus DC = 0), sebagai berikut:

$$Z_{in} = Z_1 + (Z_2 // Z_3) = R_1 + (X_c // R_2) = 1,6 \times 10^6 + \left(\frac{X_c * R_2}{X_c + R_2} \right) = 1,6 \times 10^6 + 1,6 \times 10^6 = 3,2\text{M}\Omega$$

Ket: Z_1 = impedansi Resistor; $Z_2 = X_c = \frac{1}{2\pi fC}$ = impedansi kapasitor; Z_3 = impedansi Op-Amp

ii. Perancangan Rangkaian Pengkondisi Sinyal

Tegangan keluaran Sensor masih sangat kecil di butuhkan penguat yang disebut Rangkaian pengkondisi Sinyal. Op Amp sebagai komponen utamanya yang di gunakan adalah Tlc271. Menggunakan Op-Amp jenis ini dengan alasan karena rekomendasi dari sensornya, dan juga menurut datasheet Op-Amp ini mempunyai Arus bias yang rendah dan impedansi masukan yang tinggi sehingga tidak membebani sensor.

1. Perancangan RPS Sensor CO₂

Rangkaian pengkondisi sinyal merupakan rangkaian yang menghubungkan antara keluaran sensor TGS4161 yang telah dikonversikan menjadi tegangan dengan masukan ADC dari Atmega8P. Rangkaian ini berfungsi untuk mengatur tingkat sensitivitas dan akurasi dari alat Monitoring Kadar Gas Karbondioksida. Tegangan keluaran yang dihasilkan oleh sensor berkisar antara 254mV – 468mV akan dimanipulasi sehingga menjadi tegangan antara 0 V – 2,56 V (tegangan 2,56 V merupakan tegangan referensi dari ADC pada Atmega8P yang dipilih oleh perancang).

Agar Tegangan 468mV bernilai 0V dan tegangan 254mV bernilai 2,56V, maka tegangan referensi serta penguatan yang di butuhkan oleh penguat pengurang dapat dicari dengan persamaan, berikut :

$$\frac{V_{out} - V_{ref}}{R_f} = \frac{V_{ref} - V_{in}}{R_i} \quad (4.2)$$

Pada saat : $V_{in}=0,468$, $V_{out} = 0$ V, maka :

$$\begin{aligned} \frac{0 - V_{ref}}{R_f} &= \frac{V_{ref} - 0,468}{R_i} \\ \frac{R_f}{R_i} &= \frac{- V_{ref}}{V_{ref} - 0,468} \end{aligned} \quad (4.3)$$

Pada saat : $V_{in}=0,254$ V, $V_{out} = 2,56$ V, maka:

$$\begin{aligned} \frac{2,56 - V_{ref}}{R_f} &= \frac{V_{ref} - 0,254}{R_i} \\ \frac{R_f}{R_i} &= \frac{2,56 - V_{ref}}{V_{ref} - 0,254} \end{aligned} \quad (4.4)$$

Dari persamaan di atas, persamaan 4.2 dimasukkan ke dalam persamaan 4.3, maka:

$$\frac{- V_{ref}}{V_{ref} - 0,468} = \frac{2,56 - V_{ref}}{V_{ref} - 0,254}$$

$$(-V_{ref}^2 + 0,254V_{ref}) = -V_{ref}^2 + 3,028V_{ref} - 1,19808$$

$$V_{ref} = \frac{1,19808}{2,774} = 0,432V \quad (4.5)$$

Karena $V_{ref} = 0,432 V$, maka :

$$\begin{aligned} \frac{R_f}{R_i} &= \frac{-V_{ref}}{V_{ref} - 0,468} = \frac{-0,432}{0,432 - 0,468} \\ &= \frac{-0,432}{-0,036} = 12 \text{ kali} \end{aligned} \quad (4.6)$$

Jika di tetapkan $R_i = R_2 = 1k\Omega$, maka $R_f = R_4 = 12 k\Omega$

Untuk mendapatkan $V_{ref} = 0,432V$ dapat dilakukan dengan menggunakan rangkaian pembagi tegangan, dengan nilai $V_{DD} = 5V$ dan menentukan nilai $R_3 = 100\Omega$, maka R_1 dapat di cari dengan persamaaan sebagai berikut :

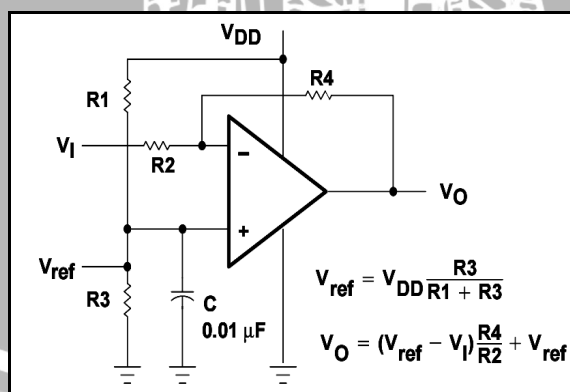
$$V_{ref} = \frac{R_3}{R_1 + R_3} V_{DD} \quad (4.7)$$

$$43,2 + 0,432R_1 = 100 \times 5$$

$$R_1 = \frac{500 - 43,2}{0,432} = 1057 \Omega$$

Keterangan : V_i adalah tegangan input dari blok sensor

Untuk Perancangan Rangkaian Pengkondisi Sinyal untuk Sensor CO2 dijelaskan dalam Gambar 4.4.



Gambar 4.4 Perancangan RPS Sensor CO₂
Sumber : Datasheet

2. Perancangan RPS Sensor O₂

Rangkaian pengkondisi sinyal merupakan rangkaian yang menghubungkan antara keluaran sensor KE-50 yang telah dikonversikan menjadi tegangan dengan masukan ADC dari Atmega8P. Rangkaian ini berfungsi untuk mengatur tingkat sensitivitas dan akurasi dari alat Monitoring kadar Gas Oksigen. Tegangan keluaran

yang dihasilkan oleh sensor berkisar antara 0 V – 230 mV akan dimanipulasi sehingga menjadi tegangan antara 0 V – 2,56 V (tegangan 2,56 V merupakan tegangan referensi dari ADC pada Atmega8P yang dipilih oleh perancang).

Agar Tegangan 0mV bernilai 0V dan tegangan 230mV bernilai 2,56V, maka tegangan referensi serta penguatan yang di butuhkan oleh penguat Tak membalik adalah :

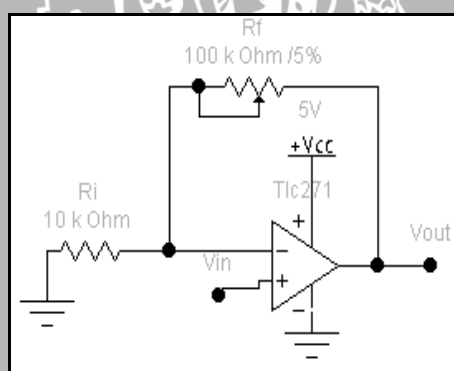
$$\frac{V_o}{V_i} = \left(\frac{R_f}{R_i} + 1 \right) \quad (4.8)$$

$$\frac{2.56}{0.23} = \left(\frac{R_f}{R_i} + 1 \right)$$

$$\frac{R_f}{R_i} = 10.130 \approx 10 \text{ kali}$$

Jika $R_i = 10\text{k}\Omega$ maka R_f nya $100\text{k}\Omega$

Untuk Perancangan Rangkaian Pengkondisi Sinyal untuk Sensor CO₂ dijelaskan dalam Gambar 4.5.



Gambar 4.5 Perancangan RPS Sensor O₂
Sumber : Perancangan

iii. Mikrokontroler ATmega8P

Sistem mikrokontroler Atmega8P berfungsi sebagai pengatur aliran data dan sinyal-sinyal kontrol dengan memanfaatkan port-port yang telah tersedia. Periperal penting yang dipakai pada perancangan alat ini adalah ADC yang ada pada Atmega8P. ADC adalah rangkaian yang berfungsi untuk mengubah sinyal analog menjadi data digital. Pada mikrokontroler Atmega8P ini terdapat ADC dengan resolusi 10 bit yang mampu mengkonversi data selama 13 – 260 μs .

Tegangan referensi yang terdapat pada ADC ini dapat dipilih dari tegangan internal atau eksternal. Pada perancangan ini tegangan referensi yang digunakan ialah tegangan internal sebesar 2.56 volt. Masukan pada ADC ini terdapat 6 buah channel

yang termultipleks, namun dalam perancangan ini hanya digunakan 2 buah channel, dimana ADC0 sebagai masukan dari rangkaian pengkondisi sinyal sensor sedangkan ADC1 sebagai masukan dari rangkaian indikator baterai. Resolusi ADC Atmega8P 10 bit dicari dengan perhitungan sebagai berikut:

$$\begin{aligned}\text{Resolusi Teg ADC} &= V_{\text{ref}} \times 2^{-n} \\ &= 2,56 \times 2^{-10} \\ &= 2,5 \text{ mV}\end{aligned}\quad (4.9)$$

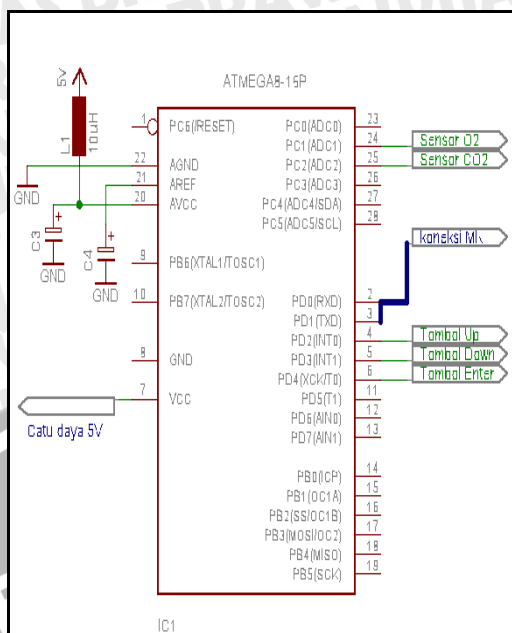
Frekuensi kerja ADC tergantung dari frekuensi mesin pada mikrokontroler yang mana frekuensi ADC dapat dipilih dengan konfigurasi CLK/2, CLK/4, CLK/8, CLK/16, CLK/32, CLK/64 dan CLK/128. Dalam perancangan ini frekuensi kerja mikrokontroler sebesar 4 MHz dan konfigurasi ADC yang digunakan sebesar CK/32 yaitu 0,125 MHz. Untuk melakukan konversi diperlukan 13 clock dan channel yang digunakan sebanyak 2 buah sehingga frekuensi sampling maksimum yang dapat dioperasikan pada masing – masing channel sebesar :

$$\begin{aligned}f_{\text{sampling}} &= \frac{\text{frekuensi ADC}}{2 \times 13} \\ &= \frac{125\,000 \text{ Hz}}{2 \times 13} \\ &= 4,808 \text{ kHz}\end{aligned}\quad (4.10)$$

Hasil konversi data yang tersimpan pada ADC Data Register 10 bit diperoleh dengan perhitungan :

$$\begin{aligned}\text{ADC Data Register} &= \frac{V_{\text{in}}}{V_{\text{ref}}} \times 2^{10} \\ &= \frac{V_{\text{in}}}{2,56} \times 1024\end{aligned}\quad (4.11)$$

Gambar 4.6 menunjukkan sistem rangkaian mikrokontroler Atmega8P yang akan diantarmukakan dengan rangkaian sensor, koneksi ke mikrokontroller yang lain, dan peripheral pendukung yang berupa tiga buah tombol push button.



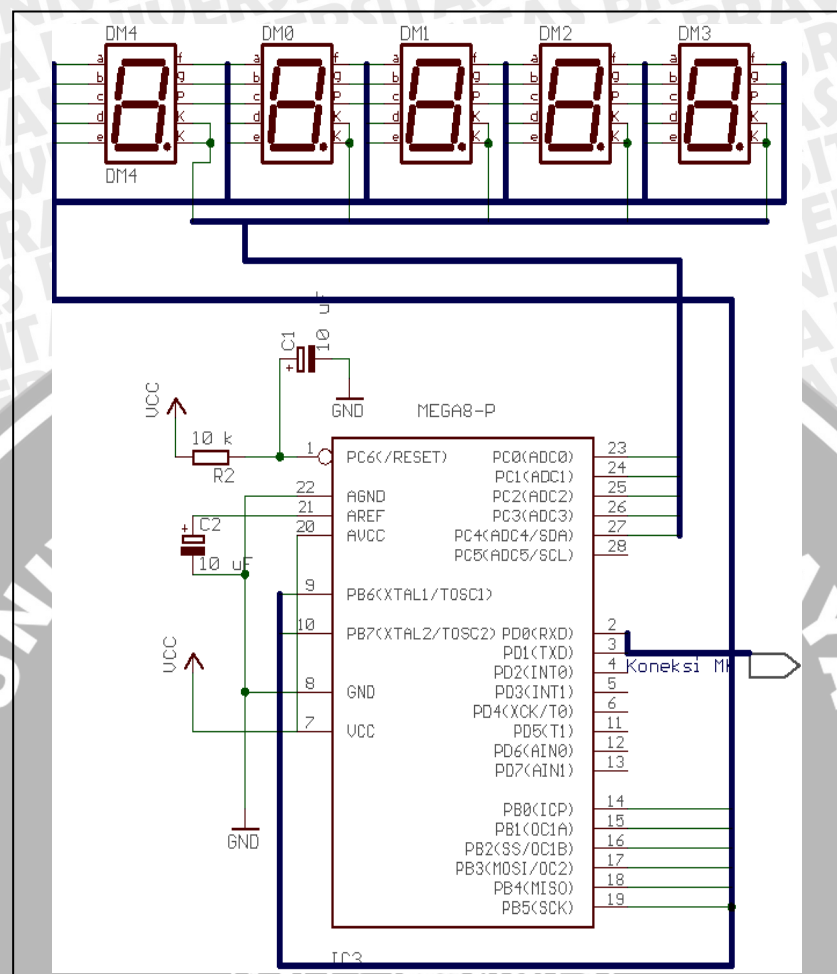
Gambar 4.6 Blok Perancangan Mikrokontroler Atmega8P sebagai kontroler utama
Sumber : perancangan

Untuk memanfaatkan peripheral ADC pada Atmega8P maka sistem mikrokontroler Atmega8P ini perlu dihubungkan dengan sebuah kapasitor sebesar 1uF pada kaki 21 pin AREF serta pin AVCC dengan Vcc. Dalam perancangan alat ini Kami memanfaatkan kristal internal sebesar 1 MHz serta tegangan referensi internal sebesar 2,56 V sebagai tegangan referensi ADC. Dalam sistem mikrokontroler ini direncanakan penggunaan pin yang tersedia sebagai berikut:

1. Pin 24 dan Pin 25 sebagai keluaran RPS sensor konversi nilai ADC
2. Pin 2 dan 3 di hubungkan dengan Pin 2 dan 3 yang ada Mikrokontroller ATmega8 untuk display
3. Pin 4, 5 dan 6 di gunakan untuk fungsi tombol

iv. Perancangan Display

Alat ini akan menampilkan nilai kadar Gas Oksigen antara 5-100 % dan untuk nilai kadar Gas karbondioksida antara 0-5000ppm. Seven segment yang di butuhkan sebanyak 5 buah. 4 buah untuk menampilkan nilainya dan 1 buah untuk menampilkan urutan abjad pada saat ingin menampilkan semua nilai. Sevent segment ini di hubungkan dengan Mikrokontroler ATmega8P sendiri. Gambar 4.7 menunjukkan Perancangan display yang akan ditampilkan ke seven segmen.

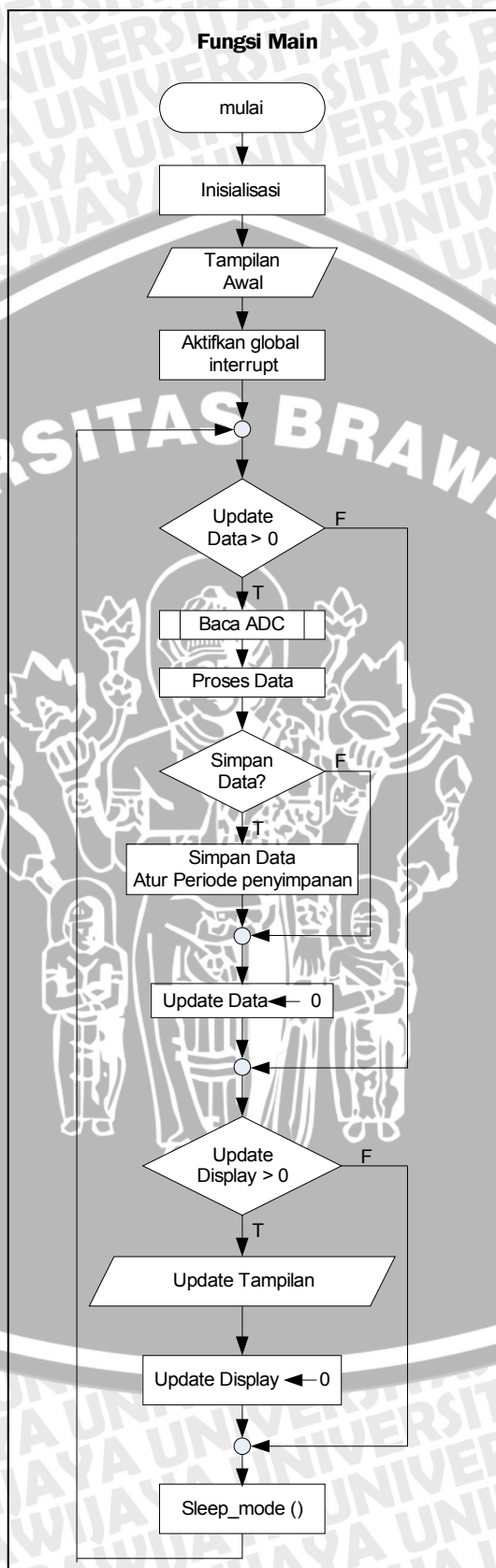


Gambar 4.7 Blok Perancangan Display
Sumber : perancangan

4.5 Perancangan Perangkat Lunak

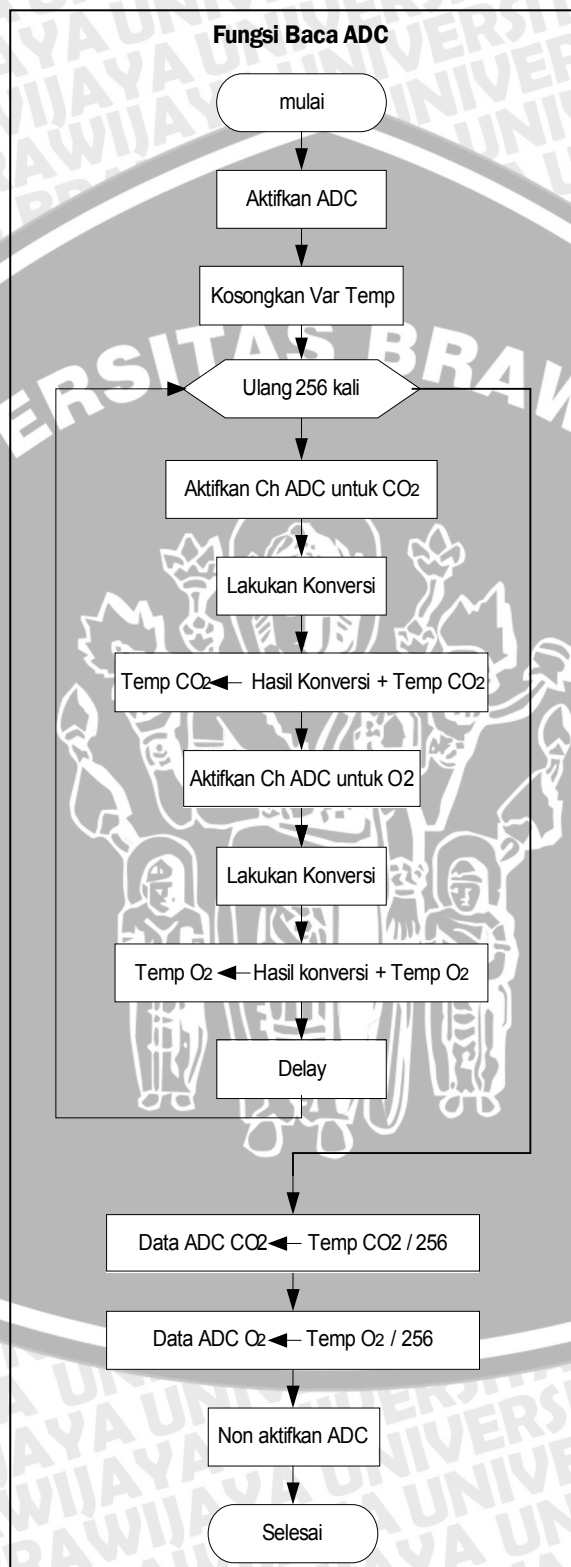
Perangkat lunak dengan bahasa C yang dirancang untuk mikrokontroler Atmega8P harus mampu memanfaatkan fasilitas ADC dan Register Data serta mampu melakukan operasi aritmatika, logika dan transfer data. ADC akan digunakan untuk mengolah data dari rangkaian sensor (rangkaian pengkondisi sinyal) dan rangkaian indikator baterai. Pemanfaatan register data dengan tepat dimaksudkan agar program yang kita buat dapat berfungsi dengan baik dan mudah dilakukan analisis. Selain itu program harus mampu membaca input berupa 3 buah tombol pushbutton, yaitu 1 buah tombol merah berfungsi sebagai operasi memilih menu ke atas (tombol Up), 1 buah tombol kuning berfungsi sebagai operasi memilih menu ke bawah (tombol Down) dan 1 buah tombol hijau untuk operasi enter/yes. Output dari program yang berupa nilai konsentrasi Gas Oksigen dan Karbondioksida akan ditampilkan pada Seven Segmen.

Diagram Alir Sistem Utama untuk sistem perancangan pada Mikrokontroler ditunjukkan dalam gambar 4.8.



Gambar 4.8 Diagram alir sistem utama perangkat lunak
Sumber : perancangan

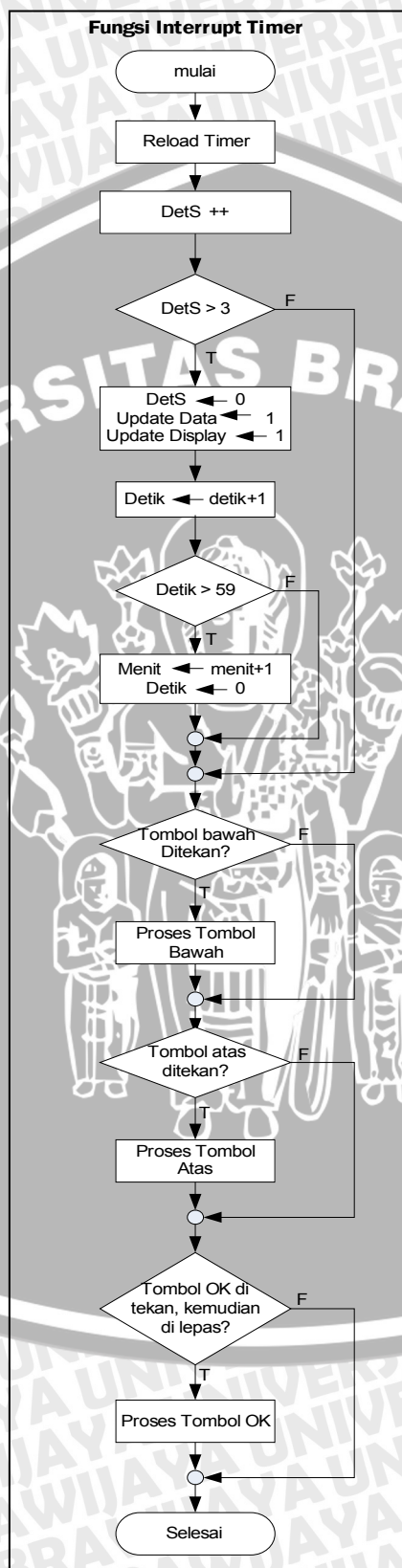
Untuk melakukan proses konversi besaran analog menjadi besaran digital, ADC dimanfaatkan dalam proses ini. Fungsi baca ADC ini akan dipanggil pada fungsi utama. Diagram alir dari perangkat lunak baca ADC adalah ditunjukkan oleh gambar 4.9.



Gambar 4.9 Diagram Alir fungsi baca ADC

Sumber : perancangan

Peripheral Timer/Counter digunakan untuk mengatur periode pembacaan push button dan tampilan seven segment. Diagram Alir dari fungsi interrupt timer ditunjukkan dalam Gambar 4.10.



Gambar 4.10 Diagram alir dari fungsi interrupt Timer
Sumber : Perancangan

BAB V

PENGUJIAN DAN ANALISIS SISTEM

Pengujian dilakukan untuk mengetahui kerja dari rangkaian yang dirancang apakah dapat berkerja dengan baik dan sesuai dengan harapan. Pengujian dan analisis mula-mula dilakukan pada tiap bagian, jika semua bagian telah dapat bekerja dilakukan pengujian pada sistem secara keseluruhan. Adapun pengujian yang dilakukan adalah:

1. Pengujian Sensor
2. Pengujian Rangkaian Pengkondisi Sinyal
3. Pengujian ADC internal
4. Pengujian Display
5. Pengujian Keseluruhan sistem

Setelah pengujian selesai, dilakukan analisis terhadap hasil pengujian yang diperoleh.

Adapun alat-alat dan *software* yang digunakan dalam pengujian sebagai berikut:

1. Adaptor MITOYOURI 3 – 12 V dengan arus maksimal 1,2 A
2. Multimeter digital HB 9205A
3. Minimum sistem AVR ATmega8L.
4. *Software* AVR Studio4 dan PonyProg2000.
5. Rangkaian Sevent Segment.
6. Tombol *pushbutton*.

5.1 Pengujian Sensor

5.1.1 Tujuan

Pengujian *Sensor* dilakukan untuk mengetahui apakah keluaran *Sensor* dalam kondisi baik dan sesuai dengan data yang diberikan pada datasheet.

5.1.2 Metode Pengujian

Metode pengujian yang dilakukan adalah dengan Pengujian yang di lakukan pada satu titik saja, yaitu ketika mendeteksi kadar Gas Oksigen dan Karbondioksida di udara.

Dari sumber situs [Http ://id.wikipedia.org/wiki/Udara](http://id.wikipedia.org/wiki/Udara), didapatkan bahwa kadar Oksigen di udara antara 20-21 %, di ambil nilai rata-ratanya bahwa di udara ada 20,5 % Oksigen. Sedangkan untuk kadar Gas Karbondioksida di udara antara 370-380ppm, diambil nilai rata-ratanya bahwa di udara ada 375 ppm Karbondioksida.

Data tersebut di dapatkan pada saat kondisi udara bersih yaitu udara yang jauh dari polusi yang di akibatkan zat-zat pencemaran (polusi udara), proses industrilisasi, pembuangan limbah, pembakaran bahan bakar posil, asap kendaraan bermotor. Udara bersih juga dipengaruhi oleh suhu, Jika terlalu panas akan mengakibatkan meningkatnya emisi-emisi polutan, seperti SO₂ dan jika dingin akan mengakibatkan udaranya lebih tercemar (Yusniwarti Yusad, *Polusi Udara Dikota-Kota Besar Dunia*)

5.1.3 Prosedur Pengujian

Prosedur pengujian sensor dilakukan sebagai berikut :

1. Meletakkan Sensor dalam kondisi terbuka, dan mencatat berapa perubahan tegangan output sensor yang di hasilkan pada saat di udara (dianggap kadar CO₂ sama dengan 375ppm dan kadar gas O₂ sama dengan 20,5%)
2. Mencari Tegangan Output Sensor menurut datasheet
3. Tegangan Output Sensor yang dihasilkan berdasarkan pengujian tersebut dibandingkan dengan Tegangan Output Sensor menurut datasheet
4. Hitung nilai rata-rata penyimpangannya

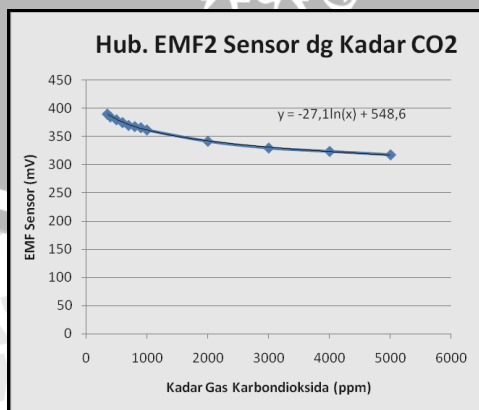
5.1.4 Analisis dan Hasil Pengujian

a. Pengujian Sensor CO₂

Setelah pengujian sensor pada kondisi di udara bebas, tegangan output yang di keluarkan sensor : 375, 380, 383, 385, 392, 395 dalam satuan mV.

(NB: udara yang diukur di ruangan berbentuk tabung dengan d=14cm dan t=18cm).

Sedangkan data konsentrasi Gas Oksigen yang di dapatkan dari datasheet, diperlihatkan dalam gambar 5.1.



Gambar 5.1 Grafik karakteristik Sensor Gas CO₂ dalam Datasheet

Dari grafik diatas, didapatkan persamaan :

$$y = -27,1 \ln(x) + 548,6 \quad (5.1)$$

Keterangan :

y adalah nilai tegangan output sensor (mV)

x adalah nilai kadar Gas Karbondioksida (ppm)

Untuk mendapatkan kadar Gas Karbondioksida sebesar 375ppm maka nilai Tegangan sensornya adalah:

$$\begin{aligned} y &= -27,1 \ln(x) + 548,6 \\ &= -26,9 \ln(375) + 535 \approx 388 \text{ mV} \end{aligned}$$

Jadi, pada saat nilai kadar Gas Oksigen sama dengan 375ppm maka nilai tegangan output sensor sebesar 388mV.

Hasil Pengujian :

Tabel 5.1 menunjukkan Data Hasil perbandingan Pengujian Sensor TGS4161 pada saat mengukur kadar Karbondioksida di udara yang di bandingkan dengan perhitungan sensor yang ada pada datasheet yang sudah di jelaskan di atas.

Tabel 5.1 Data Hasil Pengujian Sensor TGS4161

No	Kadar Gas CO2 di udara	Output Sensor (mV)		% Selisih
		Hasil Pengukuran	Hasil Perhitungan	
1	375 ppm	375	388	0.27
2	375 ppm	380	388	1.06
3	375 ppm	383	388	1.86
4	375 ppm	385	388	2.39
5	375 ppm	392	388	4.26
6	375 ppm	395	388	5.05
Rata-rata Penyimpangan				2.48

Prosentase Selisih antara hasil Pengukuran dengan hasil perhitungan dapat dihitung dengan persamaan, sebagai berikut :

$$\% \text{Selisih} = \left| \frac{\text{HasilPerhitungan} - \text{HasilPengukuran}}{\text{HasilPerhitungan}} \right| \times 100\% \quad (5.2)$$

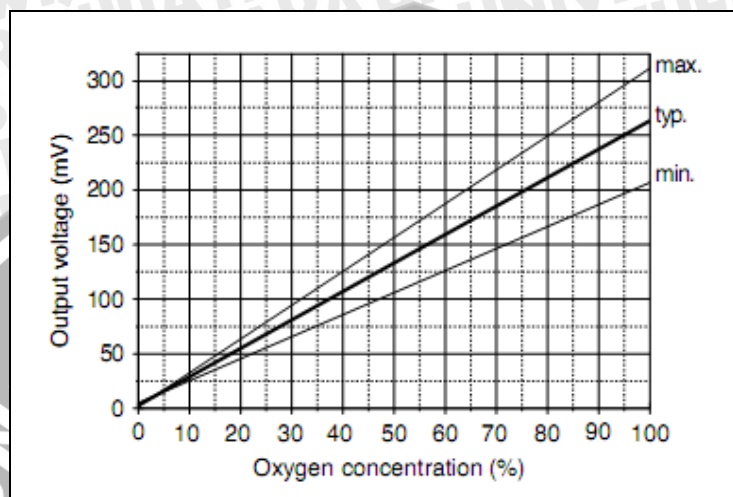
Hasil Pengujian di atas, menunjukkan Rata-rata prosentase Selisih untuk pengujian Sensor Karbondioksida sebesar 2.48 %. Sehingga dapat disimpulkan bahwa Sensor masih bekerja baik karena Selisihnya tidak terlalu besar.

b. Pengujian Sensor O2

Setelah pengujian sensor pada kondisi di udara bebas, tegangan output yang di keluarkan sensor: 50.8; 50.9; 51; 51.1; 51.2 dalam satuan mV.

(NB: udara yang diukur di ruangan berbentuk tabung dengan d=14cm dan t=18cm).

Sedangkan data konsentrasi Gas Oksigen yang di dapatkan dari datasheet, diperlihatkan dalam gambar 5.2



Gambar 5.2 Grafik karakteristik Sensor Gas Oksigen dalam Datasheet

Dari grafik dengan Type Typical diatas didapatkan persamaan :

$$y_2 = 2,284x + 3,121 \quad (5.3)$$

Untuk mendapatkan kadar Gas 20.5 %, maka nilai tegangan output sensor Ke-50 adalah :

$$\begin{aligned} y_2 &= 2,284 * 20,5 + 3,121 \\ &= 49,94mV \end{aligned}$$

Hasil Pengujian

Tabel 5.2 menunjukkan Data Hasil perbandingan Pengujian Sensor KE-50 pada saat mengukur kadar Oksigen di udara yang di bandingkan dengan perhitungan sensor yang ada pada datasheet yang sudah di jelaskan di atas.

Tabel 5.2 Data Hasil Pengujian Sensor KE-50

No	Kadar Gas O ₂ di Udara (%)	Vo Sensor di Udara (mV)		% Selisih
		pengukuran	perhitungan	
1	20.5%	50.8	49,94	1,72
2	20.5%	50.9	49,94	1,92
3	20.5%	51	49,94	2,12
4	20.5%	51.1	49,94	2,32
5	20.5%	51.2	49,94	2,52
Rata-rata penyimpangan (%)				2,12

Prosentase Selisih antara hasil Pengukuran dengan hasil perhitungan dapat dihitung dengan persamaan 5.2.

Hasil Pengujian di atas, menunjukkan Rata-rata prosentase Selisih untuk pengujian Sensor Oksigen sebesar 2.12 %. Sehingga dapat disimpulkan bahwa Sensor masih bekerja baik karena Selisihnya tidak terlalu besar.

5.2 Pengujian Rangkaian Pengkondisi Sinyal

5.2.1 Tujuan

Pengujian Rangkaian Pengkondisi Sinyal dilakukan untuk mengetahui hasil keluaran Rangkaian Pengkondisi Sinyal sesuai Penguatan dalam perancangan Alat

5.2.2 Prosedur Pengujian

Prosedur pengujian sensor dilakukan sebagai berikut :

- 1) Menyiapkan Rangkaian Pengkondisi Sinyal, rangkaian untuk mendapatkan tegangan murni, multimeter digital, dan adaptor.
- 2) Memberi Tegangan input pada Rangkaian pengondisi Sinyal dengan melihat hasil pengukurannya melalui multimeter digital
- 3) Amati tegangan output yang dihasilkan dengan menggunakan multimeter digital kemudian di catat dalam tabel
- 4) Mencari prosentase kesalahannya

5.2.3 Hasil Pengujian

Untuk mendapatkan Hasil Perhitungan Tegangan Output dari Sensor TGS4161, dapat menggunakan persamaan:

$$V_{out} = \frac{R_f}{R_i}(V_{ref} - V_{in}) + V_{ref} \quad (5.4)$$

Keterangan :

- Vin adalah tegangan masukan yang diberikan
- Vref adalah tegangan referensi dari penguat pengurang sebesar 368mV
- Rf/Ri sebagai penguatan sebesar 12 kali

Sedangkan untuk mendapatkan Hasil Perhitungan Tegangan Output dari Sensor KE-50, dapat menggunakan persamaan :

$$V_{out} = \left(\frac{R_f}{R_i} + 1 \right) * V_{in} \quad (5.5)$$

Keterangan :

- Vin adalah tegangan masukan yang diberikan
- Penguatan dari penguat sebesar 11 kali

Persamaan tersebut sudah di jelaskan dalam subbab pengujian Sensor.

Tabel 5.3 dan 5.4 menunjukkan data hasil pengujian rangkaian pengondisi sinyal untuk sensor CO2 dan sensor O2.

Prosentase Kesalahan dari hasil pengujian dibandingkan dengan hasil yang diinginkan dalm perancangan dapat dihitung dengan persamaan 5.2.

Tabel 5.3 Tabel hasil pengujian RPS untuk Sensor TGS4161

No	Tegangan Masukan (mV)	Tegangan Keluaran (mV)		% Kesalahan
		Hasil Perhitungan	Hasil Pengukuran	
1	370	1176	1178	0,17
2	360	1296	1288	0,62
3	350	1416	1424	0,56
4	340	1536	1564	1,82
5	330	1656	1644	0,72
6	320	1776	1751	1,41
7	310	1896	1870	1,37
Rata-rata penyimpangan				0.95

Tabel 5.4 Tabel Hasil pengujian RPS untuk Sensor Ke-50

No	Tegangan Masukan (mV)	Tegangan Keluaran (mV)		% Kesalahan
		Hasil Perhitungan	Hasil Pengukuran	
1	50	550	540	1.82
2	75	825	807	2.18
3	100	1100	1078	2.00

4	125	1375	1349	1.89
5	150	1650	1613	2.24
6	175	1925	1883	2.18
7	200	2200	2130	3.18
8	225	2475	2400	2.21
Rata-rata penyimpangan				2.21

Hasil Pengujian di atas, menunjukkan Rata-rata Penyimpangan untuk Rangkaian Pengkondisi Sinyal Sensor CO₂ sebesar 0.95 % sedangkan untuk sensor O₂ sebesar 2.21 %. Sehingga dapat disimpulkan bahwa Rangkaian Pengkondisi Sinyal masih bekerja baik karena Penyimpangannya tidak terlalu besar.

5.3 Pengujian ADC Internal

5.3.1 Tujuan

Pengujian ADC dilakukan untuk mengetahui hasil konversi ADC Atmega8 10-bit berdasarkan tegangan analog yang dimasukkan.

5.3.2 Prosedur Pengujian

Prosedur pengujian ADC dilakukan sebagai berikut :

1. Menyiapkan minimum system Atmega8 dan Rangkaian display yang sudah terisi dengan program yang menampilkan nilai ADCnya pada seven segment
2. Memberi tegangan masukan tertentu pada salah satu pin 24/25
3. Mencatat hasil konversi bilangan desimal yang ditampilkan
4. Mencari Prosentase Kesalahannya

5.3.3 Hasil Pengujian

Hasil pengujian menunjukkan bahwa Rangkaian Pengkondisi Sinyal dapat bekerja dengan baik. Hal ini terlihat dari perubahan Rata-rata penyimpangan masih dalam batas toleransi.

Data pengujian ADC Internal Mikrokontroler ATmega8 ditunjukkan dalam Tabel 5.5.

Tabel 5.5 Hasil pengujian ADC internal Mikrokontroler ATmega8P

No	Tegangan Masukan (mV)	Keluaran ADC		% Kesalahan
		Hasil Pengukuran	Hasil Perhitungan	
1	100	39	40	2.50
2	500	195	200	2.50

3	1000	391	400	2.25
4	1250	491	500	1.80
5	1500	590	600	1.67
6	1750	691	700	1.29
7	2000	790	800	1.25
8	2250	882	900	2.00
9	2500	984	1000	1.60
Rata-rata Penyimpangan				1.87

Prosentase Kesalahan dari hasil pengujian dibandingkan dengan hasil yang diinginkan dalam perancangan dapat dihitung dengan persamaan 5.2.

Untuk mendapatkan nilai ADC secara perhitungan dengan menggunakan persamaan :

$$\text{Nilai ADC} = \frac{V_{inadc}}{V_{ref}} * 1024 \quad (5.6)$$

Ket: V_{ref} adalah tegangan referensi ADC internal

V_{inadc} adalah tegangan output dari RPS

Hasil Pengujian di atas, menunjukkan Rata-rata Penyimpangannya sebesar 1.87 %. Sehingga dapat disimpulkan bahwa ADC dapat mengkonversi sesuai dengan perancangan yang diinginkan meskipun ada selisih antara 1-16 bit. Itu artinya beda selisih tegangan yang masuk ke ADC Mikrokontroler hanya 2,5mV - 40 mV.

5.4 Pengujian Display

5.4.1 Tujuan

Pengujian *display* dilakukan untuk mengetahui apakah keluaran *display* dapat menampilkan sesuai dengan data yang diberikan.

5.4.2 Prosedur pengujian

Prosedur pengujian sensor dilakukan sebagai berikut :

1. Menyiapkan minimum system Atmega8 dan Rangkaian display yang sudah terisi dengan program yang menampilkan bilangan desimal
2. Mengamati hasilnya dan mendokumentasikannya

5.4.3 Hasil pengujian

Pada pengujian ini data yang ingin ditampilkan pada *display* adalah angka desimal. Hasil pengujian ditunjukkan dalam Gambar 5.3.



Gambar 5.3 Hasil Pengujian Display

5.5 Pengujian keseluruhan

5.5.1 Tujuan

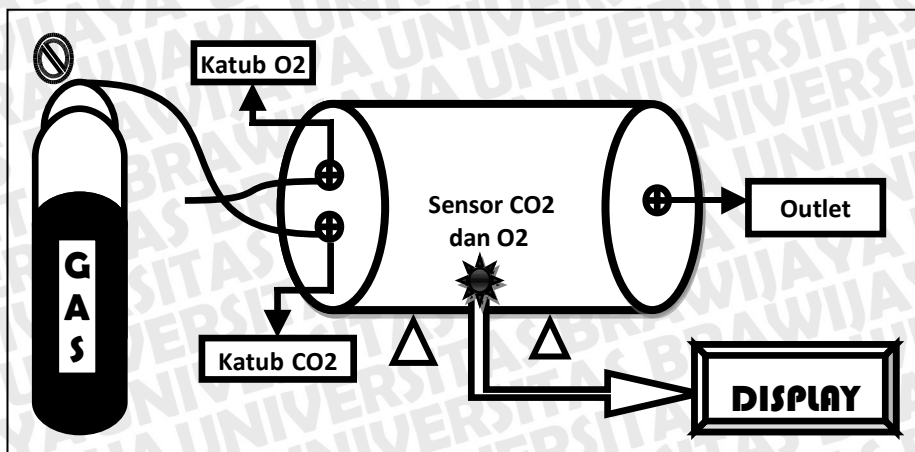
Pengujian Keseluruhan Sistem bertujuan untuk mengetahui nilai kemampuan alat dan juga untuk melihat tingkat akurasi dan presisi dari Alat

5.5.2 Prosedur pengujian

Prosedur pengujian dilakukan melalui prosedur sebagai berikut :

1. Merangkai rangkaian sesuai dengan Blok Diagram Sistem yang dirancang
2. Memasukkan program secara keseluruhan ke dalam mikrokontroller.
3. Merangkai alat dengan tabung pengujian sesuai dengan gambar 4.4.
4. Mengaktifkan “Alat Monitoring Gas CO₂ dan O₂”.
5. Tabung Gas akan memberikan Gas Oksigen terlebih dahulu kemudian gas Karbondioksida
6. Tekanan yang diberikan pada saat Kadar Oksigen dimasukan 0.5kg/cm² sedangkan waktu memasukkan kadar CO₂nya di beri tekanan 10 psi

Blok diagram rangkaian pengujian keseluruhan ditunjukkan dalam Gambar 5.4.



Gambar 5.4 Blok diagram pengujian system secara keseluruhan
Sumber : Perancangan

5.5.3 Hasil dan Analisa pengujian

Hasil Pengujian yang didapatkan sebagai berikut :

1. Awalnya pada saat belum di masukkan Gas Oksigen dan Karbondioksida, Kadar Gas O_2 menunjukkan angka antara (19,4; 19,5; 19,6) dalam satuan % dan untuk kadar Gas CO_2 menunjukkan angka (372, 376, 384, 386, 388, 395, 400) dalam satuan ppm.
2. Pada saat di berikan Gas oksigen dengan jumlah tertentu, Nilai Kadar O_2 terus naik sampai menunjukkan nilai maksimum 97 % O_2 sedangkan display CO_2 menunjukkan 0 ppm.
3. Setelah itu, Secara bergantian Gas CO_2 diberikan sehingga menyebabkan nilai kadar gas O_2 menurun sampai 5% Oksigen... Sebaliknya kadar gas CO_2 semakin naik sampai mencapai nilai 4965ppm.
4. Dari analisa tersebut dapat dibuatkan hasil pada tabel 5.6.

Tabel 5.6 Hasil Pengujian Kemampuan Alat

Kadar Oksigen (%)		Kadar karbondioksida (ppm)	
Di Udara	19	Di Udara	386
Nilai Minimum	5	Nilai Minimum	0
Nilai Maksimum	97	Nilai Maksimum	4965

$$\text{Akurasi (\%)} = \frac{\left| \frac{\text{Beda Pengukuran}}{\text{Hasil Pengukuran}} \right|}{1} \times 100\% \quad (5.7)$$

$$\text{Presisi (\%)} = \frac{(\text{Keluaran Alat}) - (\text{Keluaran Alat})}{\text{Keluaran Alat}} \times 100\% \quad (5.8)$$

dengan: Keluaran Alat = Rata-Rata Keluaran Alat

Perhitungan tingkat presisi dan akurasi pada pengukuran di udara bersih, dengan kadar Oksigen 20,5% dapat diselesaikan sebagai berikut:

$$\text{Akurasi (\%)} = \left| \frac{1,1}{19,4} \right| \times 100\% = 5,67\% \text{ (Untuk Kadar Oksigen 20,5 \%)}$$

$$\text{Akurasi (\%)} = \left| \frac{1,0}{19,5} \right| \times 100\% = 5,13\% \text{ (Untuk Kadar Oksigen 20,5 \%)}$$

$$\text{Akurasi (\%)} = \left| \frac{0,9}{19,6} \right| \times 100\% = 4,59\% \text{ (Untuk Kadar Oksigen 20,5 \%)}$$

$$\therefore \text{Akurasi Rata-Rata (\%)} = \frac{5,67 + 5,13 + 4,59}{3} = 5,13\%$$

$$\text{Presisi (\%)} = \left| \frac{19,4 - 19,5}{19,5} \right| \times 100\% = 0,513\% \text{ (Untuk Kadar Oksigen 20,5 \%)}$$

$$\text{Presisi (\%)} = \left| \frac{19,5 - 19,5}{19,5} \right| \times 100\% = 0,000\% \text{ (Untuk Kadar Oksigen 20,5 \%)}$$

$$\text{Presisi (\%)} = \left| \frac{19,6 - 19,5}{19,5} \right| \times 100\% = 0,513\% \text{ (Untuk Kadar Oksigen 20,5 \%)}$$

$$\therefore \text{Presisi Rata-Rata (\%)} = \frac{0,513 + 0,000 + 0,513}{3} = 0,342\%$$

Tabel 5.7 Hasil Pengukuran untuk O₂

N o	Kadar O2 diudara (%)	Hasil Pengukuran (%)	Beda Pengukuran	Akurasi (%)	Akurasi rata-rata (%)	Presisi (%)	Presisi rata- rata (%)
1	20,5	19,4	1,1	5,67	5,13	0,513	0,342
2		19,5	1	5,13		0,000	
3		19,6	0,9	4,59		0,513	

Perhitungan tingkat presisi dan akurasi pada pengukuran di udara bersih, dengan kadar Karbondioksida 375ppm dapat diselesaikan sebagai berikut:

$$\text{Akurasi (\%)} = \left| \frac{3}{372} \right| \times 100 \% = 0,81 \% \text{ (Untuk Kadar CO}_2 \text{ 375ppm)}$$

$$\text{Akurasi (\%)} = \left| \frac{1}{376} \right| \times 100 \% = 0,27 \% \text{ (Untuk Kadar CO}_2 \text{ 375ppm)}$$

$$\text{Akurasi (\%)} = \left| \frac{9}{384} \right| \times 100 \% = 2,34 \% \text{ (Untuk Kadar CO}_2 \text{ 375ppm)}$$

$$\text{Akurasi (\%)} = \left| \frac{11}{386} \right| \times 100 \% = 2,85 \% \text{ (Untuk Kadar CO}_2 \text{ 375ppm)}$$

$$\text{Akurasi (\%)} = \left| \frac{13}{388} \right| \times 100 \% = 3,35 \% \text{ (Untuk Kadar CO}_2 \text{ 375ppm)}$$

$$\text{Akurasi (\%)} = \left| \frac{20}{395} \right| \times 100 \% = 5,06 \% \text{ (Untuk Kadar CO}_2 \text{ 375ppm)}$$

$$\text{Akurasi (\%)} = \left| \frac{25}{400} \right| \times 100 \% = 6,25 \% \text{ (Untuk Kadar CO}_2 \text{ 375ppm)}$$

$$\therefore \text{Akurasi Rata-Rata (\%)} = \frac{0,81 + 0,27 + 2,34 + 2,85 + 3,35 + 5,06 + 6,25}{7} = 2,99 \%$$

$$\text{Presisi (\%)} = \left| \frac{372 - 385,86}{385,86} \right| \times 100\% = 3,59 \% \text{ (Untuk Kadar CO}_2 \text{ 375ppm)}$$

$$\text{Presisi (\%)} = \left| \frac{376 - 385,86}{385,86} \right| \times 100\% = 2,56 \% \text{ (Untuk Kadar CO}_2 \text{ 375ppm)}$$

$$\text{Presisi (\%)} = \left| \frac{384 - 385,86}{385,86} \right| \times 100\% = 0,48 \% \text{ (Untuk Kadar CO}_2 \text{ 375ppm)}$$

$$\text{Presisi (\%)} = \left| \frac{386 - 385,86}{385,86} \right| \times 100\% = 0,04 \% \text{ (Untuk Kadar CO}_2 \text{ 375ppm)}$$

$$\text{Presisi (\%)} = \left| \frac{388 - 385,86}{385,86} \right| \times 100\% = 0,55 \% \text{ (Untuk Kadar CO}_2 \text{ 375ppm)}$$

$$\text{Presisi (\%)} = \left| \frac{395 - 385,86}{385,86} \right| \times 100\% = 2,37 \% \text{ (Untuk Kadar CO}_2 \text{ 375ppm)}$$

$$\text{Presisi (\%)} = \left| \frac{400 - 385,86}{385,86} \right| \times 100\% = 3,66 \% \text{ (Untuk Kadar CO}_2 \text{ 375ppm)}$$

$$\therefore \text{Presisi Rata-Rata (\%)} = \frac{3,59 + 2,56 + 0,48 + 0,04 + 0,55 + 2,37 + 3,66}{7} = 1,89 \%$$

Tabel 5.8 Hasil Pengukuran untuk CO₂

No	Kadar O ₂ di udara (ppm)	Hasil Pengukuran (ppm)	Beda Pengukuran	Akurasi	Akurasi rata-rata (%)	Presisi (%)	Presisi rata- rata (%)
1	375	372	3	0,81	2,99	3,59	1,89
2	375	376	1	0,27		2,56	
3	375	384	9	2,34		0,48	
4	375	386	11	2,85		0,04	
5	375	388	13	3,35		0,55	
6	375	395	20	5,06		2,37	
7	375	400	25	6,25		3,66	

Berdasarkan hasil pengujian seperti yang ditunjukkan pada Tabel 5.7 dan Tabel 5.8 untuk pengukuran Kadar Oksigen dan Karbondioksida di udara Bersih dengan tempat pengujian berbentuk tabung yang memiliki Diameter 14 cm dan Tinggi 18 cm, dapat disimpulkan bahwa dari 3 kali percobaan untuk mengukur kadar Oksigen, menghasilkan tingkat akurasi terbesar adalah 5,67 % dan tingkat akurasi rata-rata sebesar 5,13 % dengan tingkat presisi terbesar adalah 0,513 % dan tingkat presisi rata-rata sebesar 0,342 %. Sedangkan untuk mengukur Kadar Karbondioksida, menghasilkan tingkat akurasi terbesar adalah 6,25 % dan tingkat akurasi rata-rata sebesar 2,99 % dengan tingkat presisi terbesar adalah 3,66 % dan tingkat presisi rata-rata sebesar 1,89 %

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dari hasil perancangan dan pengujian alat dapat diambil kesimpulan sebagai berikut:

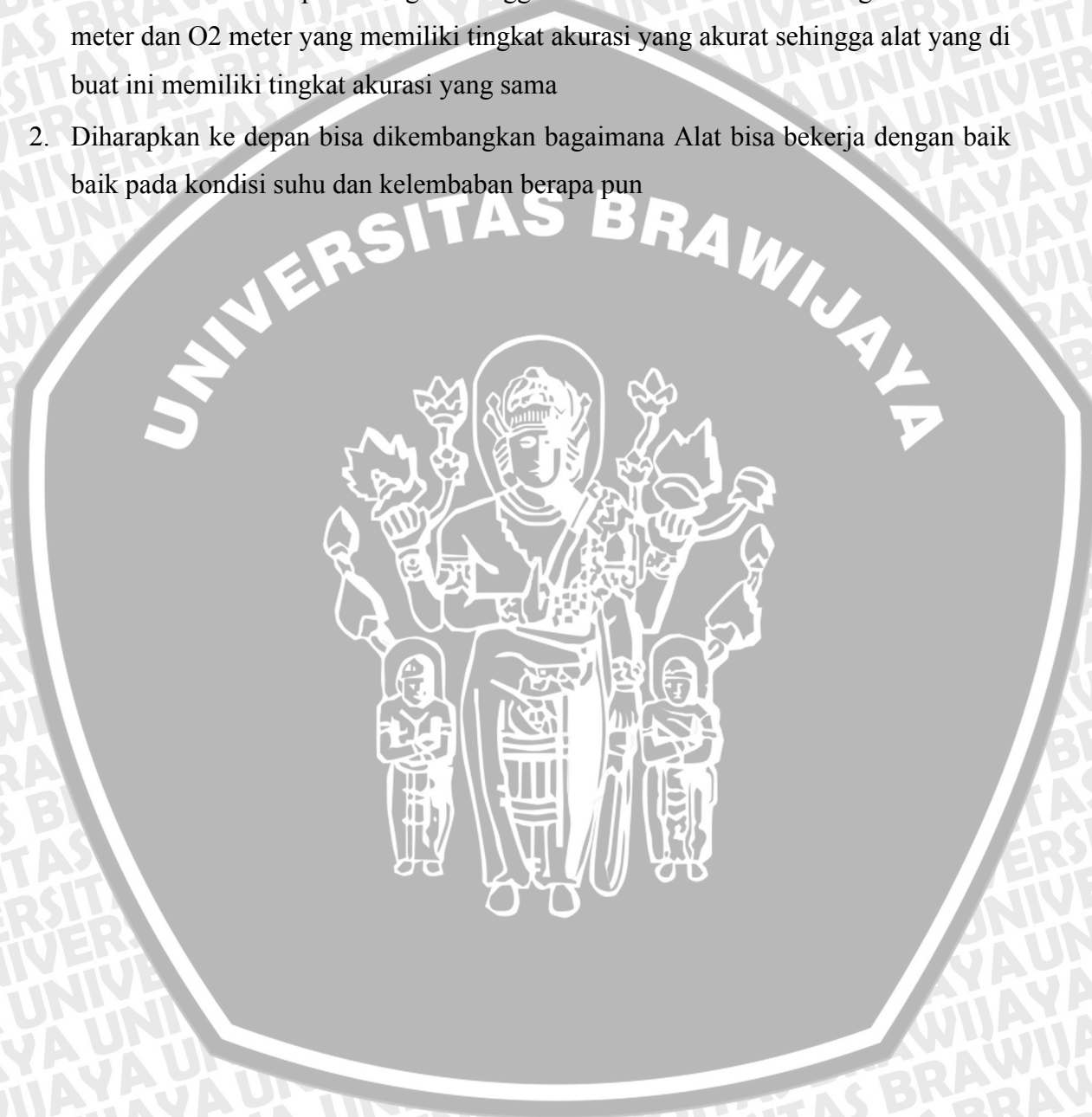
1. Dari hasil pengujian Alat Monitoring kadar Gas Oksigen dan Karbondioksida yang terjadi pada saat proses fermentasi ini didapatkan :
 - a. Pengujian sensor yang dilakukan dengan membandingkan Tegangan output sensor pada saat mendeteksi kadar gas O_2 dan CO_2 di udara dengan membandingkan nilai yang tercantum dalam datasheet. Dari hasil pengujian, Rata-rata prosentase selisih untuk pengujian Sensor CO_2 adalah 2.48 % dan untuk Sensor O_2 sebesar 2.12%.
 - b. Rangkaian untuk sensor CO_2 di gunakan rangkaian penyangga dan penguat inverting dengan tegangan Referensi, dengan penguatan sebesar 12 kali dan tegangan referensi sebesar 432mV, dalam pengujiannya didapatkan rata-rata prosentase selisih sebesar 0,95%. Sedangkan Rangkaian untuk sensor di gunakan rangkaian penyangga dan penguat tak membalik, dengan penguatan sebesar 11 kali. Dalam pengujiannya didapatkan rata-rata prosentase selisih sebesar 2,21%
 - c. Konverter analog ke digital menggunakan fasilitas yang ada dalam mikrokontroler Atmega8P, dalam pengujiannya didapatkan rata-rata prosentase kesalahan sebesar 1,87%.
 - d. Pengujian keseluruhan sistem dengan menguji kadar gas Oksigen di udara, didapatkan nilai Akurasi rata-ratanya sebesar 4,88 % dan pada saat gas Karbondioksida diudara, didapatkan rata-rata prosentase kesalahan sebesar 3,12 %.
 - e. Kemampuan alat ini mampu mengukur kadar Gas Oksigen dari 5-97% dan untuk mengukur kadar Gas CO_2 mampu mencapai 350-4965 ppm
2. Mikrokontroler Atmega8P yang di gunakan sebanyak 2 buah. Mikrokontroler pertama digunakan sebagai pengendali utama sistem. Devais ini berfungsi menangani proses input-output, penyimpanan dan pembacaan memori, serta proses

pengolahan data. Mikrokontroler kedua digunakan sebagai pengendali sistem display untuk menampilkan ke seven segmen

6.2 Saran

Saran-saran untuk pengembangan selanjutnya:

1. Kalibrasi alat diharapkan dengan menggunakan metode kalibrasi dengan alat CO₂ meter dan O₂ meter yang memiliki tingkat akurasi yang akurat sehingga alat yang di buat ini memiliki tingkat akurasi yang sama
2. Diharapkan ke depan bisa dikembangkan bagaimana Alat bisa bekerja dengan baik baik pada kondisi suhu dan kelembaban berapa pun



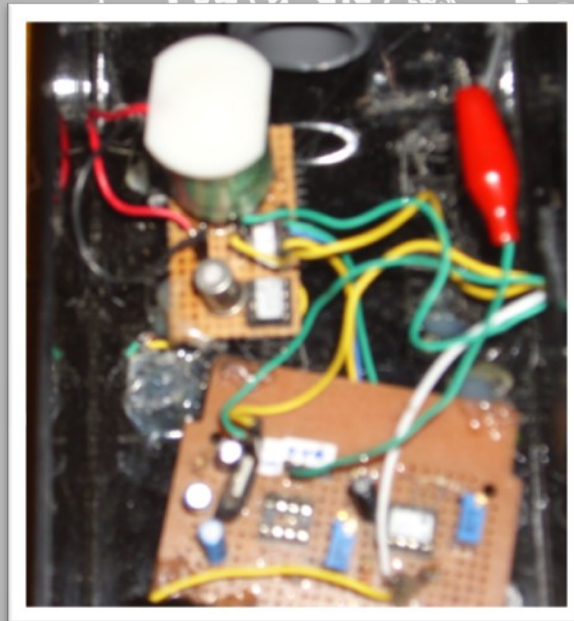
DAFTAR PUSTAKA

- Anonim. AVR ATMEGA 8 Advance Information Complete. www.atmel.com. Diakses tanggal 06 April 2009
- Anonymous, 2006. Penelitian kako. Indonesia : Penelitian kakao. <http://www.w3.org/TR/html/loose.dtd>. Diakses tanggal 06 April 2009
- Aswan. 2004. *Operational Amplifier*. www.electronclab.com Diakses 06 April 2009
- Atmel. 1999. AVR Embedded RISC Microcontroller Core. Atmel Corporation. (www.atmel.com)
- Atmel. 2007. ATMEGA 8P, 8-bit AVR Microcontroller with 8 Kbytes in System Programmable Flash. (www.atmel.com)
- Bejo, Agus. 2008. *C dan AVR Rahasia Kemudahan bahasa C dalam Mikrokontroler ATMEGA 8P*. Edisi Pertama. Yogyakarta: Graha Ilmu
- Coughlin, Robert dan Driscoll, Frederick. 1994. *Penguat Operational dan Rangkaian Terpadu Linier*. Jakarta: Penerbit Erlangga.
- Daniel, Archer. 2006. *The De Zaan Cocoa Manual*, Singapore: ADM Cocoa
- Figaro, 2009. *Data Sheet TGS4161 dan KE-50*. (<http://www.alldatasheet.com>), diakses pada 06 April 2009
- Malvino, Albert Paul. 1991. *Prinsip-Prinsip Elektronika*, Edisi Ketiga. Jakarta: Erlangga
- Misnawi. 2005. Peranan Pengolahan terhdap Pembentukan Cita Rasa Cokelat. Jember: Warta pusat Penelitian Kopi dan Kakao Indonesia
- Mulato, Sri. 2005. Pengolahan Produk Primer dan Sekunder Kakao. Jember: Pusat Penelitian Kopi dan Kakao
- Savitri. 2007. Kualitas Biji kakao Indonesia Masih rendah. Jakarta : pikiran rakyat. [Http://lipi.go.id/www.egi](http://lipi.go.id/www.egi). Diakses tanggal 06 April 2009
- Siregar, tumpal H. S. Dkk. 1988. Budidaya pengolahan dan Pemasaran Cokelat. Tanjung Morawa: Penebar Swadaya
- Wahyudi. T, dkk. 2008. *Panduan lengkap kakao*. Jakarta. Penebar swadaya.



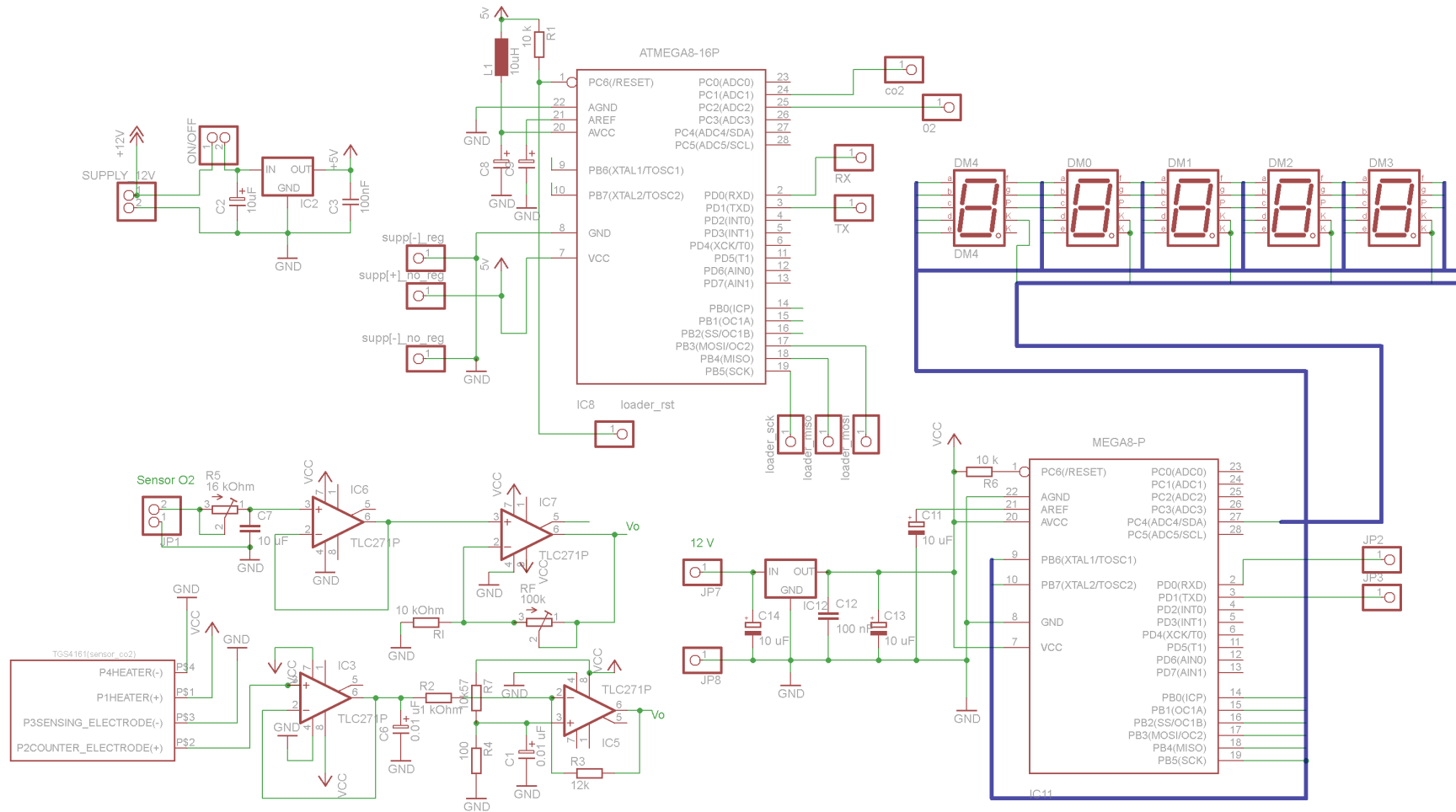
2.4.1 Alat Monitoring Proses Fermentasi Biji Kakao

2.4.2 Tampak dari depan



Gambar Peletakkan Sensor O₂ dan CO₂

LAMPIRAN



\\Program Utama monitoring CO dan O \\

2 2

#include <avr/io.h>

#include <avr/iom8.h>

#include <avr/pgmspace.h>

#include <avr/interrupt.h>

#include <util/delay.h>

#include <avr/sleep.h>

#include <avr/eeprom.h>



```
#include <math.h>
```

```
#include "gas_sens_main.h"
```

```
#include "data7segmh.h"
```

```
unsigned char segmenarray[] PROGMEM =
```

```
{
```

```
SSGMNDAT_0, SSGMNDAT_1, SSGMNDAT_2, SSGMNDAT_3, SSGMNDAT_4,
```

```
SSGMNDAT_5, SSGMNDAT_6, SSGMNDAT_7, SSGMNDAT_8, SSGMNDAT_9,
```


SSGMNDAT_A, SSGMNDAT_b, SSGMNDAT_C, SSGMNDAT_d, SSGMNDAT_E,

SSGMNDAT_F, SSGMNDAT_H, SSGMNDAT_h, SSGMNDAT_i, SSGMNDAT_J,

SSGMNDAT_L, SSGMNDAT_l, SSGMNDAT_P, SSGMNDAT_U, SSGMNDAT_a,

SSGMNDAT_SP,SSGMNDAT_MIN,SSGMNDAT_DOT,SSGMNDAT_n,SSGMNDAT_r

};

unsigned char menu_array[][DMENU_MAKS] PROGMEM =

{

{ SSGMNDAT_SP,SSGMNDAT_SP,SSGMNDAT_C,SSGMNDAT_0,SSGMNDAT_2 },

```
{ SSGMNDAT_SP,SSGMNDAT_SP,SSGMNDAT_0,SSGMNDAT_2,SSGMNDAT_SP },
```

```
{ SSGMNDAT_SP,SSGMNDAT_SP,SSGMNDAT_A,SSGMNDAT_L,SSGMNDAT_L }
```

```
};
```

```
unsigned char flashHCO2[3] PROGMEM = "CO2";
```

```
unsigned char flashHO2[2] PROGMEM = "O2";
```

```
//===== Struktur Data EEPROM=====
```

```
uint8_t EEMEM jamStart, menitStart ;
```

```
uint8_t EEMEM hCO2[3], eJmlDatCO2 ;
```



```
uint16_t EEMEM datCO2[83];
```

```
uint8_t EEMEM hO2[2], eJmlDatO2;
```

```
uint16_t EEMEM datO2[83];
```

```
//=====
```

```
volatile uint8_t menu, menuT, menit, detik, jam, menitLog, startLog;
```

```
volatile uint8_t statLog, jmlDatSuhu, jmlDatCO2, jmlDatO2;
```

```
volatile uint8_t transBuff[15], idRecv, packStatus, recvBuff[20];
```

```
volatile uint8_t wDisp, statT_Ok, updateData, updateDisp;
```

```
volatile uint8_t detS, statDispAll ;
```

```
volatile uint8_t statADC ;
```

```
volatile uint16_t datADC[5], dataSens[4] ;
```

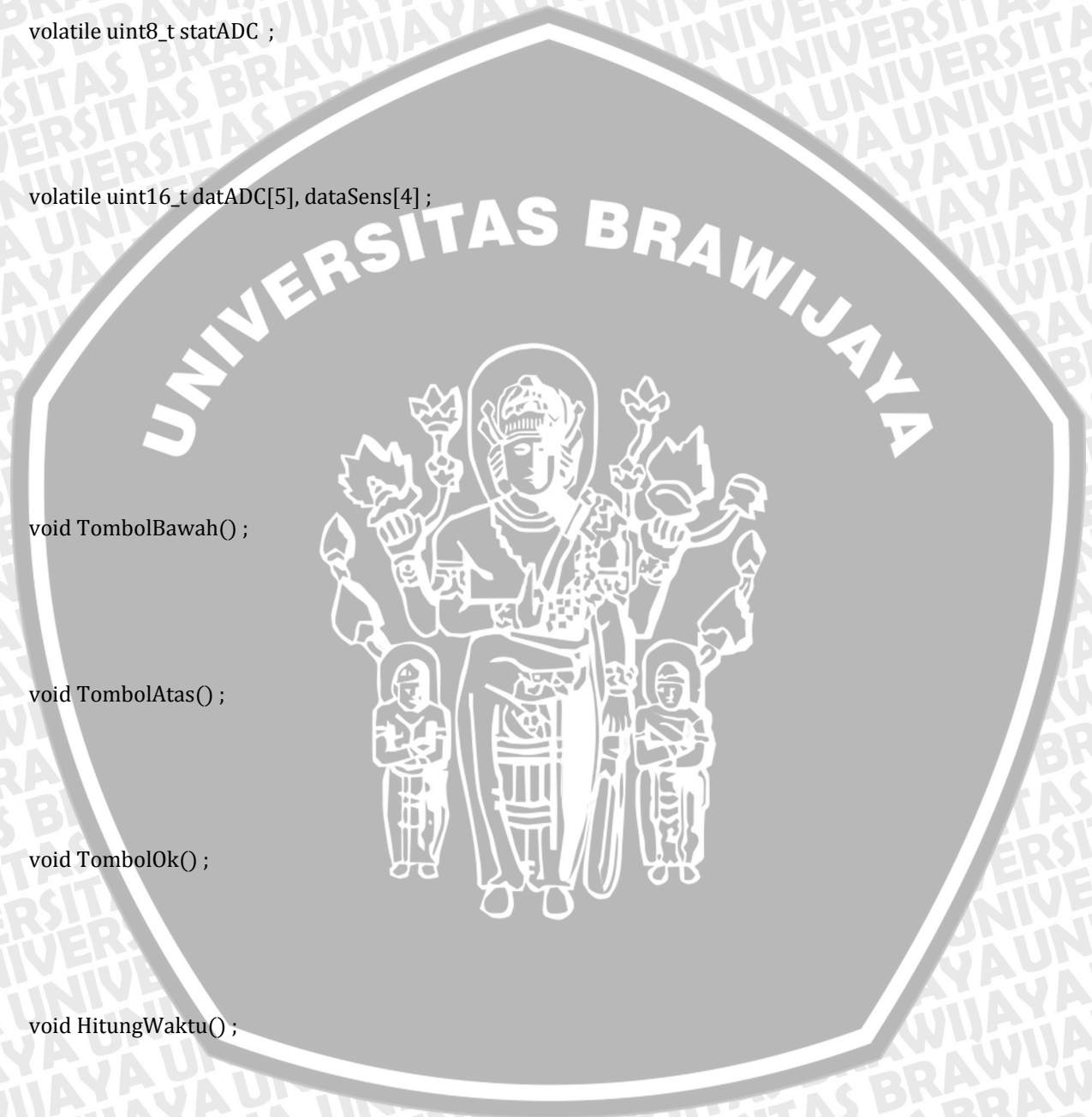
```
void TombolBawah() ;
```

```
void TombolAtas() ;
```

```
void TombolOk() ;
```

```
void HitungWaktu() ;
```

```
void StartSimpanData() ;
```




```
ISR(TIMER1_OVF_vect)
```

```
{
```

```
TCNT1 = 0xC2F7; // reload Timer1
```

```
detS ++;
```

```
if(detS>3)
```

```
{
```

```
detS = 0; detik ++;
```

```
updateDisp = 1; updateData = 1;
```

```
if(detik>59)
```

```
{
```

```
    menit ++ ; detik = 0 ;
```

```
}
```

```
}
```

```
if(bit_is_clear(TOMBOL,BAWAH)) TombolBawah();
```

```
else if(bit_is_clear(TOMBOL,ATAS)) TombolAtas();
```

```
else if(bit_is_clear(TOMBOL,OK)) statT_Ok=1 ;
```




```
else if((statT_Ok==1)&&(bit_is_set(TOMBOL,OK))) TombolOk() ;
```

```
}
```

```
ISR(ADC_vect)
```

```
{
```

```
asm volatile("nop\n\t"
```

```
"nop\n\t"
```

```
::);
```

```
// statADC = 1 ;
```



}

void TombolBawah()

{

if(menu<MMAKS) { menu++; if(menu>=MMAKS) menu = 0; }

updateDisp=1;

}

void TombolAtas()



{

if(menu<MMAKS) { menu-- ; if(menu>=MMAKS) menu=(MMAKS-1) ; }

updateDisp=1 ;

}

void TombolOk()

{

if(menu<MMAKS)

{

UNIVERSITAS BRAWIJAYA



```
menu += MMAKS ;
```

```
if(menu>=(2*MMAKS)) menu=0 ;
```

```
statDispAll = 255 ; wDisp = 255 ;
```

```
if(menu==M_END_LOG) { statLog = 0 ; }
```

```
else if(menu>=MMAKS)
```

```
{
```

```
if(statLog==0) StartSimpanData() ;
```

```
statLog = 1 ;
```

```
}
```



else

{

if(menu==M_END_LOG){ statLog = 0 ; }

menu -= MMAKS ;

}

statT_Ok = 0 ;

updateDisp=1 ;



}

void UARTSend(unsigned char *SrcDat, uint8_t jmlhDat)

{

uint8_t countDat;

for(countDat=0;countDat<jmlhDat;countDat++)

{

while(bit_is_clear(UCSRA,UDRE));

UDR = * (SrcDat +countDat);

}

UNIVERSITAS BRAWIJAYA



}

void InitEeprom()

{

uint8_t countEdata, tempEData, tempEData2 ;

// ===== Write Header Untuk Data CO2 =====

for(countEdata=0;countEdata<3;countEdata++)

{

tempEData2 = 1 ;



```
while(tempEData2==1)
```

```
{
```

```
tempEData = pgm_read_byte(&flashHCO2[countEdata]);
```

```
eeeprom_busy_wait();
```

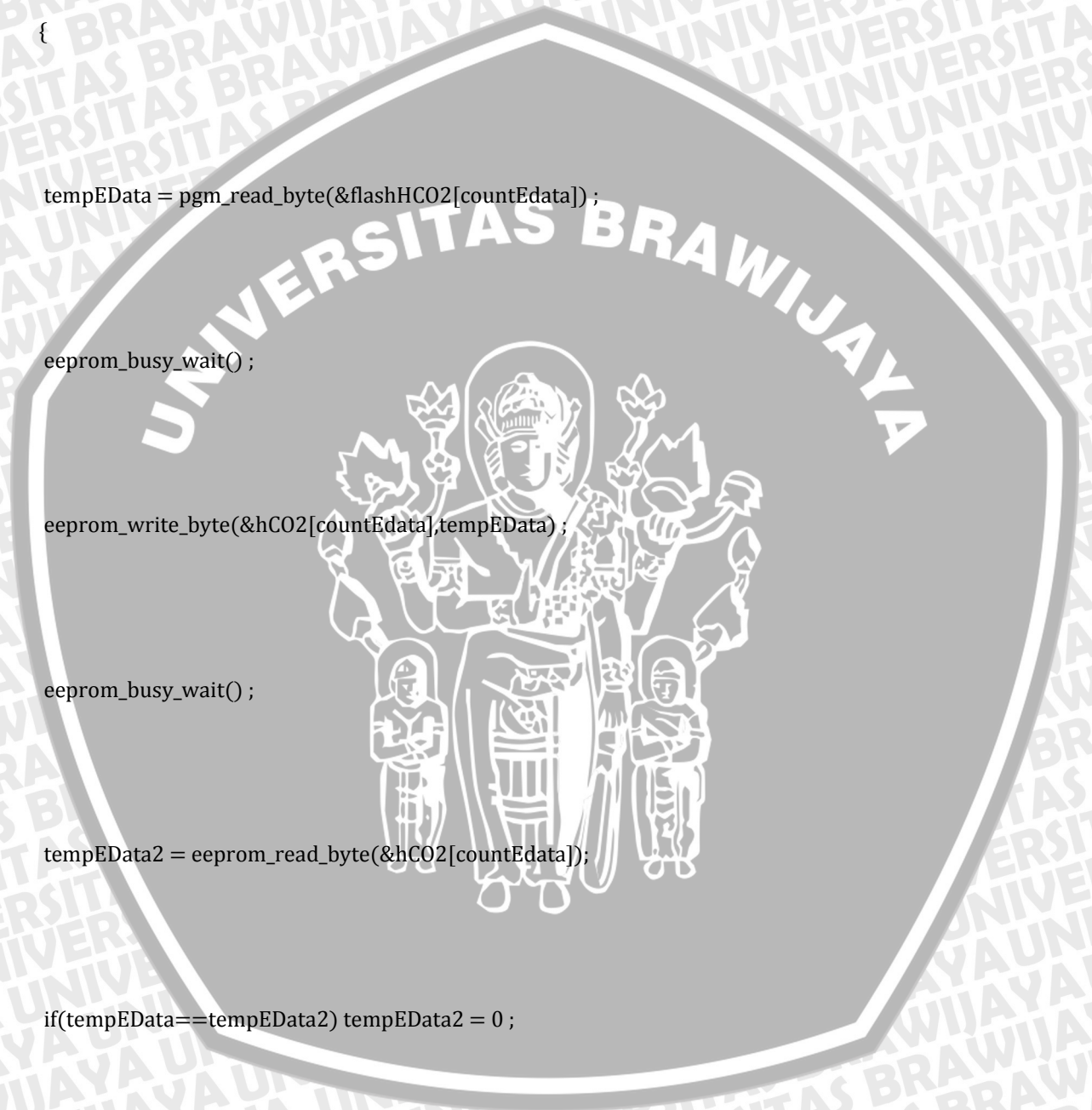
```
eeeprom_write_byte(&hCO2[countEdata],tempEData);
```

```
eeeprom_busy_wait();
```

```
tempEData2 = eeeprom_read_byte(&hCO2[countEdata]);
```

```
if(tempEData==tempEData2) tempEData2 = 0;
```

```
else tempEData2 = 1;
```




```
tempEData2 = 1 ;
```

```
while(tempEData2==1)
```

```
{
```

```
  eeprom_busy_wait() ;
```

```
  eeprom_write_byte(&eJmlDatC02,0);
```

```
  eeprom_busy_wait() ;
```

```
  tempEData = eeprom_read_byte(&eJmlDatC02) ;
```



```
if(tempEData==0) tempEData2 = 0 ;
```

```
else tempEData2 = 1 ;
```

```
}
```

```
// ===== Write Header Untuk Data 02 =====
```

```
for(countEdata=0;countEdata<2;countEdata++)
```

```
{
```

```
tempEData2 = 1 ;
```

```
while(tempEData2==1)
```

```
{
```




```
tempEData = pgm_read_byte(&flashHO2[countEdata]);
```

```
    eeprom_busy_wait();
```

```
    eeprom_write_byte(&hO2[countEdata],tempEData);
```

```
    eeprom_busy_wait();
```

```
    tempEData2 = eeprom_read_byte(&hO2[countEdata]);
```

```
    if(tempEData==tempEData2) tempEData2 = 0;
```

```
    else tempEData2 = 1;
```

```
}
```

```
}
```

UNIVERSITAS BRAWIJAYA



```
tempEData2 = 1 ;
```

```
while(tempEData2==1)
```

```
{
```

```
    eeprom_busy_wait() ;
```

```
    eeprom_write_byte(&eJmlDat02,0);
```

```
    eeprom_busy_wait() ;
```

```
    tempEData = eeprom_read_byte(&eJmlDat02);
```

```
    if(tempEData==0) tempEData2 = 0 ;
```

```
    else tempEData2 = 1 ;
```




```
void StartSimpanData()
```

```
{
```

```
uint8_t statWrite, tempEData ;
```

```
InitEeprom() ;
```

```
//===== Simpan Menit ketika Log dimulai=====
```

```
do
```



```
{
```

```
    eeprom_busy_wait() ;
```

```
    eeprom_write_byte(&menitStart,menit) ;
```

```
    eeprom_busy_wait() ;
```

```
    tempEData = eeprom_read_byte(&menitStart) ;
```

```
    if(tempEData==menit) statWrite = 0 ;
```

```
    else statWrite = 1 ;
```

```
    } while(statWrite==1) ;
```

```
//===== Simpan Jam ketika Log dimulai=====
```


do

{

 eeprom_busy_wait() ;

 eeprom_write_byte(&jamStart,jam) ;

 eeprom_busy_wait() ;

 tempEData = eeprom_read_byte(&jamStart) ;

 if(tempEData==jam) statWrite = 0 ;

 else statWrite = 1 ;

} while(statWrite==1) ;



}

void SimpanData()

{

uint8_t statWrite, tempEData2;

uint16_t tempEData;

// ===== Simpan Data CO2 =====

if(jmlDatCO2<=DAT_CO2_MAKS)




```
{
// statWrite = 1 ;
```

```
do
```

```
{
```

```
    eeprom_busy_wait() ;
```

```
    eeprom_write_word(&datCO2[jmlDatCO2],datADC[1]) ;
```

```
    eeprom_busy_wait() ;
```

```
    tempEData = eeprom_read_word(&datCO2[jmlDatCO2]) ;
```

```
    if(tempEData==datADC[1]) statWrite = 0 ;
```



```
else statWrite = 1 ;
```

```
} while(statWrite==1) ;
```

```
jmlDatCO2 ++ ;
```

```
do
```

```
{
```

```
  eeprom_busy_wait() ;
```

```
  eeprom_write_byte(&jmlDatCO2,jmlDatCO2) ;
```

```
  eeprom_busy_wait() ;
```




```
tempEData2 = eeprom_read_byte(&jmlDatC02);
```

```
if(tempEData2==jmlDatC02) statWrite = 0;
```

```
else statWrite = 1;
```

```
} while(statWrite==1);
```

```
}
```

```
// ===== Simpan Data O2 =====
```

```
if(jmlDatO2<=DAT_O2_MAKS)
```

```
{
```

```
// statWrite = 1;
```



do

{

 eeprom_busy_wait() ;

 eeprom_write_word(&dat02[jmlDat02],datADC[2]) ;

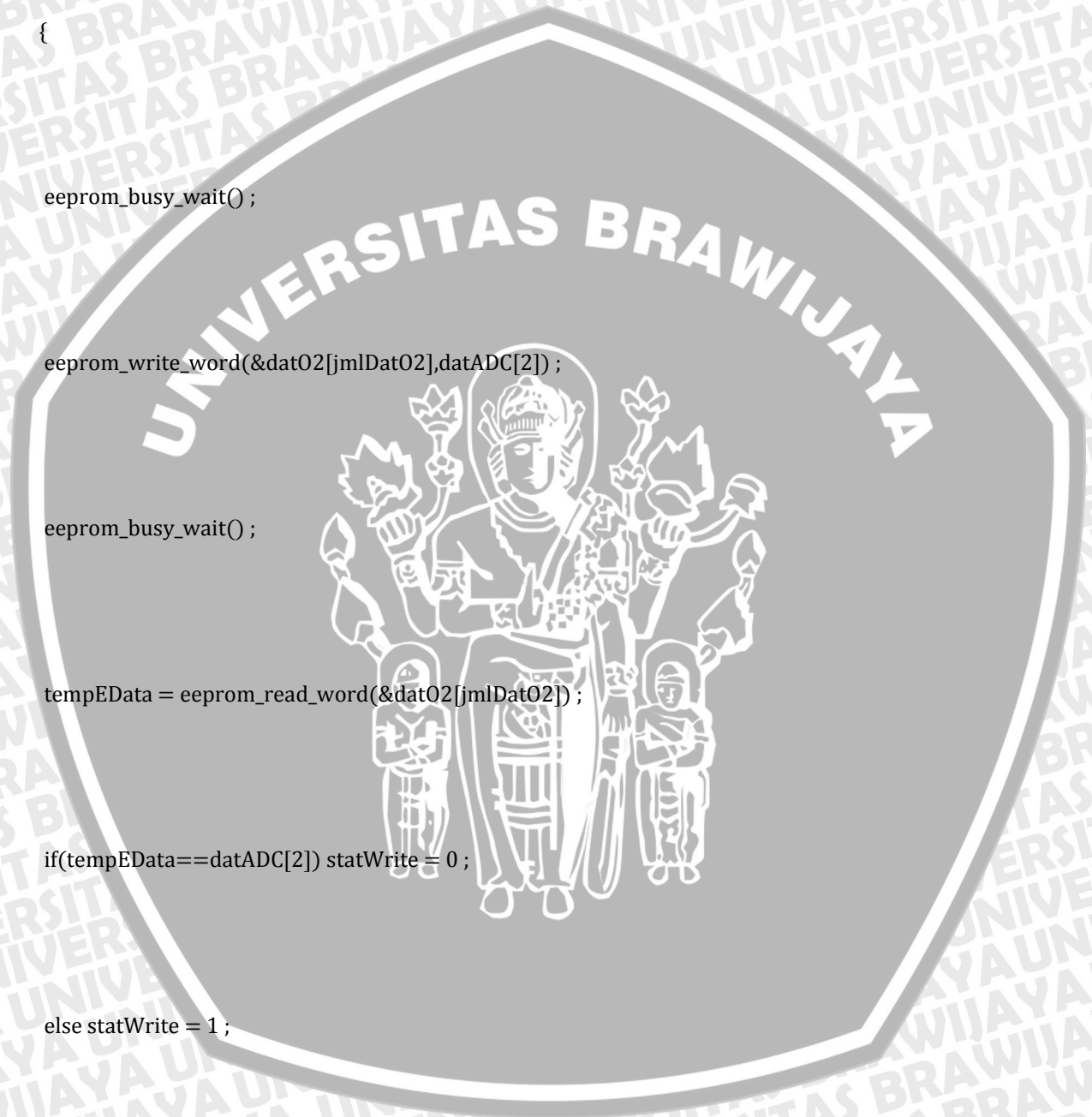
 eeprom_busy_wait() ;

 tempEData = eeprom_read_word(&dat02[jmlDat02]) ;

 if(tempEData==datADC[2]) statWrite = 0 ;

 else statWrite = 1 ;

} while(statWrite==1) ;




```
jmlDat02 ++ ;
```

```
// statWrite = 1 ;
```

```
do
```

```
{
```

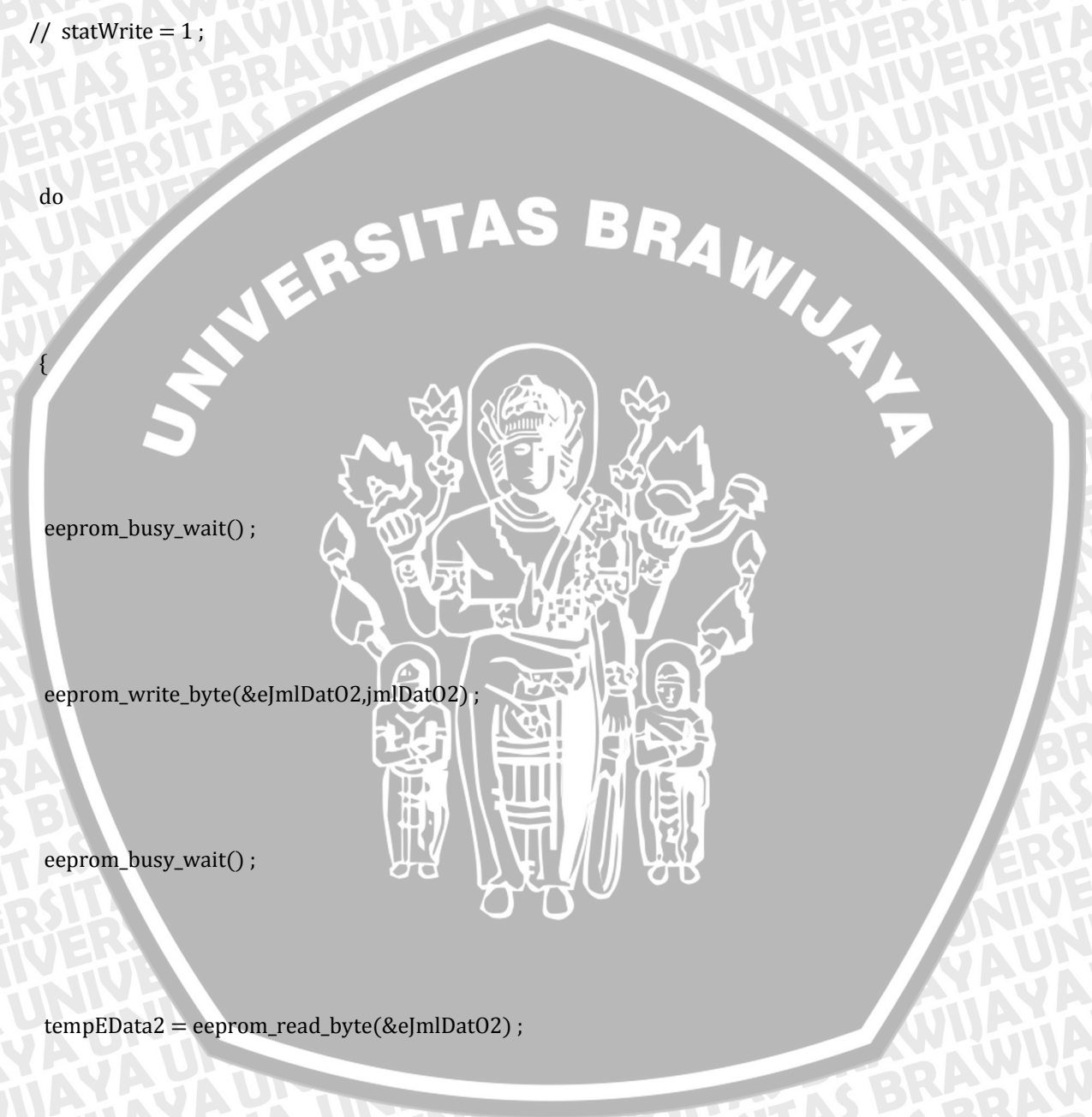
```
    eeprom_busy_wait() ;
```

```
    eeprom_write_byte(&ejmlDat02,jmlDat02) ;
```

```
    eeprom_busy_wait() ;
```

```
    tempEData2 = eeprom_read_byte(&ejmlDat02) ;
```

```
    if(tempEData2==jmlDat02) statWrite = 0 ;
```



```
else statWrite = 1 ;
```

```
} while(statWrite==1);
```

```
}
```

```
// =====
```

```
menitLog = menit + T_LOG ; // statLog = 1 ;
```

```
// if(menitLog>59) menitLog -= 60 ;
```

```
}
```

```
void ProsesDisplay()
```




```
{
```

```
uint8_t dmenu, dat[5], tempdat ;
```

```
uint16_t data = 0 ;
```

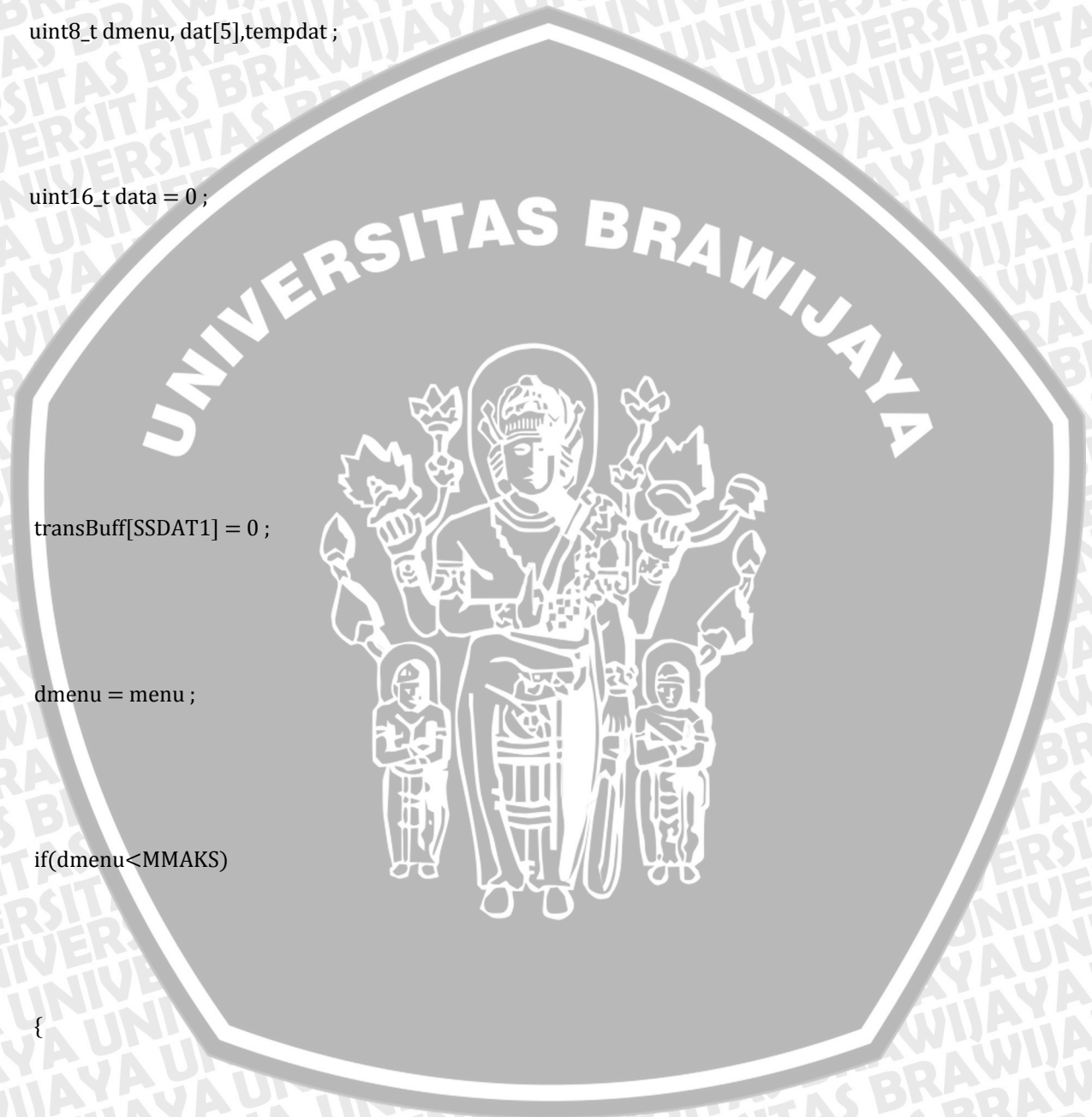
```
transBuff[SSDAT1] = 0 ;
```

```
dmenu = menu ;
```

```
if(dmenu<MMAKS)
```

```
{
```

```
for(tempdat=0;tempdat<DMENU_MAKS;tempdat++) {
```



```
transBuff[SSDAT0+tempdat] = pgm_read_byte(&menu_array[dmenu][4-tempdat]);
```

```
}
```

```
}
```

```
else if(dmenu==M_END_LOG)
```

```
{
```

```
transBuff[SSDAT4] = SSGMNDAT_F;
```

```
transBuff[SSDAT3] = SSGMNDAT_1;
```

```
transBuff[SSDAT2] = SSGMNDAT_n;
```

```
transBuff[SSDAT1] = SSGMNDAT_1;
```




```
transBuff[SSDAT0] = SSGMNDAT_5 ;
```

```
}
```

```
else
```

```
{
```

```
transBuff[SSDAT4] =0;
```

```
if(dmenu==MDISPLAY_CO2)
```

```
{
```

```
data = dataSens[1];
```

```
}
```

UNIVERSITAS BRAWIJAYA



```
else if(dmenu==MDISPLAY_02)
```

```
{
```

```
data = dataSens[2] ;
```

```
transBuff[SSDAT1] = SSGMNDAT_DOT ;
```

```
}
```

```
else // if(dmenu==M_ALL)
```

```
{
```

```
if(statDispAll<=DISP_C02)
```

```
{
```




```
data = dataSens[1];
```

```
transBuff[SSDAT4] = SSGMNDAT_b;
```

```
}
```

```
else if(statDispAll<=DISP_O2)
```

```
{
```

```
data = dataSens[2];
```

```
transBuff[SSDAT1] = SSGMNDAT_DOT;
```

```
transBuff[SSDAT4] = SSGMNDAT_C;
```

```
}
```



```
dat[0]=0; dat[1]=0; dat[2]=0;
```

```
dat[3]=0; dat[4]=0;
```

```
while(data>=10000)
```

```
{
```

```
data=data-10000;
```

```
dat[4]+=1;
```

```
};
```

```
while(data>=1000)
```



{

data=data-1000;

dat[3]+=1;

};

while(data>=100)

{

data-=100;

dat[2]+=1;

};

UNIVERSITAS BRAWIJAYA



```
while(data>=10)
```

```
{
```

```
data-=10;
```

```
dat[1]+=1;
```

```
};
```

```
dat[0] = (uint8_t)data ;
```

```
//=====
```

```
transBuff[SSDAT3] = pgm_read_byte(&segmenarray[dat[3]]);
```

```
transBuff[SSDAT2] = pgm_read_byte(&segmenarray[dat[2]]);
```




```
transBuff[SSDAT1] |= pgm_read_byte(&segmenarray[dat[1]]);
```

```
transBuff[SSDAT0] = pgm_read_byte(&segmenarray[dat[0]]);
```

```
if (dat[3]==0)
```

```
{
```

```
transBuff[SSDAT3]=0;
```

```
if (dat[2]==0) transBuff[SSDAT2]=0;
```

```
}
```

```
}
```

UNIVERSITAS BRAWIJAYA



}

void UpdtDisp()

{

transBuff[0] = PACKETSTART;

transBuff[1] = PACKETSTART ;

ProsesDisplay() ;

transBuff[7] = 0 ;

transBuff[8] = 0 ;

UNIVERSITAS BRAWIJAYA




```
transBuff[9] = PACKETFINISH ;
```

```
transBuff[10] = PACKETFINISH ;
```

```
UARTSend((void*)transBuff, 11) ;
```

```
}
```

```
float InttoFloat(uint16_t IntData)
```

```
{
```

```
float tempfloat ;
```



```
tempfloat = 0 ;
```

```
while(IntData>=10000)
```

```
{
```

```
IntData=IntData-10000;
```

```
tempfloat +=10000;
```

```
};
```

```
while(IntData>=1000)
```

```
{
```

```
IntData=IntData-1000;
```

UNIVERSITAS BRAWIJAYA




```
tempfloat +=1000 ;
```

```
};
```

```
while(IntData>=100)
```

```
{
```

```
IntData -=100;
```

```
tempfloat +=100;
```

```
};
```

```
while(IntData>=10)
```

```
{
```



```
IntData -=10;
```

```
tempfloat +=10;
```

```
};
```

```
while(IntData>0)
```

```
{
```

```
IntData -=1;
```

```
tempfloat +=1;
```

```
};
```

```
return tempfloat ;
```



}

void BacaADC()

ADCtemp[0] =0 ;

ADCtemp[1] =0 ;

ADCtemp[2] =0 ;

ADCtemp[3] =0 ;

for(countadc=0;countadc<256;countadc++)

{

UNIVERSITAS BRAWIJAYA



```
ADMUX = ADC0 ;
```

```
ADCSRA |= _BV(ADSC) ;
```

```
while(bit_is_set(ADCSRA,ADSC)) ;
```

```
ADCSRA |= _BV(ADSC) ;
```

```
sleep_mode() ;
```

```
while(bit_is_set(ADCSRA,ADSC))
```

```
sleep_mode() ;
```

```
ADCtemp[0] += ADCW ;
```

```
{
```




```
ADMUX = ADC1 ;
```

```
ADCSRA |= _BV(ADSC) ;
```

```
while(bit_is_set(ADCSRA,ADSC)) ;
```

```
ADCSRA |= _BV(ADSC) ;
```

```
sleep_mode() ;
```

```
while(bit_is_set(ADCSRA,ADSC))
```

```
sleep_mode() ;
```

```
ADCtemp[1] += ADCW ;
```

```
//=====
```



```
_delay_us(100);
```

```
}
```

```
datADC[1] = (uint16_t)((ADCtemp[1]/256)); // data CO2
```

```
datADC[2] = (uint16_t)((ADCtemp[2]/256)); // data O2
```

```
}
```

```
int main()
```

```
{
```

```
float x,y;
```

```
_delay_ms(500);
```

UNIVERSITAS BRAWIJAYA



//===== inisialisasi =====

ACSR = (1<<ACD);

PORTB = ~((1<<6)|(1<<2)|(1<<1));

DDRB = ((1<<6)|(1<<2)|(1<<1));

PORTC = 0b10000000 ;

DDRC = 0;



PORTD = 255 ;

DDRD = 0 ;

UCSRB = ((1<<RXCIE)|(0<<TXCIE)|(1<<RXEN)|(1<<TXEN)) ;

UCSRC = 0b10000110 ;

TCCR1B = 3 ;

TIMSK = 4 ;

TCNT1 = 0xB000 ;



TIFR = 4 ;

menu = 0 ;

_delay_ms(500) ;

//=====

UpdtDisp() ;

sei();

while(1)

{

if(updateData)



{

BacaADC() ;

//===== PERHITUNGAN KADAR CO2=====

x = InttoFloat(datADC[1]) ;

// CO2 satuan ppm

x=0.0077*x;

x=pow(2.71828,x);

y=19.705*x;




```
y += 0.4 ;
```

```
dataSens[1] = (uint16_t) y ;
```

```
//=====PERHITUNGAN KADAR O2=====
```

```
x = InttoFloat(datADC[2]) ;
```

```
// O2 satuan %
```

```
y = 0.0991*x - 1.318 ;
```

```
if(y<0) y = 0 ;
```

```
y = y*10 ;
```

```
y += 0.4 ;
```



```
dataSens[2] = (uint16_t) y ;
```

```
//=====
```

```
if((menit==menitLog)&&(statLog==1))
```

```
{ SimpanData(); }
```

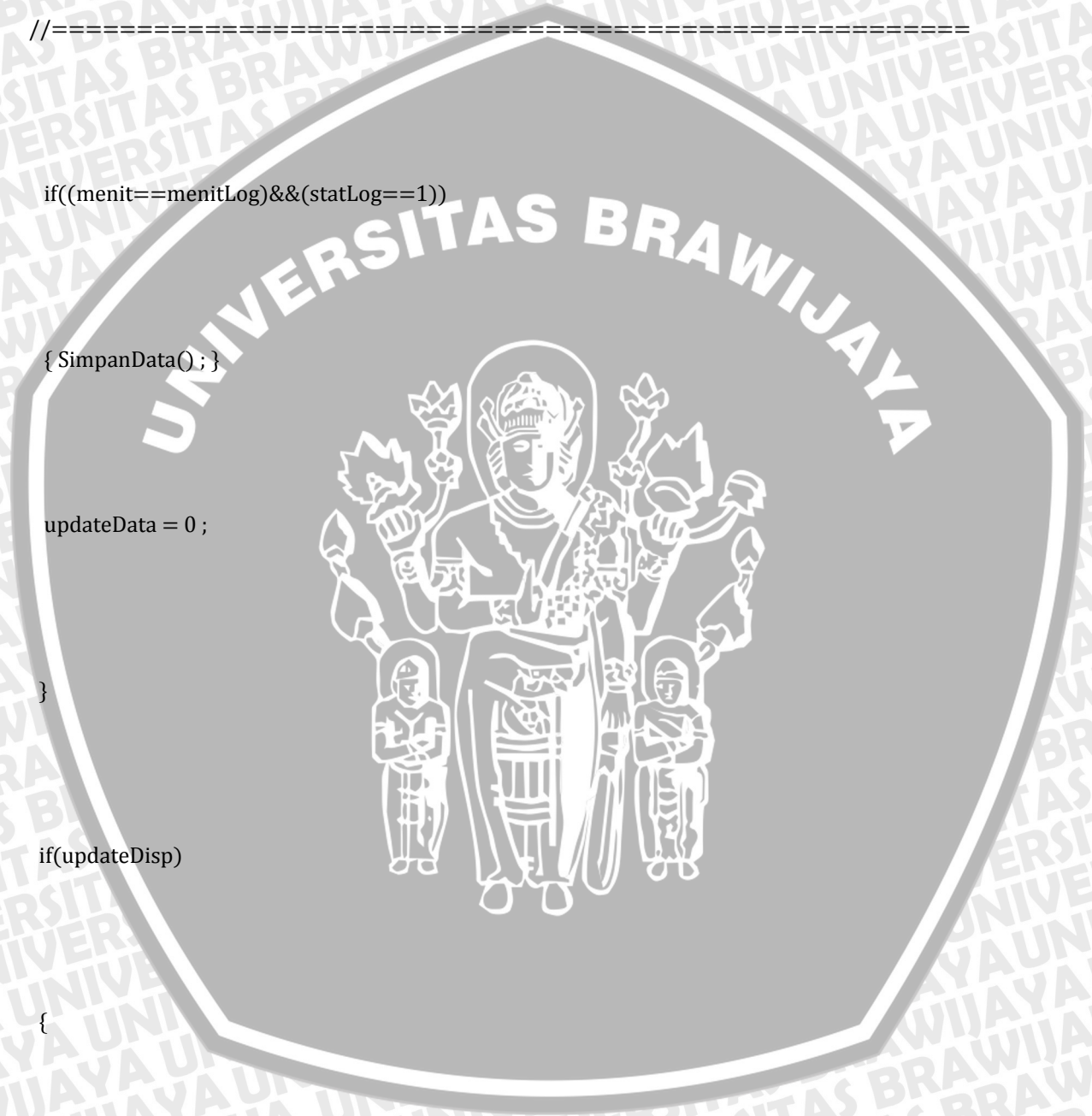
```
updateData = 0 ;
```

```
}
```

```
if(updateDisp)
```

```
{
```

```
updateDisp = 0 ; UpdtDisp() ;
```




```
sleep_mode();
```

```
};
```

```
}
```

UNIVERSITAS BRAWIJAYA



\\Program Mendefinisikan pin seven segmen\\(data7segmn.h)

```
#define PIN_A 7
```



#define PIN_B 6

#define PIN_C 4

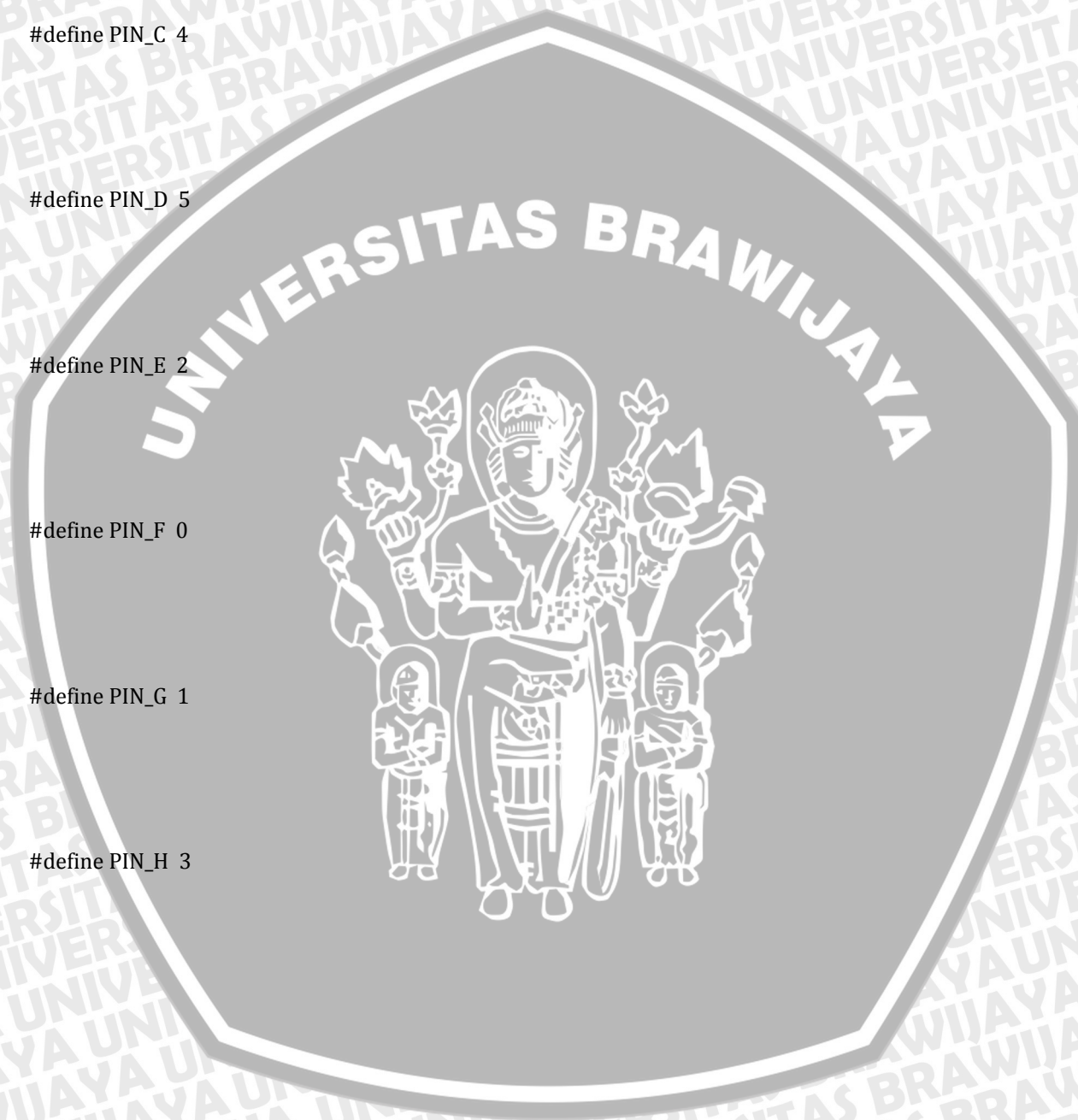
#define PIN_D 5

#define PIN_E 2

#define PIN_F 0

#define PIN_G 1

#define PIN_H 3




```
#define SSGMNDAT_0 ((1<<PIN_F)|(1<<PIN_E)|(1<<PIN_D)|(1<<PIN_C)|(1<<PIN_B)|
```

```
(1<<PIN_A))
```

```
#define SSGMNDAT_1 ((1<<PIN_C)|(1<<PIN_B))
```

```
#define SSGMNDAT_2 ((1<<PIN_G)|(1<<PIN_E)|(1<<PIN_D)|(1<<PIN_B)|(1<<PIN_A))
```

```
#define SSGMNDAT_3 ((1<<PIN_G)|(1<<PIN_D)|(1<<PIN_C)|(1<<PIN_B)|(1<<PIN_A))
```

```
#define SSGMNDAT_4 ((1<<PIN_G)|(1<<PIN_F)|(1<<PIN_C)|(1<<PIN_B))
```



```
#define SSGMNDAT_5 ((1<<PIN_G)|(1<<PIN_F)|(1<<PIN_D)|(1<<PIN_C)|(1<<PIN_A))
```

```
#define SSGMNDAT_6 ((1<<PIN_G)|(1<<PIN_F)|(1<<PIN_E)|(1<<PIN_D)|(1<<PIN_C)|
```

```
(1<<PIN_A))
```

```
#define SSGMNDAT_7 ((1<<PIN_C)|(1<<PIN_B)|(1<<PIN_A))
```

```
#define SSGMNDAT_8 ((1<<PIN_G)|(1<<PIN_F)|(1<<PIN_E)|(1<<PIN_D)|(1<<PIN_C)|
```

```
(1<<PIN_B)|(1<<PIN_A))
```




```
#define SSGMNDAT_9 ((1<<PIN_G)|(1<<PIN_F)|(1<<PIN_D)|(1<<PIN_C)|(1<<PIN_B)|
```

```
(1<<PIN_A))
```

```
#define SSGMNDAT_A ((1<<PIN_G)|(1<<PIN_F)|(1<<PIN_E)|(1<<PIN_C)|(1<<PIN_B)|
```

```
(1<<PIN_A))
```

```
#define SSGMNDAT_b ((1<<PIN_G)|(1<<PIN_F)|(1<<PIN_E)|(1<<PIN_D)|(1<<PIN_C))
```

```
#define SSGMNDAT_C ((1<<PIN_F)|(1<<PIN_E)|(1<<PIN_D)|(1<<PIN_A))
```

```
#define SSGMNDAT_d ((1<<PIN_G)|(1<<PIN_E)|(1<<PIN_D)|(1<<PIN_C)|(1<<PIN_B))
```



```
#define SSGMNDAT_E ((1<<PIN_G)|(1<<PIN_F)|(1<<PIN_E)|(1<<PIN_D)|(1<<PIN_A))
```

```
#define SSGMNDAT_F ((1<<PIN_G)|(1<<PIN_F)|(1<<PIN_E)|(1<<PIN_A))
```

```
#define SSGMNDAT_H ((1<<PIN_G)|(1<<PIN_F)|(1<<PIN_E)|(1<<PIN_C)|(1<<PIN_B))
```

```
#define SSGMNDAT_h ((1<<PIN_G)|(1<<PIN_F)|(1<<PIN_E)|(1<<PIN_C))
```

```
#define SSGMNDAT_i (1<<PIN_C)
```




```
#define SSGMNDAT_J ((1<<PIN_D)|(1<<PIN_C)|(1<<PIN_B))
```

```
#define SSGMNDAT_L ((1<<PIN_F)|(1<<PIN_E)|(1<<PIN_D))
```

```
#define SSGMNDAT_I ((1<<PIN_F)|(1<<PIN_E))
```

```
#define SSGMNDAT_P ((1<<PIN_G)|(1<<PIN_F)|(1<<PIN_E)|(1<<PIN_B)|(1<<PIN_A))
```

```
#define SSGMNDAT_U ((1<<PIN_F)|(1<<PIN_E)|(1<<PIN_D)|(1<<PIN_C)|(1<<PIN_B))
```



```
#define SSGMNDAT_a ((1<<PIN_G)|(1<<PIN_E)|(1<<PIN_D)|(1<<PIN_C)|(1<<PIN_B)|
```

```
(1<<PIN_A))
```

```
#define SSGMNDAT_SP 0
```

```
#define SSGMNDAT_MIN (1<<PIN_G)
```

```
#define SSGMNDAT_DOT (1<<PIN_H)
```




```
#define SSGMNDAT_n ((1<<PIN_F)|(1<<PIN_E)|(1<<PIN_C)|(1<<PIN_B)|(1<<PIN_A))
```

```
#define SSGMNDAT_r ((1<<PIN_F)|(1<<PIN_E)|(1<<PIN_A))
```

```
#define PACKETSTART ((1<<PIN_H)|(1<<PIN_D))
```

```
#define PACKETFINISH ((1<<PIN_H)|(1<<PIN_C))
```

```
\\program tampilan menu\\ (gas_sensor_main.h)
```

#define ADC1 0xC1

#define ADC2 0xC2

#define ADCEN 0b10001101

#define OFFSETLM 3

#define MMAKS 4




```
#define MDISPLAY_CO2 MMAKS
```

```
#define MDISPLAY_O2 MMAKS+1
```

```
#define M_ALL MMAKS+2
```

```
#define M_END_LOG M_ALL +1
```

```
#define T_LOG
```

16

```
#define DAT_SUHU_MAKS
```

83

```
#define DAT_CO2_MAKS 83
```

```
#define DAT_O2_MAKS 83
```



#define TDISP

2

#define DISP_CO2

(TDISP)-1

#define DISP_O2

(TDISP*2)-1

#define TDISP_MAKS

(TDISP*2)-1

#define DMENU_MAKS

5

#define setPB 0

#define setDB 255

#define setPC 0

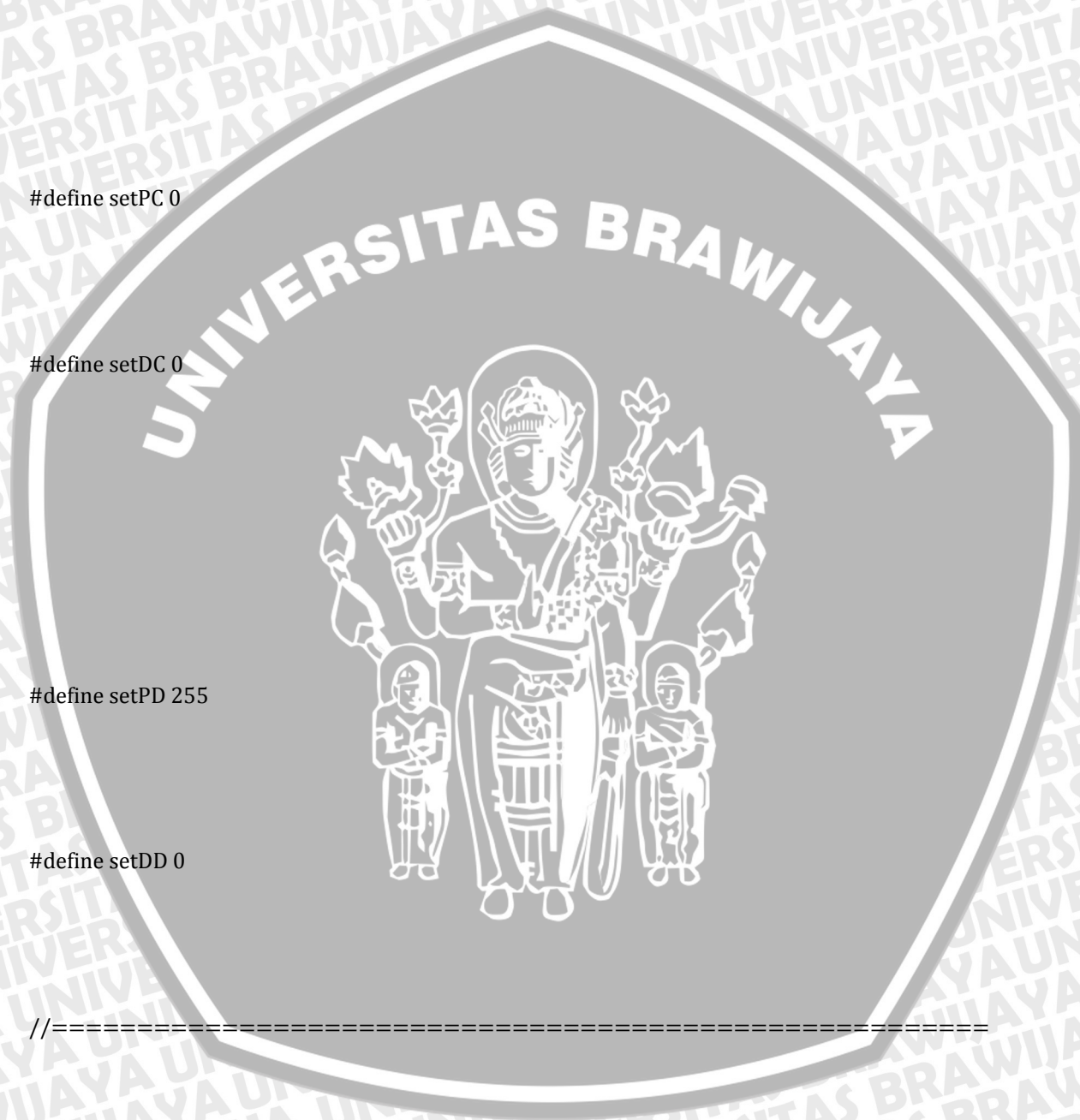
#define setDC 0

#define setPD 255

#define setDD 0

//=====

#define TOMBOL PIND



#define BAWAH 2

#define ATAS 3

#define OK 4

//=====

#define PFINISH 3

#define PSTART 1

#define SSDAT0 2

#define SSDAT1 3




```
#define SSDAT2      4
```

```
#define SSDAT3      5
```

```
#define SSDAT4      6
```

```
\\program main MK2\\(gas_sens_display.c)
```

```
#include <avr/io.h>
```

```
#include <avr\\iom8.h>
```

```
#include <avr/pgmspace.h>
```

```
#include <avr/interrupt.h>
```

```
#include <util/delay.h>
```

```
#include <avr/sleep.h>
```

```
#include "gas_sens_display.h"
```

```
#include "data7segmn.h"
```

```
volatile char segmndat[9];
```

```
volatile uint8_t transbuff[8], idrecv, packstatus, buffrecv[20];
```




```
ISR(USART_RXC_vect)
```

```
{
```

```
uint8_t RXCtemp, RXCtemp2, RXCid_temp ;
```

```
RXCtemp = 0 ;
```

```
RXCid_temp = idrecv ;
```

```
buffrecv[RXCid_temp] = UDR ;
```

```
if(RXCid_temp>0)
```



```
{
```

```
    RXCtemp = buffrecv[RXCid_temp] ; RXCtemp2 = buffrecv[RXCid_temp-1] ;
```

```
    RXCtemp2 -= RXCtemp ;
```

```
    RXCtemp += RXCtemp2 ;
```

```
}
```

```
if(((packstatus & PSTART)==0)&&(RXCtemp==PACKETSTART))
```

```
{
```

```
    packstatus = PSTART ;
```

```
    RXCid_temp = 255 ;
```




```
else if(((packstatus & PFINISH) == PSTART) && (RXCtemp == PACKETFINISH))
```

```
{
```

```
//=====
```

```
if(RXCid_temp == PACKETMAKS)
```

```
{
```

```
for(RXCtemp=0; RXCtemp < (PACKETMAKS-3); RXCtemp++)
```

```
{
```

```
segmndat[RXCtemp] = buffrecv[RXCtemp];
```

```
//=====
```

```
packstatus = 0 ; RXCid_temp = 255 ;
```

```
}
```

```
RXCid_temp ++ ;
```

```
idrecv = RXCid_temp ;
```

```
}
```




```
ISR(TIMER0_OVF_vect)
```

```
{
```

```
asm volatile("nop\n\t"
```

```
"nop\n\t"
```

```
::);
```

```
}
```

```
void initio()
```

```
{
```



ACSR = (1<<ACD) ;

PORTB = setPB ;

DDRB = setDB ; // 255 ;

PORTC = setPC ; // 255 ;

DDRC = setDC ; // 0b00111111 ;

PORTD = setPD ; // 0b11101111 ;




```
DDR = setDDR ; //0b10011100 ;
```

```
UCSRB = 0b10011000 ;
```

```
UCSRC = 0b10000110 ;
```

```
// MCUCR = 0 ; // idle ;
```

```
TCCR1B = 3 ;
```

```
TIMSK = 4 ;
```

```
TCNT1 = 0xC2F7 ;
```



```
sei();
```

```
// UBR = 0 ;
```

```
// UBRL = 0 ;
```

```
}
```

```
void start_up()
```

```
{
```

```
uint8_t countstart, n ;
```




```
_delay_ms(2) ;
```

```
countstart = 0 ;
```

```
while(countstart < 30) {
```

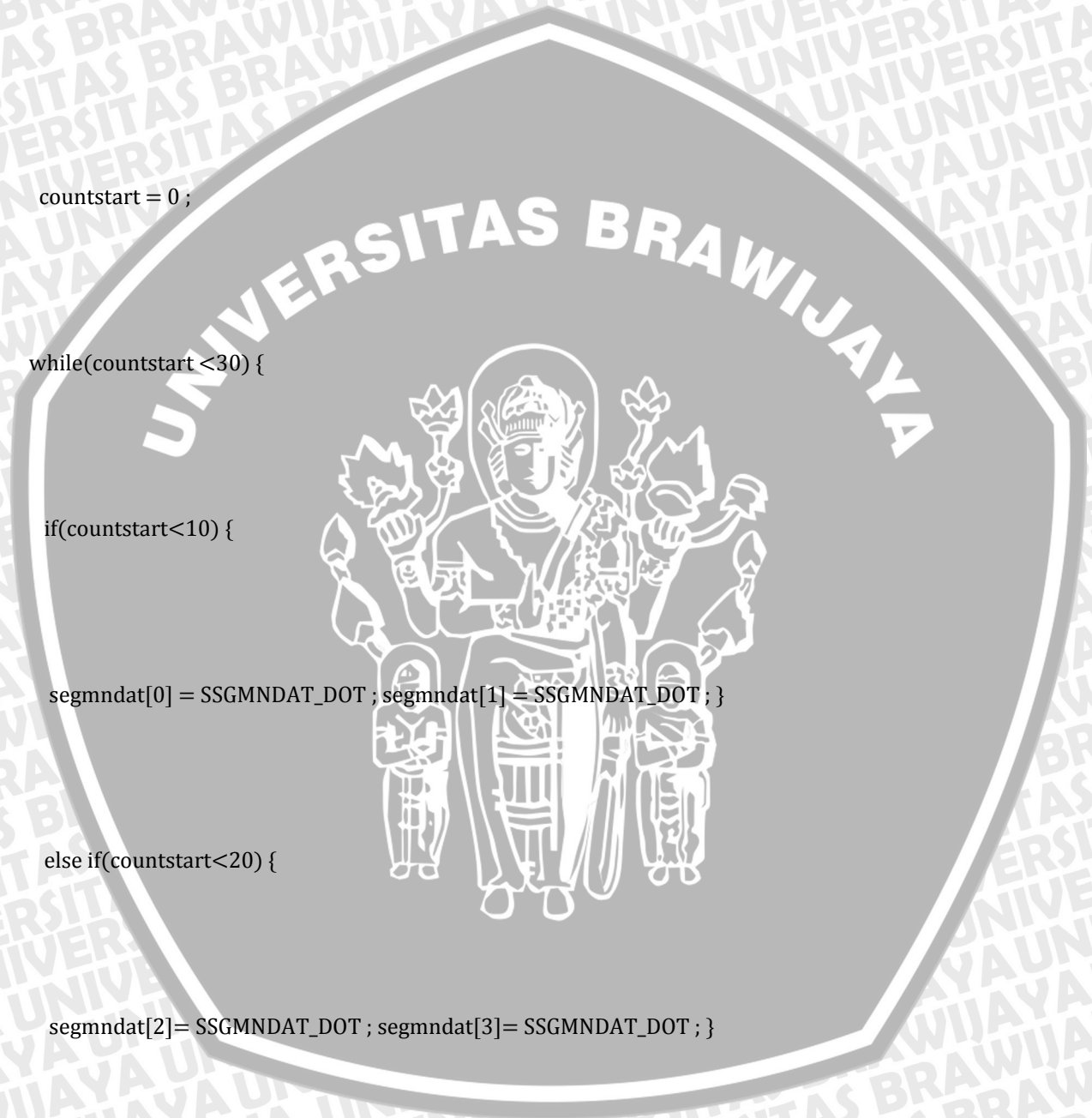
```
    if(countstart < 10) {
```

```
        segmndat[0] = SSGMNDAT_DOT ; segmndat[1] = SSGMNDAT_DOT ; }
```

```
    else if(countstart < 20) {
```

```
        segmndat[2] = SSGMNDAT_DOT ; segmndat[3] = SSGMNDAT_DOT ; }
```

```
    else if(countstart < 30) segmndat[4] = SSGMNDAT_DOT ;
```



```
for(n=0;n<5;n++) {
```

```
    Segmn4off() ;
```

```
    P_SS DAT = segmndat[0] ;
```

```
    Segmn0on() ;
```

```
    _delay_ms(5) ;
```

```
    Segmn0off() ;
```

```
    P_SS DAT = segmndat[1] ;
```



Segmn1on() ;

_delay_ms(5) ;

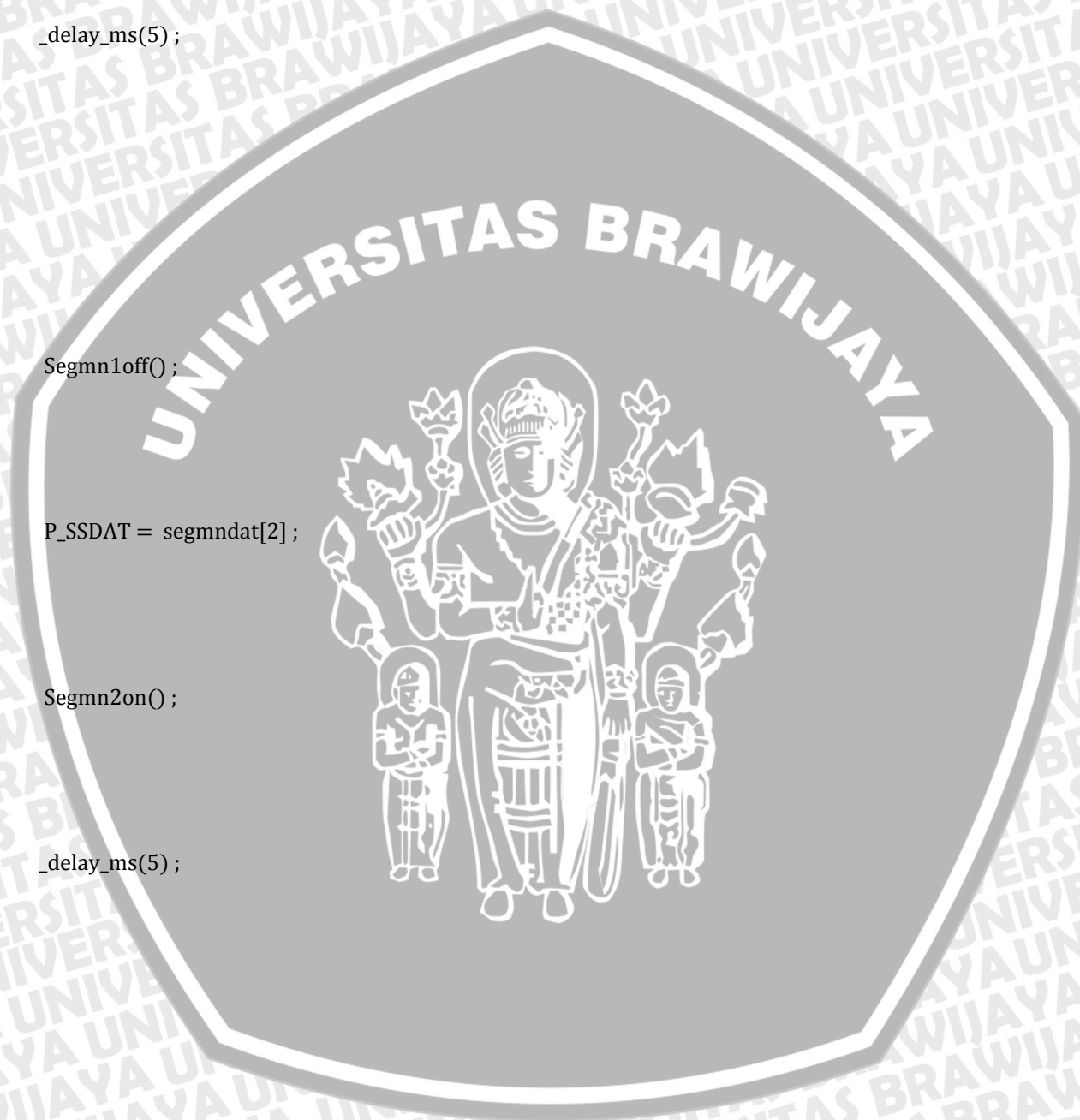
Segmn1off() ;

P_SS DAT = segmndat[2] ;

Segmn2on() ;

_delay_ms(5) ;

Segmn2off() ;



```
P_SS DAT = segmndat[3];
```

```
Segmn3on();
```

```
_delay_ms(5);
```

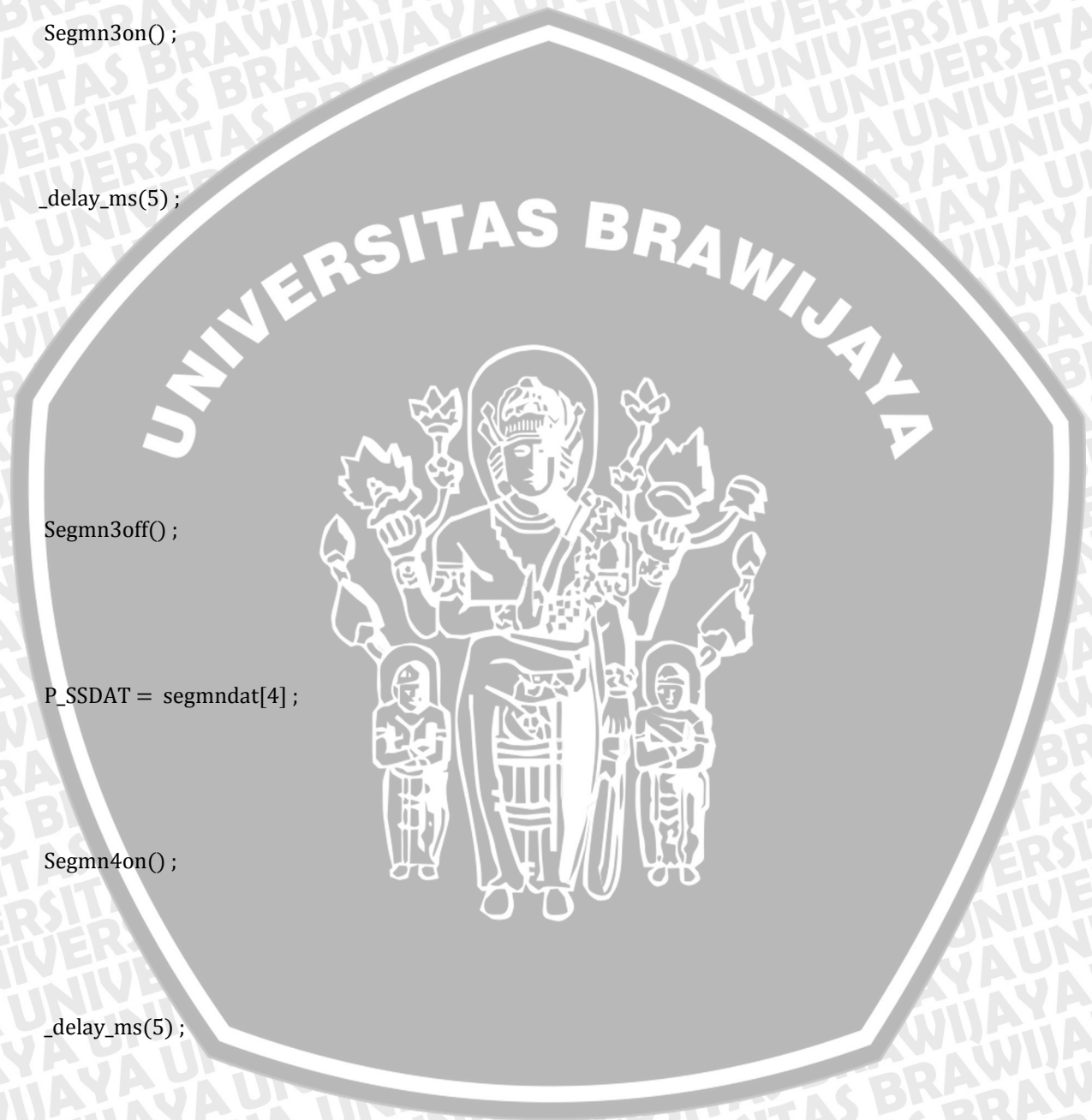
```
Segmn3off();
```

```
P_SS DAT = segmndat[4];
```

```
Segmn4on();
```

```
_delay_ms(5);
```

```
}
```




```
countstart ++ ;
```

```
};
```

```
}
```

```
int main()
```

```
{
```

```
// unsigned char n ;
```

```
_delay_ms(500) ;
```



initio();

start_up();

while(1)

{

Segmn4off();

P_SS DAT = segmndat[0] ;

Segmn0on();

_delay_ms(5);



Segmn0off() ;

P_SSDAT = segmndat[1] ;

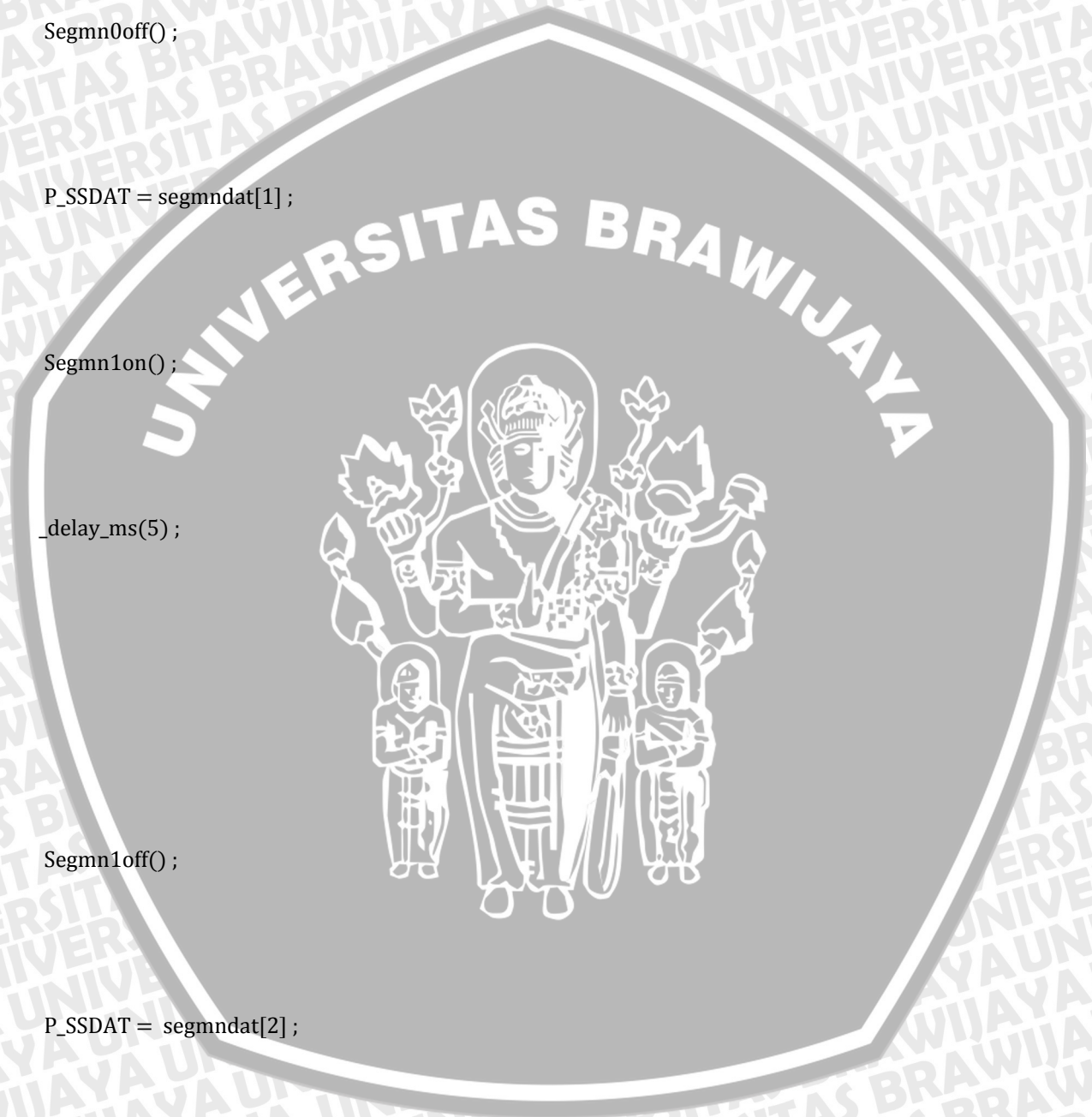
Segmn1on() ;

_delay_ms(5) ;

Segmn1off() ;

P_SSDAT = segmndat[2] ;

Segmn2on() ;



_delay_ms(5) ;

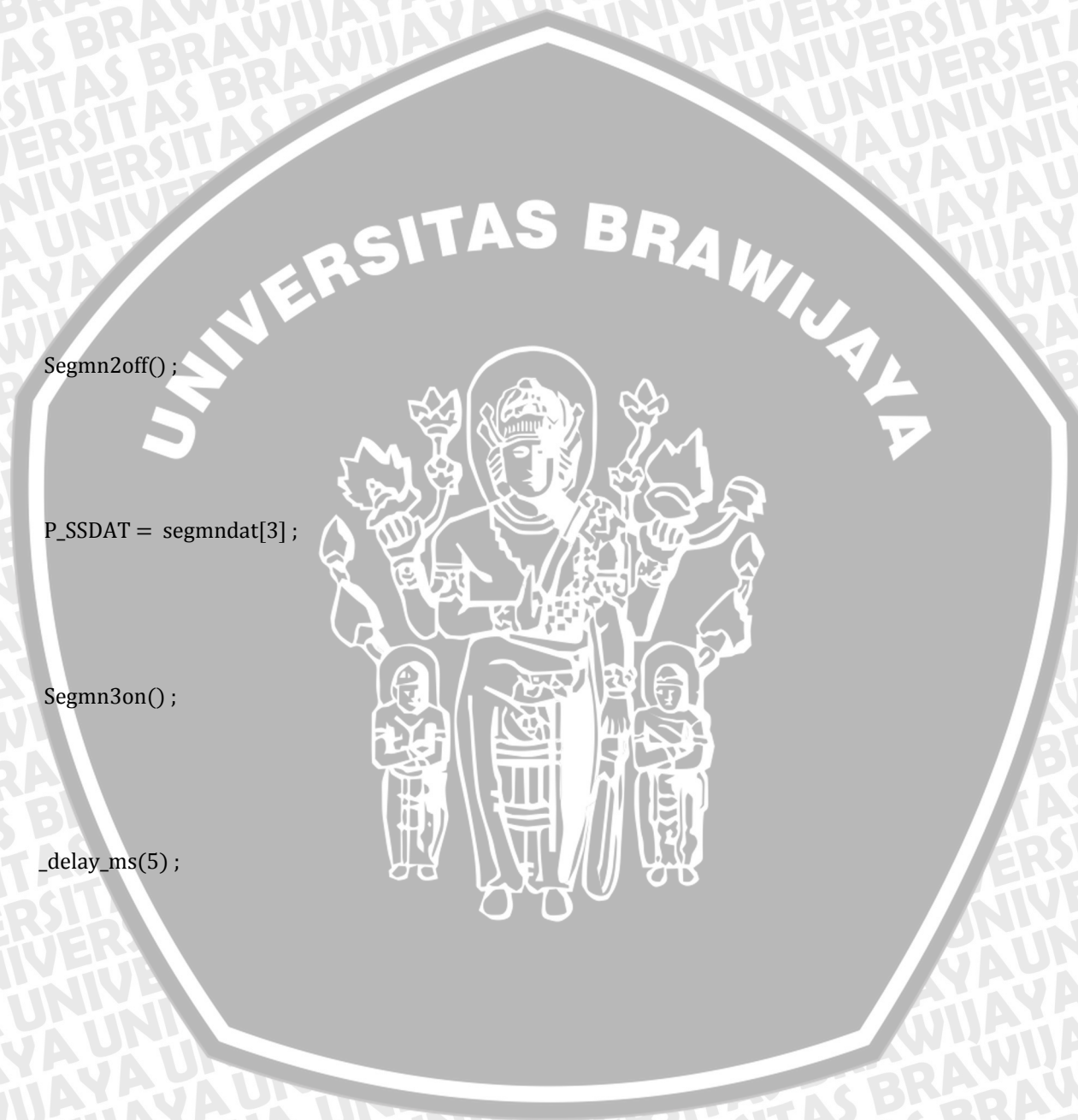
Segmn2off() ;

P_SS DAT = segmndat[3] ;

Segmn3on() ;

_delay_ms(5) ;

Segmn3off() ;




```
P_SS DAT = segmndat[4] ;
```

```
Segmn4on() ;
```

```
_delay_ms(5) ;
```

```
};
```

```
}
```

```
\\program.MK2\\ (gas_sens_display.h)
```



```
#define P_Led_bat PORTD
```

```
#define Led_batH 2
```

```
#define Led_batM 3
```

```
#define P_Tombol PIND
```

```
#define plh_up 7
```

```
#define plh_down 5
```

```
#define ok 6
```

```
#define setDB 0xFF
```



#define setDD 0b10000000

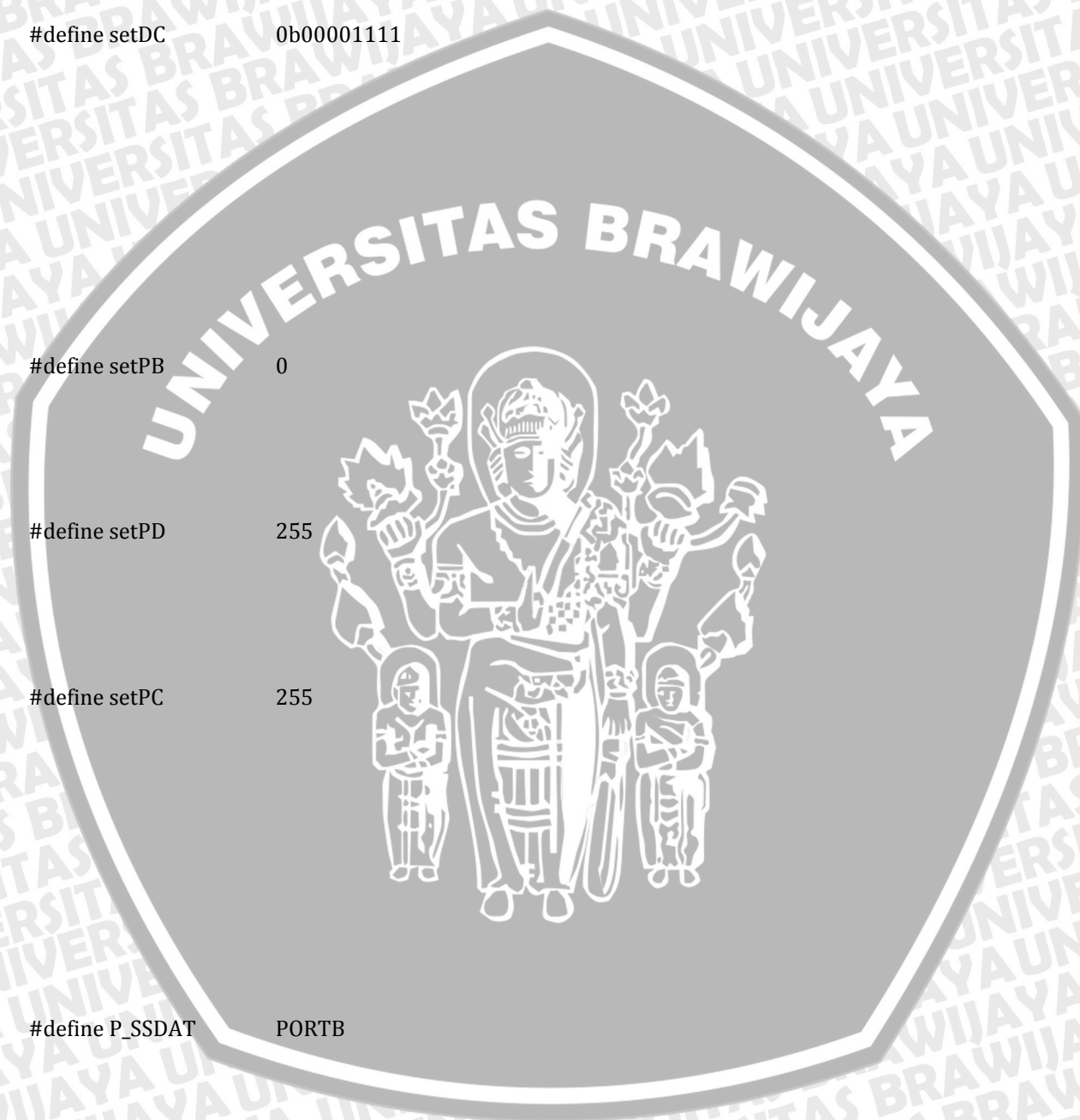
#define setDC 0b00001111

#define setPB 0

#define setPD 255

#define setPC 255

#define P_SS DAT PORTB



```
#define P_SSCTRL    PORTC
```

```
#define SEGMENT4    3
```

```
#define SEGMENT3    2
```

```
#define SEGMENT2    1
```

```
#define SEGMENT1    0
```

```
#define SEGMENT0    7
```

```
#define Segmn4on()  PORTC &= ~(1<<SEGMENT4)
```

```
#define Segmn3on()  PORTC &= ~(1<<SEGMENT3)
```




```
#define Segmn2on()    PORTC &= ~(1<<SEGMENT2)
```

```
#define Segmn1on()    PORTC &= ~(1<<SEGMENT1)
```

```
#define Segmn0on()    PORTD &= ~(1<<SEGMENT0)
```

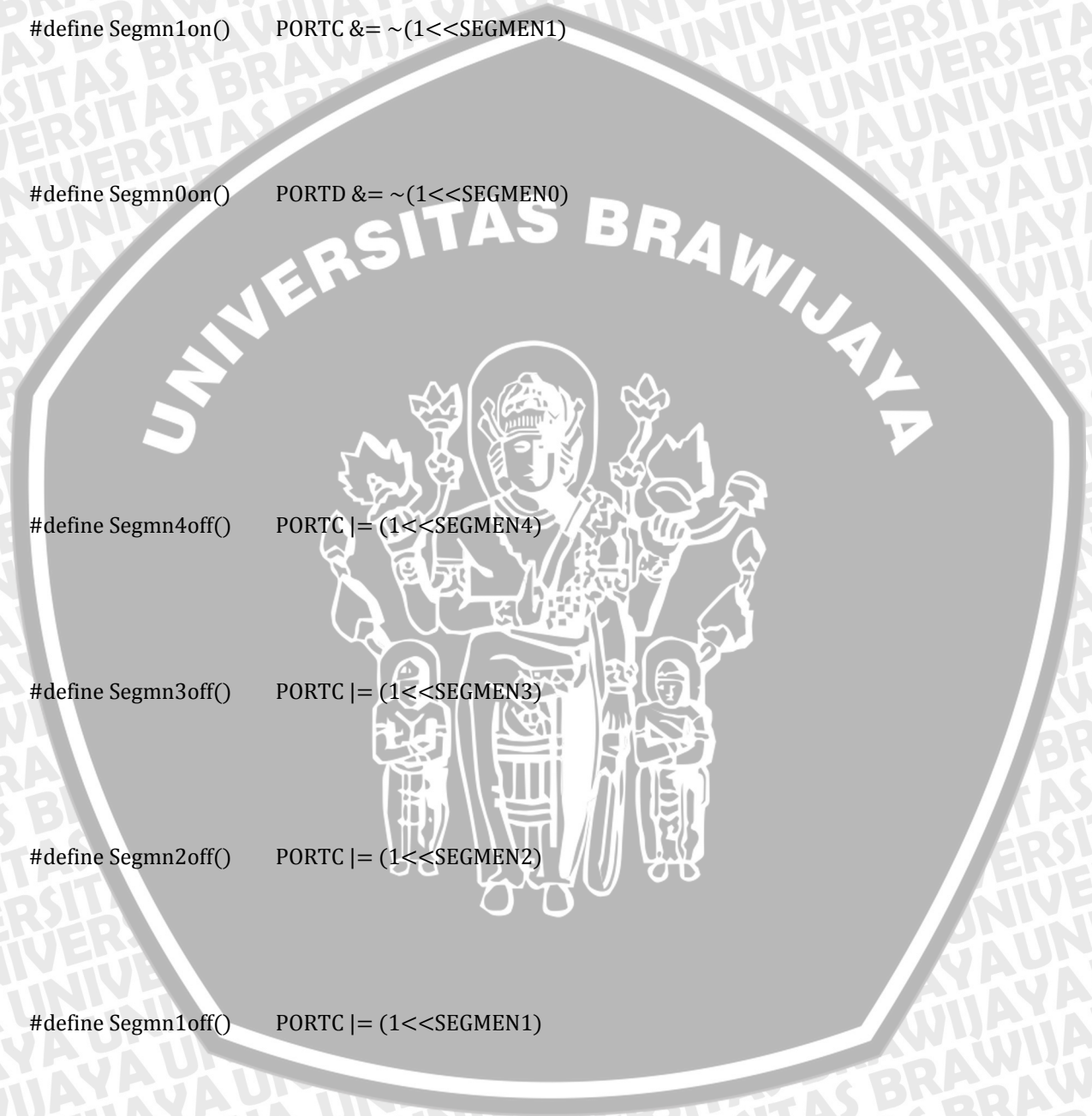
```
#define Segmn4off()   PORTC |= (1<<SEGMENT4)
```

```
#define Segmn3off()   PORTC |= (1<<SEGMENT3)
```

```
#define Segmn2off()   PORTC |= (1<<SEGMENT2)
```

```
#define Segmn1off()   PORTC |= (1<<SEGMENT1)
```

```
#define Segmn0off()   PORTD |= (1<<SEGMENT0)
```



#define SEGMENT0DAT 0

#define SEGMENT1DAT 1

#define SEGMENT2DAT 2

#define SEGMENT3DAT 3

#define SEGMENT4DAT 4

#define PACKETMAKS 8

#define PFINISH 3




```
#define PSTART 1
```

```
#define Led_batH_on() P_Led_bat |= _BV(Led_batH)
```

```
#define Led_batH_off() P_Led_bat &= ~_BV(Led_batH)
```

```
#define Led_batM_on() P_Led_bat |= _BV(Led_batM)
```

```
#define Led_batM_off() P_Led_bat &= ~_BV(Led_batM)
```

