

**PENENTUAN SUMBER PANAS
DENGAN METODE TOMOGRAFI
MENGUNAKAN SENSOR TERMOMETER DIGITAL DS18B20**

skripsi

oleh

RATIH PRATIWI

0305020764



**DEPARTEMEN FISIKA
fakultas matematika dan ilmu pengetahuan alam
UNIVERSITAS INDONESIA
2009**

**PENENTUAN SUMBER PANAS
DENGAN METODE TOMOGRAFI
MENGUNAKAN SENSOR TERMOMETER DIGITAL DS18B20**

skripsi

Diajukan Untuk Melengkapi Persyaratan Memperoleh Gelar Sarjana

oleh

RATIH PRATIWI

0305020764



**DEPARTEMEN FISIKA
fakultas matematika dan ilmu pengetahuan alam
UNIVERSITAS INDONESIA
2009**

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Ratih Pratiwi

NPM : 0305020764

Tanda Tangan :

Tanggal : 17 November 2009

HALAMAN PENGESAHAN

Nama Mahasiswa : Ratih Pratiwi
NPM : 0305020764
Departemen : Fisika
Peminatan : Instrumentasi
Tanggal Sidang : 30 November 2009
Judul Skripsi : Penentuan Sumber Panas Dengan Metode
Tomografi Menggunakan Sensor Termometer
Digital DS18B20

Skripsi ini telah diperiksa dan disetujui oleh

Dr. Eng. Supriyanto, M.Sc

Pembimbing

Dr. Santoso Soekirno

Penguji I

Dr. B.E.F Da Silva, M.Sc

Penguji II

Dr. Santoso Soekirno

Ketua Departemen Fisika

KATA PENGANTAR

Puji syukur hanya kepada Allah SWT yang selalu memberikan anugrah terindah-Nya kepada penulis dan telah menuntun penulis dalam menyelesaikan skripsi ini dengan baik.

Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Sains Jurusan Fisika pada Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia.

Penulis menyadari bahwa, selesainya skripsi ini tidak terlepas dari bantuan, bimbingan, dorongan dan doa yang tulus dari banyak pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini. Tanpa itu semua sangatlah sulit bagi penulis untuk menyelesaikan skripsi ini. Untuk itu penulis ingin menyampaikan ucapan terima kasih yang setulus - tulusnya kepada:

1. Bapak, mamah, kakak-kakakku Ridwan, Kokom, Riani, Surya, Rina, Rahmat, adik-adikku Rosi dan Rian, serta keponakan-keponakanku Zulva, Mutia, Nayla, Anissa yang selalu memberikan bekal mental, motivasi, hiburan, tawa, doa, semangat, pengertian, perhatian, dan banyak hal selama berlangsungnya proses pembelajaran, “Luph U Full”.
2. Dr. Eng. Supriyanto, M.Sc selaku dosen pembimbing yang telah membimbing penulis dalam segala hal, baik dalam ilmu pengetahuan, nasehat, dorongan, semangat serta segala fasilitas yang telah diberikan dalam menyelesaikan tugas akhir ini, “Thanks for all, Juragan”.
3. Kak Muchtar yang telah bersedia meluangkan waktu untuk diskusi dan bimbingan sehingga skripsi ini dapat selesai dengan baik, “Semoga cepet dapetin kampret betina kak..cie..cie..!”.
4. Om Karno yang selalu meluangkan waktu untuk diskusi, semangat, dan bimbingan sehingga skripsi ini dapat selesai dengan baik, ”Jangan kebanyakan makan dodol pak..he..he”.
5. Kak Tyo dan Kak Alva yang telah bersedia meluangkan waktu untuk memberikan motivasi dan bantuan

6. Rizki Edogawa Mahmud (06) yang telah menemani dan memberikan banyak sekali bantuan, “Hati-hati penyakit gila dapat menular”.
7. Junior-junior tim H8, Lia, Ami, Momoi, Faldo, Rusyda, dan lain-lain, ”Selamat meneruskan perjuangan berteman dengan ”makhluk2” @ Cisco”.
8. Rekan-rekan CISCO Fisika UI, Bpk Dwi Seno, Bpk Arif, Om Gindo, Kak Azis, yang telah bersedia memfasilitasi tempat penelitian, tempat santai, dan tempat bermalam yang nyaman.
9. Dr. Santoso Sukirno selaku penguji I dan Dr. B.E.F Da Silva, M.Sc selaku penguji II atas saran dan kritiknya baik sebelum maupun sesudah penulis melakukan sidang.
10. Dr. Agus Salam selaku pemimpin sidang atas saran dan motivasinya sebelum dan sesudah penulis melakukan sidang.
11. Mba Ratna, sekretariat yang paling baik sedunia yang telah memberikan segala informasi dan motivasi, “Jangan pindah lagi ke dekanat ya mba..”
12. Bpk. Suparno, Pa Dwi, dan Pak H. Dono yang telah banyak membantu penulis dalam proses izin inap menginap.
13. Seluruh dosen dan karyawan departemen Fisika atas segala ilmu dan bantuan teknis yang penulis peroleh selama menjadi mahasiswa Fisika UI.
14. Sahabat-sahabat terbaikku @ instrument’05 , Idha, Dian, Thea, Fandi, Imam, Dedy, Aha, Eno, Panggih, Indra, Wahyu, dan Taqwa yang telah memberikan banyak motivasi, canda, tawa, kekonyolan, dan banyak hal selama penulis menjadi mahasiswi Fisika UI, “the best lah pokonya, yang belum lulus cepet2 nyusul kita2....hehehe”
15. Sahabat-sahabat fisika 2005, Ninik, mba sri, Surya, Rahmah, Kupple, Ipin, Kitin, Debi, dan smuanya....”Makasih ya buat smuanya”
16. Sahabat jauh penulis, neng Risa, neng Aneng, neng Rika, neng2 yang lain yang selalu memberikan semangat dan setia mengisi keceriaan saat dilanda kebosanan.

17. Sahabat terbaikku Nurmaitun Nurhasanah Maimunah Sugianah yang selalu menjadi tempat penulis mengeluh, diskusi, menangis, marah-marah, gila-gilaan, sedih-sedihan, “makasih buat semuanya ya ma” dan Ainurrahman yang selalu memberikan dukungan, motivasi, semangat, nasihat, dan buannnnnyyyak yang lain “jadi pengen nangis ni gw”.
18. Calon Sahabatku seumur hidup, Moku, Mohamad Kurnadi, Onhetkuw, Mas Adhi, May Haniy Baniy Switiykuw, makasiiiiiiiiiiiiiiiih banyak buat yang udah dikasih dari no 1-16 yang ga cukup kalo diketik, “Inget...kalo dapet nikmat..”Syukurilah” kalo dapet musibah atau masalah “bersabarlah””, semoga.....(amin)”.
19. My Laptop, Lab Cisco, My Home, My Room yang setia menemani penulis selama proses pembuatan skripsi ini.
20. Luna maya, Britney Spears, TaTu, Tata Young, Mulan Jamilah, Gita Gutawa, Andra n the Backbone, dll yang menemani penulis dengan musik-musik yang memberi semangat.
21. Terima kasih untuk semua pribadi yang secara sadar ataupun tidak, telah menjadi ‘guru’ dalam kehidupan penulis.

Semoga skripsi ini dapat berguna bagi siapa saja yang mengkajinya, serta dapat dikembangkan dan disempurnakan agar lebih bermanfaat untuk kepentingan orang banyak.

Depok, 30 November 2009

Penulis

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

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

Nama : Ratih Pratiwi
NPM : 0305020764
Program Studi : S1 Fisika
Departemen : Fisika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis Karya : Skripsi

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

**PENENTUAN SUMBER PANAS
DENGAN METODE TOMOGRAFI
MENGUNAKAN SENSOR TERMOMETER DIGITAL DS18B20**

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

Dibuat di : Depok

Pada Bulan : November 2009

Yang menyatakan

(Ratih Pratiwi)

ABSTRAK

Nama : Ratih Pratiwi
Program Studi : S1 Fisika
Judul : Penentuan Sumber Panas dengan Metode Tomografi
Menggunakann Sensor Termometer Digital DS18B20

Sistem penentuan sumber panas dengan metode tomografi menggunakan 36 sensor termometer digital DS18B20 telah dibuat dengan menggunakan mikrokontroler H8/3069F dengan komunikasi secara *1 wire*. Komunikasi secara *1 wire* ini memberi kemudahan dan penghematan penggunaan pin mikrokontroler sehingga pada penelitian ini hanya 1 pin mikrokontroler saja yang digunakan untuk memperoleh data yang terukur pada sensor sebanyak 36 secara bersamaan. Sistem ini menggunakan suatu ruang tertutup yaitu sebuah kotak inkubator berdimensi 60 x 40 x 40 cm. *Graphical User Interface* (GUI) yang dibuat dengan bahasa python digunakan untuk pengolahan data dan penampilan hasil keluaran sensor untuk kemudian disimpan dalam database MySQL. Data temperatur yang diperoleh pada sensor termometer digital DS18B20 dijadikan syarat batas oleh metode *finite difference* untuk menghasilkan data distribusi temperatur dan diolah menjadi sebuah grafik kontur yang dibuat dalam program wxPython dan Python GUI. Namun dengan pengolahan data menggunakan finite difference, hasilnya kurang baik sehingga dicoba pengolahan data menggunakan interpolasi kriging menggunakan *software* surfer. Grafik kontur yang dihasilkan adalah berupa hubungan antara letak sensor pada sumbu x dan y dengan temperatur yang terukur pada sumbu x dan y tersebut. Letak sumber panas yang dihasilkan adalah berupa sumbu koordinat x dan y yang menunjukkan temperatur yang paling tinggi.

Kata kunci : Tomografi , DS18B20, mikrokontroler H8/3069F, *1 wire*, python, *Graphical User Interface* (GUI), dan *finite difference*

ABSTRACT

Name : Ratih Pratiwi

Program study : Physics

Title of essay : *Heat Source Determination
Based on Tomography Method Using
Digital Thermometer Sensor DS18B20*

Determination system with the heat source tomography method using a digital thermometer sensor 36 has been made DS18B20 using H8/3069F microcontroller with 1 wire communication. 1 wire communication gives convenience and economy, so the use of pin microcontroller in this study only 1 pin microcontroller are used to obtain measurable data on as many as 36 sensors simultaneously. This system uses an enclosed space that is an incubator box dimensions 60 x 40 x 40 cm. Graphical User Interface (GUI) that is made with python language used for data processing and viewing of the sensor output to then be stored in a MySQL database. Temperature data obtained on a digital thermometer sensor DS18B20 made by the boundary condition Finite difference methods to generate data on the distribution of temperature and processed into a contour graph created in the program GUI wxPython and Python. But with data processing using Finite difference, the result is not so good, so try using a data processing kriging interpolation using the software surfer. The resulting graph is the contour of the relationship between the location of sensors on the x and y axis with the temperature measured on the x axis and y are. Location of the source of heat is generated in the form of axis x and y coordinates that show the highest temperatures.

Key Word : Tomography , DS18B20, microcontroller H8/3069F, 1 wire, Python, Graphical User Interface (GUI), and finite difference

DAFTAR ISI

Halaman

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
ABSTRAK	iii
<i>ABSTRACT</i>	iv
DAFTAR ISI	v
DAFTAR TABEL	vii
DAFTAR GAMBAR	viii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Batasan Masalah	2
1.3 Tujuan dan Penelitian	2
1.4 Metodologi Penelitian	2
1.5 Sistematika Penulisan	3
BAB 2 LANDASAN TEORI	5
2.1 Kalor dan Perpindahannya	5
2.1.1 Konduksi	5
2.1.2 Konveksi	6
2.1.3 Radiasi	6
2.2 Sensor Temperatur Digital DS18B20	7
2.2.1 ROM <i>Command</i>	12
2.2.1.1 <i>Search</i> ROM [F0h]	12
2.2.1.2 <i>Read</i> ROM [33h]	13
2.2.1.3 <i>Match</i> ROM [55h]	13
2.2.1.4 <i>Skip</i> ROM [CCh]	13
2.2.1.5 <i>Alarm Search</i> [ECh]	13
2.2.2 DS18B20 <i>Function Commands</i>	14
2.2.2.1 <i>Convert T</i> [44h]	14
2.2.2.2 <i>Write Scratch Pad</i> [4Eh]	14
2.2.2.3 <i>Read Scratch Pad</i> [BEh]	15
2.2.2.4 <i>Copy Scratch Pad</i> [48h]	15
2.2.2.5 <i>Recall E²</i> [B8h]	15
2.2.2.6 <i>Read Power Supply</i> [B4h]	15
2.3 <i>One Wire</i>	19
2.3.1 <i>Reset/Presence Signal</i> (Inisialisasi)	19
2.3.2 <i>Signal Write 0</i> dan <i>Write 1</i>	20
2.3.3 <i>Signal Read 0</i> dan <i>Read 1</i>	20
2.3.4 <i>Read Time Slot</i>	21
2.3.5 <i>Address</i> pada 1- <i>Wire</i>	21
2.4 Mikrokontroler H8/300H	23
2.4.1 <i>Internal</i> ROM	24

2.4.2	Internal RAM.....	24
2.4.3	I/O port: 11 I/O port	24
2.4.4	Internal SCI (Serial Communication Interface) x 3 Channel.....	24
2.4.5	Internal 16-bit Timer x3 Channel dan 8-bit Timer x4 Channel.....	25
2.4.6	Internal TPC (Timing Pattern Controller).....	25
2.4.7	Internal Watch-dog Timer (WDT)	25
2.4.8	Internal A/D Converter dengan Resolusi 10-bit × 8 Channel	25
2.4.9	Internal 8-bit D/A Converter × 2 Channel.....	25
2.4.10	Internal DMA Controller (DMAC) × 4 Channel	25
2.5	Perangkat Lunak Python.....	26
2.5.1	Database MySQL.....	27
BAB 3	PERANCANGAN ALAT.....	17
3.1	Perangkat Keras	29
3.1.1	Pengoprasian Sensor DS18B20.....	30
3.1.2	Proses Kalibrasi Sensor DS18B20	33
3.2	Perangkat Lunak	34
3.2.1	Tampilan Data pada GUI dan Database	34
3.2.2	Pengolahan Data Temperatur dengan Algoritma <i>Finite Difference</i>	36
BAB 4	ANALISA HASIL PENELITIAN	49
4.1	Perangkat Keras.....	49
4.1.1	Hasil dan Analisis Data Rom Code DS18B20	49
4.1.2	Hasil dan Analisis Data Temperatur	50
4.1.3	Hasil dan Analisis Data Kalibrasi Sensor DS18B20.....	51
4.1.4	Hasil dan Analisis Data dan Grafik Distribusi Temperatur.....	57
BAB 5	KESIMPULAN DAN SARAN	61
5.1	Kesimpulan	61
5.2	Saran	61
DAFTAR ACUAN	62
LAMPIRAN	64

Daftar Tabel

Tabel 2. 1. Deskripsi Pin DS18B20	8
Tabel 2.2. Konfigurasi nilai R0 dan R1 terhadap nilai resolusi	11
Tabel 2.3. Tabel Perintah Fungsi DS18B20	16
Tabel 2.4. Contoh 1 Operasi DS18B20.....	21
Tabel 2.5. Contoh 2 Operasi DS18B20.....	22
Tabel 3.1 Posisi DIP Switch Mode-3	32
Tabel 3.2 Posisi DIP Switch Mode-7	33
Tabel 4.1 Data temperatur sensor DS18B20(Jarak antara sensor dan sumber panas semakin didekatkan).....	52
Tabel 4.2 Data temperatur sensor termometer raksa(Jarak antara sensor dan sumber panas semakin didekatkan).....	53
Tabel 4.3 Data temperatur sensor DS18B20 dibandingkan dengan termometer raksa	53
Tabel 4.4 Data temperatur sensor DS18B20(Jarak antara sensor dan sumber panas semakin dijauhkan)	55
Tabel 4.5 Data temperatur sensor termometer raksa(Jarak antara sensor dan sumber panas semakin dijauhkan)	56

Daftar Gambar

Gambar 1.1	Skematik perancangan alat.....	2
Gambar 1.2	Skematik langkah-langkah penelitian	3
Gambar 2. 1	Bentuk fisik DS18B20	8
Gambar 2.2	Blok diagram DS18B20	8
Gambar 2.3.	Register temperatur	9
Gambar 2.4.	Pensupplaiian dengan <i>mode</i> parasit power.....	10
Gambar 2.5.	Pensupplaiian dengan <i>mode</i> konvensional.....	10
Gambar 2.6.	64-bit ROM Code.....	11
Gambar 2.7.	Peta Memori Scratchpad DS18B20.....	11
Gambar 2.8.	Register Konfigurasi	11
Gambar 2.9.	Generator CRC	12
Gambar 2.10.	<i>Flowchart</i> ROM Command DS18B20.....	19
Gambar 2.11.	<i>Flowchart</i> Perintah Fungsi DS18B20	20
Gambar 2.12.	Hardware 1 Wire	20
Gambar 2.13.	Reset/Presence signal	20
Gambar 2.14.	Signal <i>Write</i> 0 dan <i>Write</i> 1	20
Gambar 2.15.	Signal <i>Read</i> 0 dan <i>Read</i> 1	20
Gambar 2.16	Minimum sistem mikrokontroler H8/3069F.....	23
Gambar 3.1	Blok Diagram Sistem	29
Gambar 3.2	Ilustrasi peletakan sensor pada tepi permukaan atas kotak incubator	30
Gambar 3.3	Rangkaian Operasi Sensor DS18B20 ke Mikrokontroler H8/3069F.....	32
Gambar3.4	Rangkaian Operasi Seluruh Sensor ke Mikrokontroler H8/3069F	33

Gambar 3.5	Langkah-langkah saat melakukan writing program pengoperasian DS18B20 modul mikrokontroler H8/3069F	34
Gambar 3.6	Langkah-langkah saat melakukan writing program pembacaan ROM Code DS18B20 ke modul mikrokontroler H8/3069F	35
Gambar 3.7	Ilustrasi proses pengkalibrasian DS18B20 dengan termometer raksa.....	34
Gambar 3.8	Draft Tampilan Utama	35
Gambar 3.9	Tampilan GUI saat pengambilan data temperatur sensor DS18B20	35
Gambar 3.10	Susunan <i>syarat batas</i> untuk mendapatkan distribusi temperatur pada sisi permukaan incubator.....	37
Gambar 3.11	Skema <i>grid lines</i> dan <i>mesh points</i> pada penentuan distribusi temperatur.....	37
Gambar 4.1	(a),(b),(c),(d),(e) Data rom code pada hiperterminal.....	50
Gambar 4.2	Cara mengkonversi dan data yang tampil di hiperterminal.....	51
Gambar 4.3	Grafik hubungan antara jarak dan temperatur yang terukur pada DS18B20	54
Gambar 4.4	Grafik hubungan antara jarak data temperatur yang terukur pada termometer raksa.....	54
Gambar 4.6	Gambar grafik berdasarkan semua data-data yang diperoleh pada DS18B20 dan termometer raksa.....	54
Gambar 4.7	Data Temperatur 35 Sensor.....	59
Gambar 4.8	Tampilan GUI python saat menampilkan nilai temperatur tertinggi, temperatur terendah, rata-rata, dan menunjukan letak sumber panas	60

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi tomografi sedang menjadi topik yang begitu menarik di berbagai bidang. Terdapat beberapa macam teknik tomografi, yaitu tomografi terkomputasi, tomografi optik, tomografi kuantum, tomografi seismik, tomografi X-Ray, tomografi transmisi ultrasonik, tomografi akustik, tomografi termal, dan lain-lain. Beberapa teknologi yang menggunakan metode tomografi yaitu *electrical capacitance volume tomography (ECVT)* yang dikembangkan di dunia industri dan kedokteran, tomografi akustik yang digunakan sebagai pelacak penyusup dalam laut di bidang pertahanan, sistem tomografi komputer translasi rotasi yang digunakan sebagai pendeteksi produk palsu di dunia perindustrian.[1][2][3][4]

Selain aplikasi-aplikasi tersebut terdapat salah satu aplikasi tomografi termal yang menarik untuk diteliti, yaitu suatu perancangan alat untuk mengetahui distribusi temperatur pada suatu bidang. Tomografi termal adalah teknik untuk menghasilkan citra temperatur suatu obyek dengan memanfaatkan objek pembatas yang diletakkan di beberapa tempat yang berbeda. [5] Dalam terminologi fisika, citra dapat didefinisikan sebagai representasi distribusi suatu besaran fisis (temperatur). Sistem tomografi sendiri terdiri atas sensor, pengolah sinyal, dan komputer untuk mengendalikan dan merekonstruksi citra. [2]

Hal ini membuat peneliti tertarik untuk menerapkan metode tomografi termal pada rancangan sistem pendeteksi sumber panas dalam ruang tertutup inkubator. Dengan metode ini peneliti tidak perlu meletakkan sensor di titik yang ingin peneliti ketahui temperturnya. Peneliti hanya meletakkan sensor di tepi saja yang dapat ditentukan sebagai syarat batas. Rancangan sistem yang peneliti buat ini memanfaatkan mikrokontroler 16 bit H8/3069F sebagai pengendali utama akuisisi data.

1.2 Batasan Masalah

Pada penelitian ini peneliti menggunakan DS18B20 sebagai sensor temperatur digital, *1 wire* sebagai komunikasi data, pengolah data dengan metode numerik, tampilan GUI menggunakan bahasa pemrograman Python, dan *database MySQL* sebagai penyimpan data.

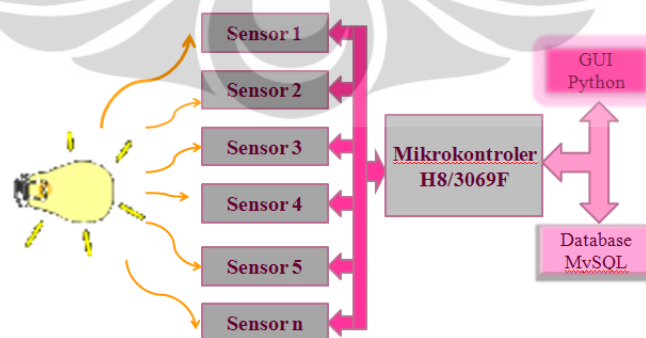
1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk menentukan sumber panas menggunakan metode tomografi

1.4 Metodologi Penelitian

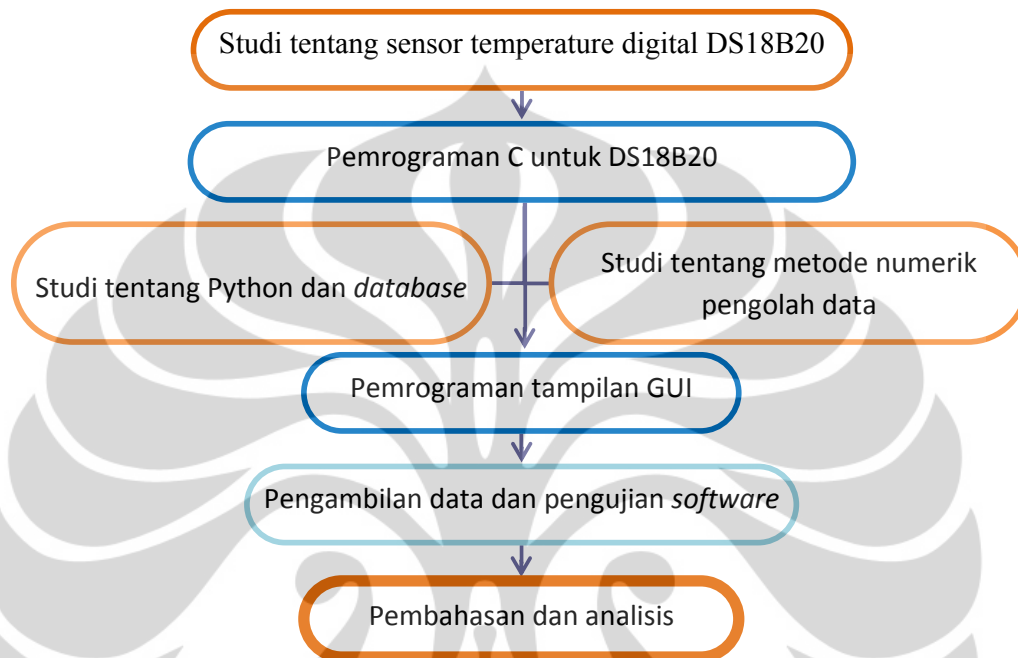
Dalam perancangan sistem ini peneliti menggunakan sejumlah sensor temperatur digital DS18B20 yang diletakan disisi permukaan inkubator. Keluaran sensor adalah berupa sinyal digital yang langsung menunjukkan nilai temperatur yang terukur. Sensor DS18B20 ini terhubung ke mikrokontroler menggunakan komunikasi 1-wire. Karena sistem ini memiliki sensor yang jumlahnya banyak, maka data distribusi temperatur pada sisi permukaan inkubator dapat terukur secara langsung. Data tersebut kemudian akan diolah menggunakan algoritma *finite difference* sehingga diperoleh letak sumber panasnya.

Berikut ini merupakan skematik perancangan alat:



Gambar 1.1 Skematik Perancangan Alat

Berikut ini merupakan skematik langkah-langkah penelitian:



Gambar 1.2 Skematik langkah-langkah penelitian

1.5 Sistematika Penulisan

Sistematika penulisan skripsi terdiri atas lima bab yang secara garis besar dapat diuraikan sebagai berikut:

1. Bab 1 Pendahuluan
Bab ini memuat tentang latar belakang dari penelitian, tujuan, metode yang digunakan, dan juga pembatasan masalah pada penelitian yang dilakukan.
2. Bab 2 Landasan Teori
Bab ini memuat secara garis besar teori dasar yang berhubungan dengan penelitian.
3. Bab 3 Perancangan Alat
Bab ini memuat perancangan perangkat keras dan perangkat lunak
4. Bab 4 Analisis Hasil Penelitian

Bab ini berisi penjelasan mengenai hasil penelitian dan analisis dari sistem yang telah dibuat.

5. Bab 5 Penutup

Bab ini berisi kesimpulan atas hasil analisis dan saran yang mendukung penelitian agar memberikan hasil yang lebih baik lagi untuk pengembangannya.



BAB 2

LANDASAN TEORI

2.1 Kalor dan Perpindahannya

Terdapat suatu perbedaan antara kalor (heat) dan energi dalam dari suatu bahan. Kalor hanya digunakan bila menjelaskan perpindahan energi dari satu tempat ke yang lain. Kalor adalah energi yang dipindahkan akibat adanya perbedaan temperatur. Sedangkan energi dalam (termis) adalah energi karena temperaturnya. Bila dua benda atau lebih terjadi kontak termal maka akan terjadi aliran kalor dari benda yang bertemperatur lebih tinggi ke benda yang bertemperatur lebih rendah, hingga tercapainya kesetimbangan termal. Proses perpindahan kalor ini berlangsung dalam 3 mekanisme, yaitu : konduksi, konveksi dan radiasi.[6]

2.1.1 Konduksi

Proses transfer energi yang paling jelas terkait dengan perbedaan suhu adalah konduksi termal. Dalam proses ini, transfer dapat diwakili pada skala atom sebagai pertukaran energi kinetik antara mikroskopis partikel - molekul, atom, dan elektron. Partikel yang energinya rendah dapat meningkat dengan menumbuk partikel dengan energi yang lebih tinggi.[6]

Sebagai contoh, jika kita memegang salah satu ujung logam yang panjang dan memasukkan ujung yang lain ke dalam api, kita akan merasakan temperatur dari logam di tangan kita segera meningkat. Energi mencapai tangan kita dengan cara konduksi. Sebelum dipanaskan atom dan elektron dari logam bergetar pada posisi setimbang. Pada ujung logam mulai dipanaskan, pada bagian ini atom dan elektron bergetar dengan amplitudo yang makin membesar. Selanjutnya bertumbukan dengan atom dan elektron disekitarnya dan memindahkan sebagian energinya. Kejadian ini berlanjut hingga pada atom dan elektron di ujung logam yang satunya. Konduksi terjadi melalui getaran dan gerakan elektron bebas.[6]

2.1.2 Konveksi

Perpindahan kalor secara konveksi adalah perpindahan kalor karena berpindahnya partikel-partikel atau materi zat itu sendiri. Misalnya jika materi zat tersebut zat cair atau gas yang berpindah adalah zat cair atau gas itu sendiri. Tidak seperti perpindahan kalor pada konduksi, dimana materi zat itu tidak berpindah. Besarnya kalor yang dialirkan tergantung pada luas permukaan (A), jenis fluida (h), dan perbedaan suhu (ΔT). [6]

Pada perpindahan kalor secara konveksi ini memiliki 2 jenis aliran kalor, yaitu aliran yang terjadi secara alami dan aliran yang dipaksakan. Contoh perpindahan konveksi yang mengalir secara alami adalah aliran udara di pantai, pencampuran yang terjadi saat memasak air dan pada permukaan danau. Sedangkan contoh perpindahan kalor secara konveksi yang dipaksakan adalah pada pendingin kendaraan bermotor, pengering rambut, atau pada teko listrik untuk merebus air.[6]

2.1.3 Radiasi

Radiasi adalah perpindahan energi kalor dalam bentuk gelombang elektromagnetik dimana cara perambatannya tidak memerlukan medium seperti cara perambatan secara konduksi dan konveksi. Semua benda terus-menerus memancarkan energi dalam bentuk gelombang elektromagnetik yang dihasilkan oleh getaran termal dari molekul. Kita akan memahami radiasi elektromagnetik dalam bentuk cahaya jingga dari kompor listrik, pemanas ruangan listrik, atau gulungan pemanggang roti. [6]

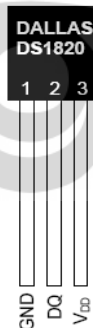
Sebuah benda dapat memancarkan (meradiasikan) energi atau kalor dalam bentuk gelombang-gelombang elektromagnetik dengan berbagai panjang gelombang atau frekuensi walaupun benda tersebut memancarkan cahaya atau tidak. Bila pancaran tersebut mengenai permukaan benda maka ada sebagian energi yang dipantulkan namun ada pula yang diserap bahkan ada sebagian pula terdapat energi pancaran yang diteruskan melalui benda. Permukaan benda yang hitam sempurna merupakan permukaan yang sifat memancarkan atau menyerap radiasi sangat sempurna.[6]

2.2 Sensor Temperatur Digital DS18B20

Banyak sensor suhu yang dipakai dalam implementasi sistem instrumentasi, salah satu contohnya adalah DS18B20. Sensor suhu DS18B20 ini telah memiliki keluaran digital meskipun bentuknya kecil (TO-92), cara untuk mengaksesnya adalah dengan metode serial *1 wire*. Sensor ini sangat menghemat pin *port* mikrokontroler, karena 1 pin *port* mikrokontroler dapat digunakan untuk berkomunikasi dengan beberapa divais lainnya. Sensor ini juga memiliki tingkat akurasi cukup tinggi, yaitu $0,5^{\circ}\text{C}$ pada rentang suhu -10°C hingga $+85^{\circ}\text{C}$, sehingga banyak dipakai untuk aplikasi sistem pemantauan suhu. Aplikasi-aplikasi yang berhubungan dengan sensor seringkali membutuhkan ADC dan beberapa pin port mikrokontroler namun pada DS18B20 ini tidak dibutuhkan ADC agar dapat berkomunikasi dengan mikrokontroler. Bentuk fisik DS18B20 ditunjukkan oleh Gambar 2.1, sementara deskripsi pin dan blok diagram DS18B20 ditunjukkan oleh Tabel 2.1 Gambar 2.2. [7]

Spesifikasi lain dari DS18B20 adalah sebagai berikut:

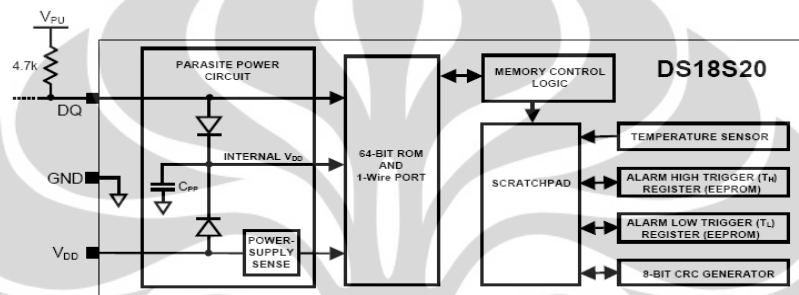
- Memiliki kode serial 64-bit yang unik
- Dapat beroperasi tanpa *Power supply* dari luar
- Power supply* 3-5,5 V. Dapat diperoleh dari aliran data.
- Pengukuran temperatur dari -55°C - $+125^{\circ}\text{C}$
- Resolusi ADC: 9-bit.
- Waktu konversi: maks. 750 ms.



Gambar 2.1. Bentuk Fisik DS18B20

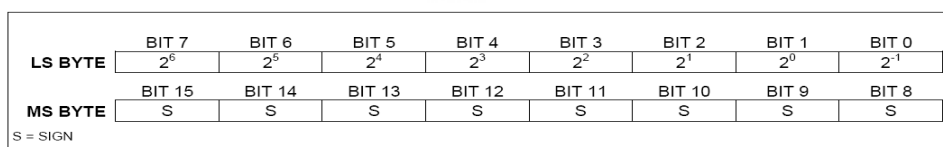
Tabel 2. 1. Deskripsi Pin DS18B20

Pin	Name	Fungsi
1	GND	Ground
2	DQ	Data input/output
3	Vdd	Vdd (cadangan). Saat menggunakan <i>mode parasit power</i> Vdd harus dihubungkan terhadap Ground



Gambar 2.2 Blok Diagram DS18B20

Pada saat beroperasi maka akan terjadi proses pengkonversian temperatur dan konversi ADC pada perintah 44h. Dimana data temperatur yang terukur akan disimpan di memori scratchpad. Dengan *mode power* dari luar maka setelah perintah pengkonversian temperatur DS18B20 akan merespon dengan mengirim bit 0 saat pengkonversian masih dalam proses dan mengirim bit 1 saat pengkonversian telah selesai lalu data temperatur akan disimpan dalam register temperatur 16 bit seperti yang ditunjukkan oleh Gambar 2.3. Pada MS BYTE S menunjukkan tanda bila S diisi oleh bit 0 maka berarti temperatur yang terukur adalah temperatur positif. Jika bit 1 maka temperatur yang terukur adalah temperatur negatif.[7]

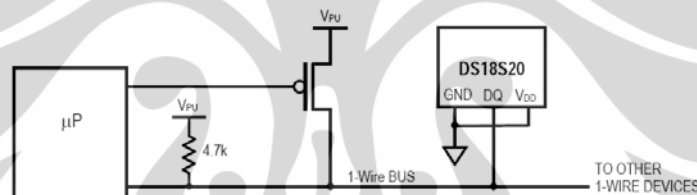


Gambar 2.3. Register Temperatur

Pensuplaian pada DS18B20 terdapat 2 jenis *mode* yaitu pensuplaian dari luar dan *mode* pensuplaian secara parasit(*parasite power*). Pada *mode* pensuplaian

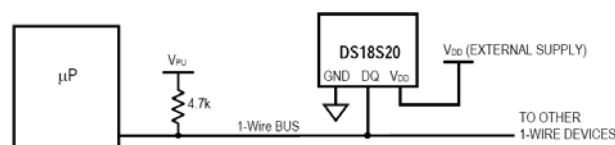
dari luar maka suplai harus dihubungkan pada pin Vdd sedangkan jika menggunakan *mode parasit power* DS18B20 tidak memerlukan suplai dari luar. Pada *mode parasit power* hanya “mencuri” daya dari jalur *1 wire* melalui pin DQ saat jalur dalam keadaan *high*. Sebagian *power* akan disimpan di Cpp untuk memberikan *power* saat jalur dalam keadaan *low*. Saat menggunakan *mode parasit power* maka Vdd harus disambungkan dengan pin *ground*. Dalam *mode parasit power*, jalur *1 wire* dan Cpp akan memberikan arus yang cukup untuk waktu operasi yang lama. [7]

Dengan menggunakan *mode parasit power* saat DS18B20 dalam proses pengkonversian temperatur atau menyalin data dari memori scratchpad ke EEPROM, arus yang beroperasi mencapai 1,5mA. Untuk memastikan bahwa DS18B20 mendapatkan arus yang cukup, maka diperlukan *pullup* yang kuat pada jalur *1 wire*nya. Hal ini dapat dicapai dengan menggunakan MOSFET untuk menarik jalur secara langsung seperti yang ditunjukkan pada Gambar 2.4.[7]



Gambar 2.4. Pensuplaian dengan *mode parasit power*

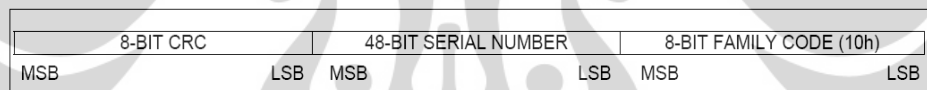
DS18B20 dapat juga diberikan *power* dengan metode konvensional dengan menghubungkan *power supply* luar dengan pin Vdd seperti yang ditunjukkan oleh Gambar 2.5. Dengan menggunakan *mode parasite power* tidak direkomendasikan untuk pengukuran temperatur di atas 100⁰ C karena DS18B20 tidak mampu menahan komunikasi yang disebabkan kebocoran arus yang tinggi. Untuk aplikasi pada temperatur tinggi tersebut sangat disarankan untuk menggunakan *power supply* dari luar.[7]



Gambar 2.5. Pensuplaian dengan *mode konvensional*

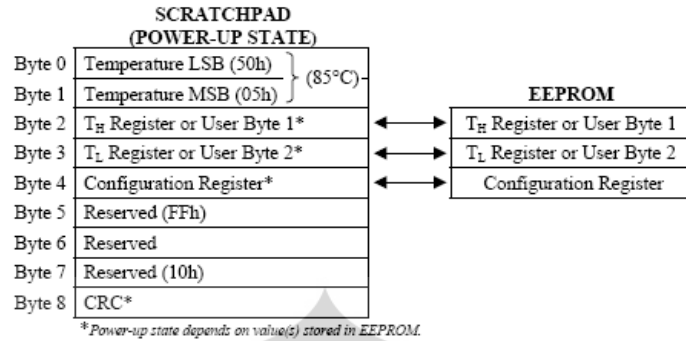
Dalam situasi yang sama jalur *master* mungkin tidak mengetahui apakah DS18B20 menggunakan *mode parasite power* atau *mode power supply* dari luar. *Master* membutuhkan informasi ini untuk menentukan apakah *pullup* jalur yang kuat dibutuhkan atau tidak selama proses pengkonversian temperatur. Untuk mendapatkan informasi ini, *master* memberi perintah untuk melakukan *skip ROM*(CCh) diikuti dengan perintah *Read Power Supply*(B4h) lalu diikuti dengan “*Read time slot*”. Selama pembacaan *time slot*, dengan menggunakan *mode parasite power* DS18B20 akan menarik jalur yang *low* dan dengan menggunakan *mode power* dari luar DS18B20 akan melepaskan jalur yang *high*. Jika jalur yang ditarik *low*, maka *master* akan mengetahui bahwa harus diberi *pullup* yang kuat pada jalur *1 wire* selama proses pengkonversian temperatur. [7]

Setiap DS18B20 memiliki 64 bit kode yang tersimpan di ROM dimana 8 bit pertama(LSB) merupakan kode family DS18B20 seperti yang terlihat pada Gambar 2.6. Pembacaan kode *family* ini ada pada perintah 28h, 48 bit selanjutnya merupakan nomor serial dan 8 bit terakhir(MSB) adalah *byte Cyclic Redudancy Check* (CRC) yang dihitung dari 56 bit pertama pada ROM *code* ini.[7]



Gambar 2.6. 64-bit ROM *code*

Memori pada DS18B20 terdiri dari sebuah SRAM scratchpad dengan register penyimpanan EEPROM (register TH dan TL) dan register konfigurasi. Dimana *Byte 0* dan *byte 1* (LSB dan MSB) pada scratchpad ini diisi oleh register temperatur, *byte 2* dan *byte 3* diisi oleh register TH dan TL, *byte 4* diisi oleh register konfigurasi yang ditunjukkan oleh Gambar 2.8, dimana *byte 5,6*, dan *7* merupakan memori internal yang sudah terisi. Memori scratchpad ini dapat ditunjukkan oleh Gambar 2.7.[7]



Gambar 2.7. Peta Memori Scratchpad DS18B20

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0	R1	R0	1	1	1	1	1

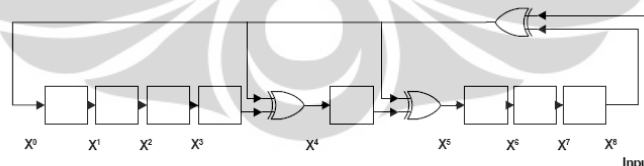
Gambar 2.8. Register Konfigurasi

Pada register konfigurasi terdapat user R0 dan R1 yang mengisi bit 5 dan bit 6. User R0 dan R1 ini dapat mengatur resolusi pengkonversian. Dimana konfigurasi nilai resolusi pengkonversian ini ditunjukkan oleh Tabel 2.2. [7]

Tabel 2.2. Konfigurasi nilai R0 dan R1 terhadap nilai resolusi.

R1	R0	RESOLUTION (BITS)	MAX CONVERSION TIME	
0	0	9	93.75ms	($t_{CONV}/8$)
0	1	10	187.5ms	($t_{CONV}/4$)
1	0	11	375ms	($t_{CONV}/2$)
1	1	12	750ms	(t_{CONV})

Untuk menjamin data integrity maka setiap data yang diterima *master* harus di koreksi terlebih dahulu dimana metode yang dipakai adalah CRC8 yang ditunjukkan oleh Gambar 2.9.[7]



Gambar 2.9. Generator CRC

2.2.1 ROM Command

Semua transaksi pada jalur *1 wire* dimulai dengan langkah inisialisasi. Pada tahap inisialisasi dilakukan reset pulsa yang dikirim oleh *master* dan

presense pulsa yang dikirim oleh *slave*. Setelah *master* mendeteksi pulsa yang muncul ,maka akan berlanjut ke step ROM *command*. Perintah ini akan mengoperasikan kode 64 bit ROM pada setiap *device* dan *master* akan memilih *device* tertentu jika terdapat beberapa *device* dalam 1 jalur. Perintah ini juga memungkinkan *master* untuk menentukan jumlah dan *mode device* yang diperbolehkan dalam jalur *1 wire*. Dimana terdapat 5 ROM *Command* yang masing-masing terdiri dari 8 bit. [7]

Pada ROM *command* ini terdapat beberapa perintah yaitu:

- *Search* ROM
- *Read* ROM
- *Match* ROM
- *Skip* ROM
- *Alarm Search*

Flowchart untuk ROM *Command* DS18B20 ini ditunjukkan oleh Gambar 2.10.

2.2.1.1 Search ROM [F0h]

Saat sistem sedang melakukan penginisialisasian *power*, *master* harus mengidentifikasi kode ROM semua *slave device* yang terdapat pada jalur, sehingga *master* dapat menentukan jumlah *device* dan *mode device* yang terdapat pada jalur tersebut . Jika hanya ada 1 *device* dalam 1 jalur maka perintah *read* ROM dapat digunakan untuk proses *search* ROM.[7]

2.2.1.2 Read ROM [33h]

Perintah ini hanya digunakan pada saat hanya ada 1 *device* dalam 1 jalur sehingga tidak perlu melakukan prosedur *search* ROM. Jika perintah ini digunakan saat ada beberapa *device* dalam 1 jalur maka akan terjadi tabrakan data saat semua *device* membutuhkan respon dalam waktu yang sama.[7]

2.2.1.3 Match ROM [55h]

Perintah *match* ROM diikuti oleh 64 bit kode ROM yang berurutan membuat *master* jalur dapat mengalamatkan *slave device* tertentu. Hanya *slave*

yang persis cocok dengan urutan kode 64 bit ROM yang akan merespon perintah fungsi dari *master* dan *slave* yang lain menunggu sampai pulsanya direset.[7]

2.2.1.4 *Skip ROM [CCh]*

Master dapat menggunakan perintah *skip ROM* untuk mengalamatkan semua *device* pada jalur sekaligus tanpa mengirim beberapa kode informasi yang terdapat pada ROM. Contohnya *master* akan membuat semua DS18B20 melakukan pengkonversian temperatur sekaligus dengan melakukan perintah *skip ROM* diikuti oleh perintah *convert T [44h]*. [7]

2.2.1.5 *Alarm Search [ECh]*

Pengoperasian pada perintah ini sama persis dengan pengoperasian pada perintah *search Rom*. Perintah ini akan memungkinkan *master* untuk menentukan jika ada beberapa DS18B20 yang mengalami kondisi peringatan selama proses pengkonversiaan temperatur. Setelah semua proses pencarian *alarm* selesai maka *master* jalur kembali ke penginisialisasian.[7]

2.2.2 DS18B20 *Function Commands*

Setelah jalur digunakan oleh ROM *Command* untuk mengalamatkan DS18B20 agar dapat berkomunikasi, maka *master* dapat melakukan langkah *funcion command* pada DS18B20. Perintah-perintah ini memungkinkan *master* untuk menulis dan membaca data dari memori *scratchpad*, menginisialisasi pengkonversian temperatur, dan menentukan *mode Power Supply*. Pada perintah fungsi DS18B20 ini terdapat beberapa perintah, yaitu:

- *Convert(44h)*
- *Write Scratchpad(4Eh)*
- *Read Scratchpad(BEh)*
- *Copy Scratchpad(48h)*
- *Recall E²(B8h)*
- *Read Power Supply(B4h)*

Flowchart untuk perintah fungsi DS18B20 ini ditunjukkan oleh Gambar 2.11 dan perintah-perintah yang terdapat pada tahap DS18B20 *function command* ini dapat diringkas seperti yang terlihat pada Tabel 2.3.[7]

2.2.2.1 Convert T [44h]

Perintah ini merupakan penginisialisasian untuk proses pengkonversian temperatur. Hasil data temperatur disimpan dalam 2 *byte* register temperatur dalam memori *scratchpad*. Jika menggunakan *mode parasit power*, sekitar 10 μ s (max) setelah perintah *convert T master* harus dapat menarik jalur dengan kuat dan DS18B20 akan merespon dengan mengirim bit 0 saat pengkonversian temperatur masih dalam proses dan mengirimkan bit 1 saat pengkonversian telah selesai .[7]

2.2.2.2 Write Scratchpad [4Eh]

Perintah ini memungkinkan *master* untuk menulis 3 *byte* data ke memori *scratchpad* dimana *byte 2* yang diisi oleh register TH, *byte 3* oleh register TL, dan *byte 4* oleh register konfigurasi. Data yang dikirim harus bit LSB terlebih dahulu. Ketiga *byte* ini harus tertulis sebelum *master* melakukan *reset* atau data akan rusak.[7]

2.2.2.3 Read Scratchpad [BEh]

Perintah ini memungkinkan *master* untuk membaca konten pada *scratchpad*. Transfer data dimulai dengan LSB pada *byte 0* sampai *byte* ke 9 selesai dibaca. *Master* akan melakukan *reset* untuk menghentikan pembacaan setiap waktu jika hanya bagian pada *scratchpad* dibutuhkan.[7]

2.2.2.4 Copy Scratch Pad [48h]

Perintah ini digunakan untuk *menyalin* konten pada *scartchpad* yaitu register TH,TL, dan konfigurasi pada *byte 2, 3, dan 4* pada EEPROM.[7]

2.2.2.5 Recall E² [B8h]

Perintah ini untuk memanggil kembali nilai TH dan TL serta data konfigurasi dari EEPROM. *Master* melakukan *read time slot* lalu melakukan perintah *recall E²*. DS18B20 akan mengirim bit 0 saat proses pemanggilan dan akan mengirimkan bit 1 jika pemanggilan telah selesai.[7]

2.2.2.6 Read Power Supply [B4h]

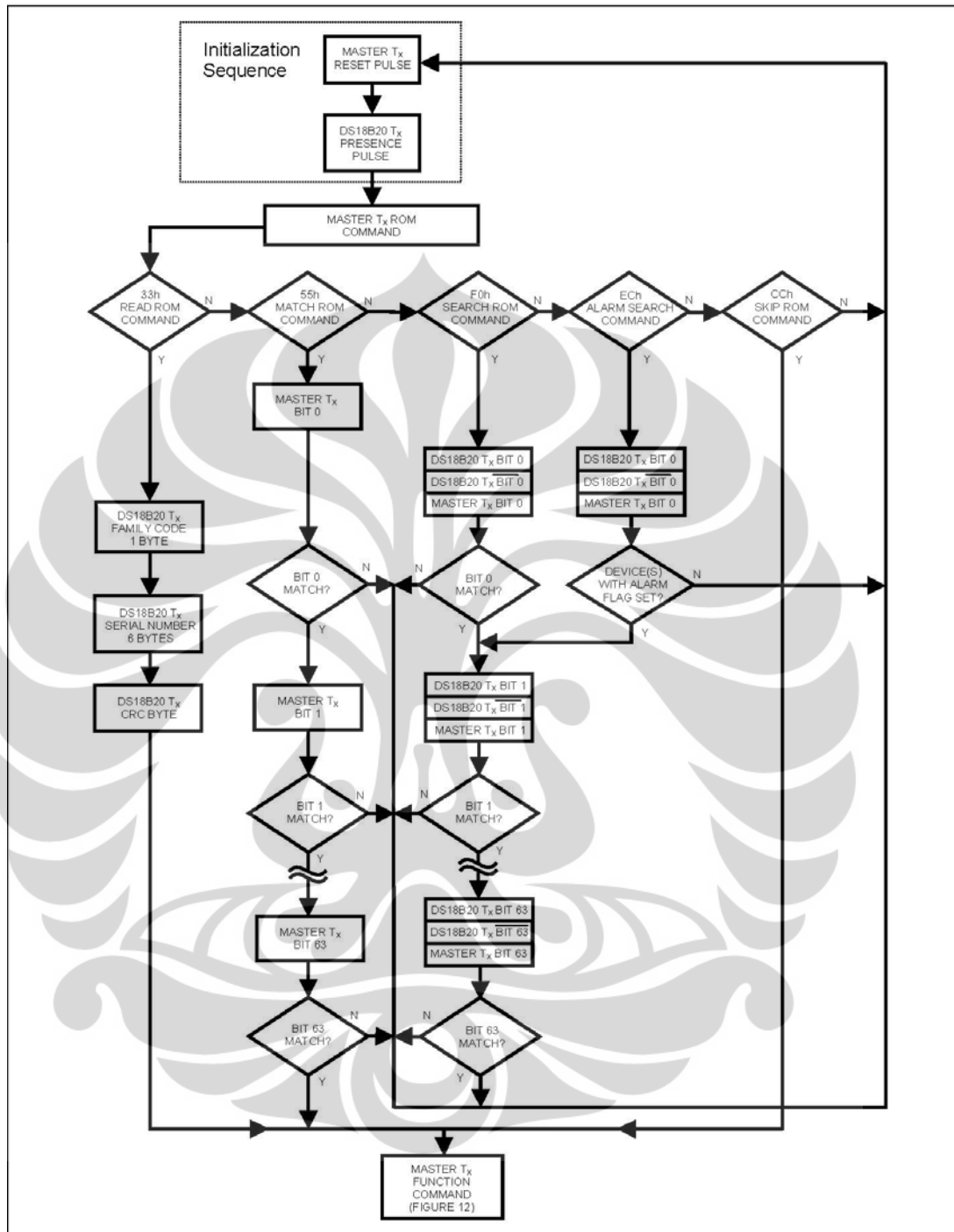
Perintah ini diberikan agar *master* melakukan perintah untuk membaca *time slot* untuk menentukan jika ada DS18B20 pada jalur yang menggunakan *mode parasite power*. Selama pembacaan *time slot*, pada *mode parasite power* jalur dalam keadaan *low* sehingga mesti di tarik sedangkan jika menggunakan *power supply* dari luar jalur sudah dalam keadaan *high*. [7]

Tabel 2.3. Tabel Perintah Fungsi DS18B20

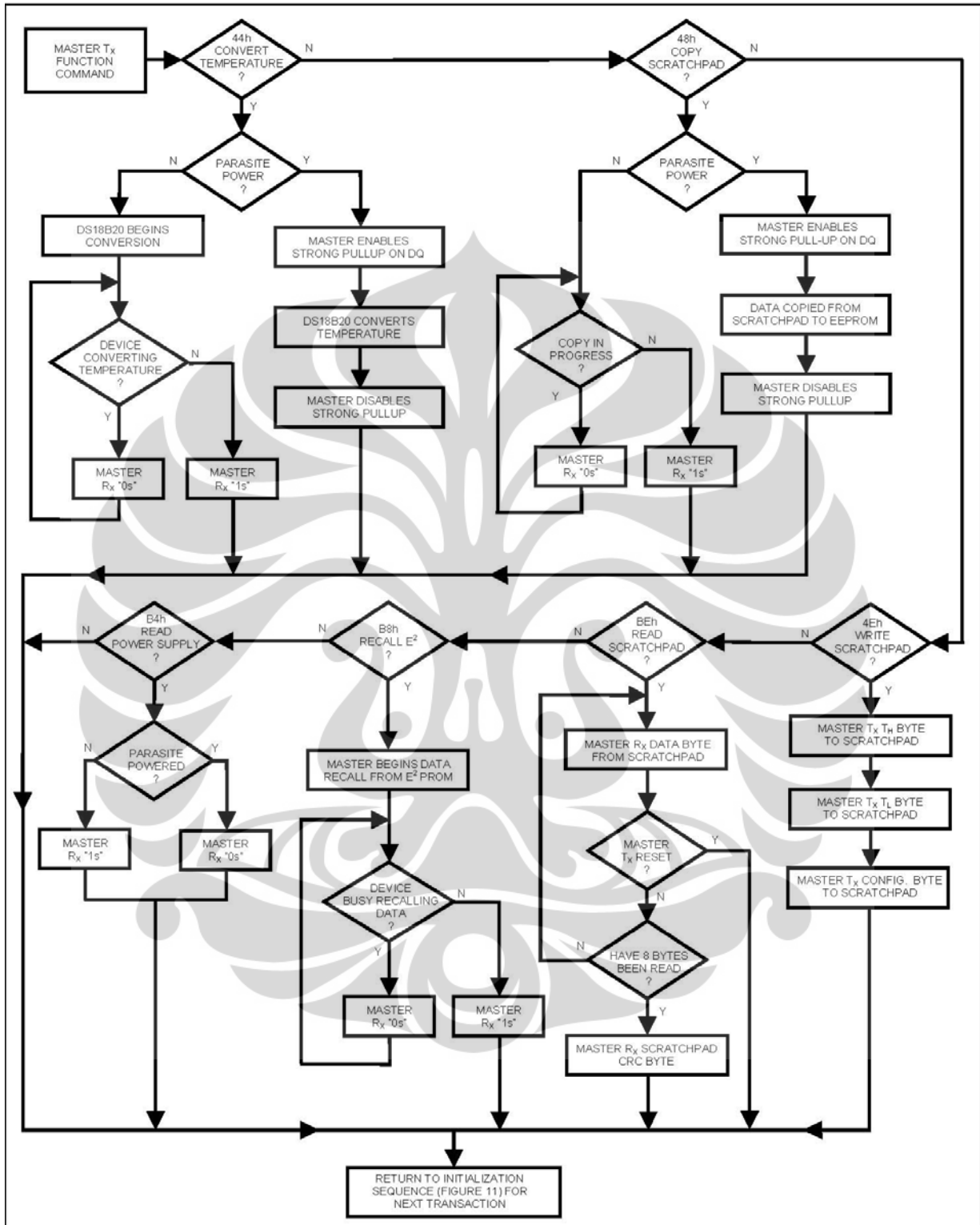
PERINTAH	DESKRIPSI	PROTOKOL	AKTIFITAS JALUR 1 WIRE SETELAH DIBERI PERINTAH
PERINTAH PENGKONVERSIAN TEMPERATUR			
Konversi T	Inisialisasi pengkonversian temperatur	44h	DS18B20 mengirimkan status pengkonversian kepada <i>master</i> (tidak untuk <i>mode parasit power</i>).
PERINTAH MEMORI			
Membaca Scratchpad	Membaca semua yang ada di scratchpad termasuk <i>byte</i> CRC	BEh	DS18B20 mengirimkan data sampai 9 <i>byte</i> kepada <i>master</i>
Menulis Scratchpad	Menulis data kedalam scratchpad pada <i>bytes</i> 2, 3, dan 4 (TH, TL, dan register konfigurasi).	4Eh	<i>Master</i> mengirimkan 3 <i>byte</i> data kepada DS18B20.
<i>Menyalin</i> Scratchpad	<i>Menyalin</i> data TH, TL, dan register	48h	-

	konfigurasi dari scratchpad ke EEPROM.		
Memanggil kembali E ²	Memanggil kembali data TH, TL, dan register konfigurasi dari EEPROM ke scratchpad.	B8h	DS18B20 mengirim status <i>recall</i> pada <i>master</i> .
Membaca <i>Power supply</i>	Membaca <i>mode Power supply</i> .	B4h	DS18B20 mengirim status <i>Power supply</i> pada <i>master</i>





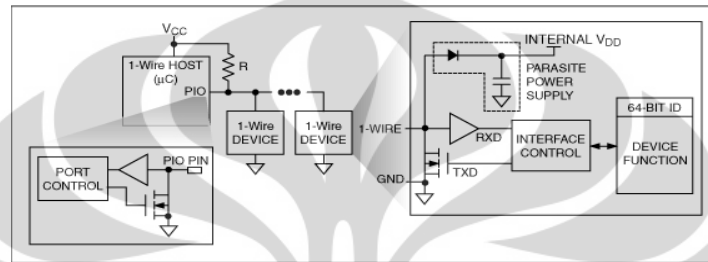
Gambar 2.10. Flowchart ROM command DS18B20



Gambar 2.11. Flowchart perintah fungsi DS18B20

2.3 One Wire

1 wire merupakan buatan dari Dallas Semiconductor. Dirancang untuk transfer data yang rendah dan menggabungkan sinyal data dan *power* dalam satu jalur. Kelebihan *1 wire* terletak pada jalur data yang digunakan yakni hanya satu jalur data (data dan *power*) dan satu jalur *ground*. *Hardware 1 wire* ini dapat ditunjukkan oleh Gambar 2.12.[7]



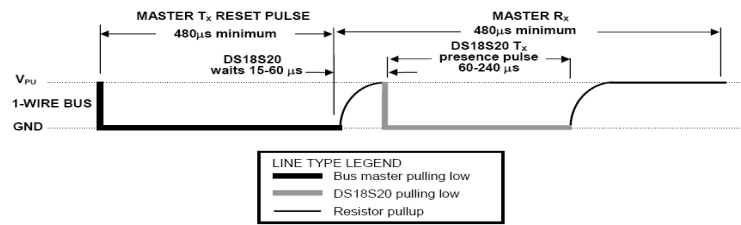
Gambar 2.12. *Hardware 1 wire*

Model komunikasi pada *1 wire* adalah *master-slave*. *Master* adalah mikrokontroler atau mikroprosesor dan *slave* adalah *device 1 wire* (contoh DS1820). Pensinyalan pada jalur *1 wire* dibagi kedalam *slot time* dimana 1 *slot time* besarnya 60 μ s. Pensinyalan dilakukan dengan cara membuat jalur menjadi *low* (defaultnya jalur adalah *high* karena resistor pull up).[7] Jenis-jenis pensinyalan pada *1 wire* adalah sebagai berikut :

- Reset/Presence
- Write 0 dan Write 1
- Read 0 dan Read 1

2.3.1 Reset/Presence signal (Inisialisasi)

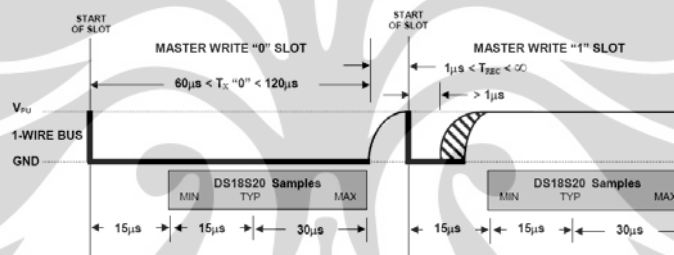
Pada tahap inisialisasi *master* akan mengirim signal reset dengan cara *me-low*-kan selama 8 *time slot*. Setelah itu *master* akan mengubah *mode* receive. Setelah itu *slave* akan mengirim signal presence (*me-low*-kan jalur) selama 1 hingga 4 *time slot* seperti yang ditunjukkan oleh Gambar 2.13.[7]



Gambar 2.13. Reset/Presence signal

2.3.2 Signal Write 0 dan Write 1

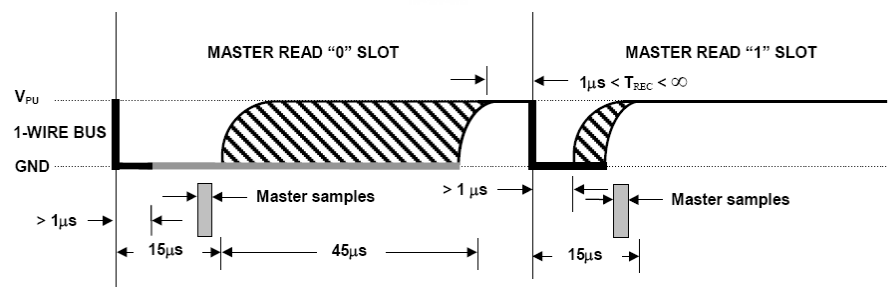
Write 0 dipakai untuk mengirim bit 0 (lebarnya 1-2 *slot time*). *Write 1* dipakai untuk mengirim bit 1 (lebarnya 1 *slot time* dan 1 *slot time* lagi kondisi *high* sebelum mengirim bit berikutnya) seperti yang ditunjukkan oleh Gambar 2.14.[7]



Gambar 2.14. Signal Write 0 dan Write 1

2.3.3 Signal Read 0 dan Read 1

Lebar *time slot* signal *read* adalah 1 *slot time*. *Master* harus me-*low*-kan jalur selama 1 µs sebelum *slave* mengirim data. *Slave* akan mengirim bit 0 dengan mengirim signal *Read 0* dengan membuat jalur tetap *low* selama 1 *time slot*. *Slave* akan mengirim bit 1 dengan mengirim signal *Read 1* dengan membuat jalur *high* selama 1 *time slot* seperti yang ditunjukkan oleh Gambar 2.15.[7]



Gambar 2.15. Signal Read 0 dan Read 1

2.3.4 Read Time Slot

DS18B20 hanya akan mengirim data pada master saat telah membaca *time slot*. *Master* harus segera membaca *time slot* setelah perintah *Read Scratchpad*[Beh] atau *read power supply*[B4h] maka DS18B20 akan memberikan data yang diminta. Satu waktu *master* akan membaca *time slot* setelah perintah *convert T*[44h] atau *Recall E²*[B8h] untuk menemukan status saat sedang beroperasi.[7]

2.3.5 Address Pada 1-Wire

Jalur *1 wire* harus hanya memiliki satu *master* dan lebih dari satu *slave*. Untuk membedakan satu *slave* dengan yang lainnya maka setiap *slave* memiliki *slave address* yang berbeda (meskipun *device slave* nya sama). Panjang alamat dari *slave* adalah 64 bit. Alamat sudah di bakar di dalam ROM *Slave*. Contoh pengoperasian DS18B20 dengan jumlah sensor lebih dari 1 dengan *mode parasite power* dapat ditunjukkan oleh Tabel 2.4.[7]

Tabel 2.4. Contoh 1 operasi DS18B20 dengan *mode parasite power*

MODE MASTER	DATA (LSB FIRST)	COMMENTS
Tx	Reset	<i>Master</i> melakukan peresetan pulsa.
Rx	Presence	DS18B20 merespon dengan mengirimkan pulsa presence
Tx	55h	<i>Master</i> melakukan perintah <i>Match ROM</i> .
Tx	64-bit ROM code	<i>Master</i> mengirim kode ROM DS18B20..
Tx	44h	<i>Master</i> memerintahkan untuk mengkonversi temperatur.
Tx	DQ <i>high</i> karena <i>pullup</i>	<i>Master</i> menggunakan <i>pullup</i> yang kuat pada DQ untuk durasi saat pengkonversian (<i>tCONV</i>).
Tx	Reset	<i>Master</i> melakukan peresetan pulsa.
Rx	Presence	DS18B20 merespon dengan mengirimkan pulsa presence

Tx	55h	<i>Master</i> melakukan perintah <i>Match</i> ROM.
Tx	Kode ROM 64-bit	<i>Master</i> mengirim kode ROM DS18B20.
Tx	BEh	<i>Master</i> membaca perintah Scratchpad.
Rx	9 data bytes	<i>Master</i> membaca seluruh data pada scratchpad termasuk CRC. <i>Master</i> lalu akan menghitung kembali CRC pada 8 data <i>byte</i> pertama dari scratchpad dan membandingkan perhitungan CRC dengan yang terbaca pada CRC(<i>byte</i> 9) Jika cocok akan lanjut, jika tidak pengoperasian harus diulang.

Master jalur menginisialisasi pengkonversian temperatur pada DS18B20 tertentu lalu membaca memori scratchpad dan menghitung ulang CRC untuk memverifikasi data. Contoh pengoperasian DS18B20 dengan jumlah sensor lebih dari 1 dengan *mode parasite power* dapat ditunjukkan oleh Tabel 2.5.[7]

Tabel 2.5. Contoh 2 Operasi DS18B20

MASTER MODE	DATA (LSB FIRST)	COMMENTS
Tx	Reset	<i>Master</i> melakukan peresetan pulsa.
Rx	Presence	DS18B20 merespon dengan mengirimkan pulsa presence
Tx	CCh	<i>Master</i> melakukan perintah <i>skip</i> ROM .
Tx	4Eh	<i>Master</i> melakukan perintah <i>write</i> Scratchpad.
Tx	3 data bytes	<i>Master</i> mengirim 3 <i>byte</i> data pada scratchpad (TH, TL, dan register konfigurasi).
Tx	Reset	<i>Master</i> melakukan peresetan pulsa.
Rx	Presence	DS18B20 merespon dengan mengirimkan pulsa presence
Tx	CCh	<i>Master</i> melakukan perintah <i>skip</i> ROM .
Tx	BEh	<i>Master</i> melakukan perintah <i>read</i> scratchpad.
Rx	9 data bytes	<i>Master</i> membaca seluruh data pada scratchpad termasuk CRC. <i>Master</i> lalu akan menghitung kembali CRC pada 8 data <i>byte</i> pertama dari scratchpad dan membandingkan

		perhitungan CRC dengan yang terbaca pada CRC(byte 9) Jika cocok akan lanjut, jika tidak pengoperasian harus diulang.
Tx	Reset	<i>Master</i> melakukan peresetan pulsa.
Rx	Presence	DS18B20 merespon dengan mengirimkan pulsa presence
Tx	CCh	<i>Master</i> melakukan perintah <i>skip</i> ROM .
Tx	48h	<i>Master</i> melakukan perintah <i>copy</i> Scratchpad.
Tx	DQ <i>high</i> karena <i>pullup</i>	<i>Master</i> menggunakan <i>pullup</i> yang kuat untuk DQ setidaknya selama 10 ms sementara pengkopian sedang berlangsung.

Master menulis TH, TL, dan register konfigurasi di dalam scratchpad lalu membaca scratchpad dan menghitung ulang CRC untuk memverifikasi data. Lalu *master menyalin* data dari scratchpad ke EEPROM.[7]

2.4 Mikrokontroler H8/300H

Mikrokontroler yang digunakan pada penelitian ini adalah seri *H8/300H* yaitu *single-chip microcomputer* berperforma tinggi yang mempunyai sebuah CPU 16-bit (*Control Processing Unit*) sebagai prosessornya. Gambar 2.16 menunjukkan rangkaian minimum sistem mikrokontroler H8/3069F.[8]



Gambar 2.16 Minimum sistem mikrokontroler H8/3069F.

H8/3069F adalah salah satu mikrokontroler yang dilengkapi *processor H8/300H*. *H8/3069F* mempunyai sebuah *internal writable flash memory* yang menggunakan *single Power supply (5V)*. Mikrokontroler ini dapat difungsikan sebagai *single-chip* mikrokomputer karena didalamnya sudah tersedia ROM, RAM dan fungsi I/O. Mikrokontroler ini juga dapat difungsikan sebagai *multi-chip* mikrokomputer saat terjadi penambahan memori. [8]

2.4.1 Internal ROM

Mikrokontroler ini mempunyai flash memory 512K *byte* yang dapat di *Write* dengan sebuah *Power supply (5V)*. [9]

2.4.2 Internal RAM

Mikrokontroler ini mempunyai 16K *byte internal RAM*. [9]

2.4.3 I/O port: 11 I/O port

Tersedia 11 I/O port, yaitu port 1 sampai port 9, port A dan port B. Masing-masing port memiliki fungsi yang sama yaitu sebagai register, *port data register* (PDR) dan *port data direct register* (PDDR). Beberapa port juga bekerja sesuai *mode* yang ada dan beberapa port memiliki kelebihan sebagai input Pull-Up MOS Control Register (PCDR). I/O port dapat digunakan sebagai input status on/off atau sinyal dari berbagai sensor. [9]

2.4.4 Internal SCI (Serial Communication Interface) x 3 Channel

Ketiga *channel* SCI mempunyai fungsi yang sama. *Mode* dari SCI ini adalah sinkron dan asinkron. SCI mikrokontroler ini juga mempunyai komunikasi multiprocessor dengan dua atau lebih processor. SCI juga dapat dihubungkan dengan smart card interface dengan mengubah setting pada register CPU. [9]

2.4.5 Internal 16-bit Timer x 3 Channel dan 8-bit Timer x 4 Channel

channel 0 dan 1 pada 16-bit *timer* mempunyai fungsi yang sama, sedangkan *channel* 2 mempunyai register sendiri pada CPU. 8-bit *timer* dibagi

menjadi dua grup dengan masing-masing dua *channel*. Grup 0 terdiri dari *channel* 0 dan 1 dan grup 1 terdiri dari *channel* 2 dan 3.[9]

2.4.6 Internal TPC (Timing Pattern Controller)

H8/3069F mempunyai TPC yang menyediakan output pulsa dengan berbasis 16-bit *timer*. Pulsa output dari TPC dibagi menjadi grup 4-bit (grup 3 sampai grup 0) yang dapat beroperasi secara serempak dan *independent*. [9]

2.4.7 Internal Watch-dog Timer (WDT)

WDT dapat dioperasikan untuk mengawasi jalannya program, atau hanya sebagai interval *timer*. Ketika WDT digunakan, WDT akan membangkitkan sinyal reset pada chip *H8/3069F* bila sistem *crash*. [9]

2.4.8 Internal A/D converter dengan resolusi 10-bit \times 8 channel

8 *channel* analog input dibagi menjadi dua grup yaitu grup 0 dan grup 1. AV_{CC} dan AV_{SS} adalah *power supply* sirkuit analog pada A/D *converter*, dan V_{REF} adalah tegangan referensi. [9]

2.4.9 Internal 8-bit D/A converter \times 2 channel

Resolusi dari D/A *converter* ini adalah 8-bit. Tegangan output-nya berkisar antara 0V sampai V_{REF} . Pengaturan D/A *converter* ini diatur pada sebuah register di CPU. [9]

2.4.10 Internal DMA controller (DMAC) \times 4 channel

H8/3069F digunakan untuk transfer data berkecepatan tinggi. *DMAC* memungkinkan transfer data lebih cepat dari penggunaan CPU. Umumnya digunakan dengan sebuah *timer* dan fungsi komunikasi lainnya. *H8/3069F* dapat digunakan sebagai single-chip mikrokomputer. Pada kondisi ini, hanya *internal* memori yang dapat digunakan. [9]

2.5 Perangkat Lunak Python

Bahasa pemrograman adalah bahasa yang dapat diterjemahkan menjadi kumpulan perintah-perintah dasar atau bahasa mesin. Penerjemahan dilakukan oleh program komputer yang disebut kompilator atau *compiler*. Setiap bahasa pemrograman mempunyai kompilatornya sendiri. Contohnya, kompilator C++ tidak akan mengerti program yang ditulis dengan bahasa Java. Sintaks dari bahasa pemrograman lebih mudah dipahami oleh manusia daripada bahasa mesin. Namun tentu saja komputer hanya dapat melaksanakan perintah dasar itu. Maka di sinilah peran penting kompilator sebagai perantara antara bahasa pemrograman dengan perintah dasar. Bahasa pemrograman yang digunakan dalam skripsi ini adalah Python.[10]

Python merupakan salah satu bahasa pemrograman tingkat tinggi. Sintaks inti Python dan strukturnya merupakan kode yang mudah dibaca dengan *library* yang luas dan dapat disesuaikan penggunaannya. Python menggunakan pembatasan indentasi dalam penulisan *script*. Python mendukung beberapa paradigma pemrograman terutama berorientasi obyek, prosedur, dan fungsi, juga sistem dinamis dengan fitur yang lengkap dan jenis pengelolaan memori otomatis, mirip dengan Perl, Ruby, Scheme, dan Tcl.[10]

Pengembangan bahasa Python dikelola oleh perusahaan nirlaba Python Software Foundation, yang mempertahankan *de facto* standar definisi bahasa CPython sebagai acuan pengembangannya. Python dibuat pada akhir tahun 1980 oleh Guido van Rossum di CWI Belanda sebagai pengganti bahasa pemrograman ABC yang mampu menangani sistem operasi *interfacing* dengan Amoeba. Bahasa pemrograman ini sendiri terinspirasi oleh SETL.[10]

Python memiliki *library* standar yang besar, yang umum digunakan sebagai salah satu kelebihan dari Python, sehingga dapat dipakai pada banyak keperluan. Modul *library* standar dapat diperluas dengan modul pilihan yang ditulis dalam salah satu bahasa, C atau Python. Baru-baru ini, Boost C++ masuk dalam *library*-nya, yang dinamakan Boost.Python, agar dapat saling menjalankan antar C++ dan Python. Karena berbagai jenis alat-alat dijalankan oleh *library* standart, dimana diperlukan bahasa tingkat rendah seperti C dan C++ yang sudah

mampu menjadi antar muka dengan *library* lainnya, maka Python secara langsung dapat terhubung dengan alat.[10]

Library standar dapat juga digunakan dalam pembuatan aplikasi *desktop* internet, dengan banyak format standar dan protokol, seperti HTTP. Modul-modul Python juga dapat digunakan untuk pembuatan *Graphical User Interface* (GUI), penghubung dengan *database*, perhitungan matematik dengan angka-angka desimal, dan pembuatan persamaan manipulasi.[11]

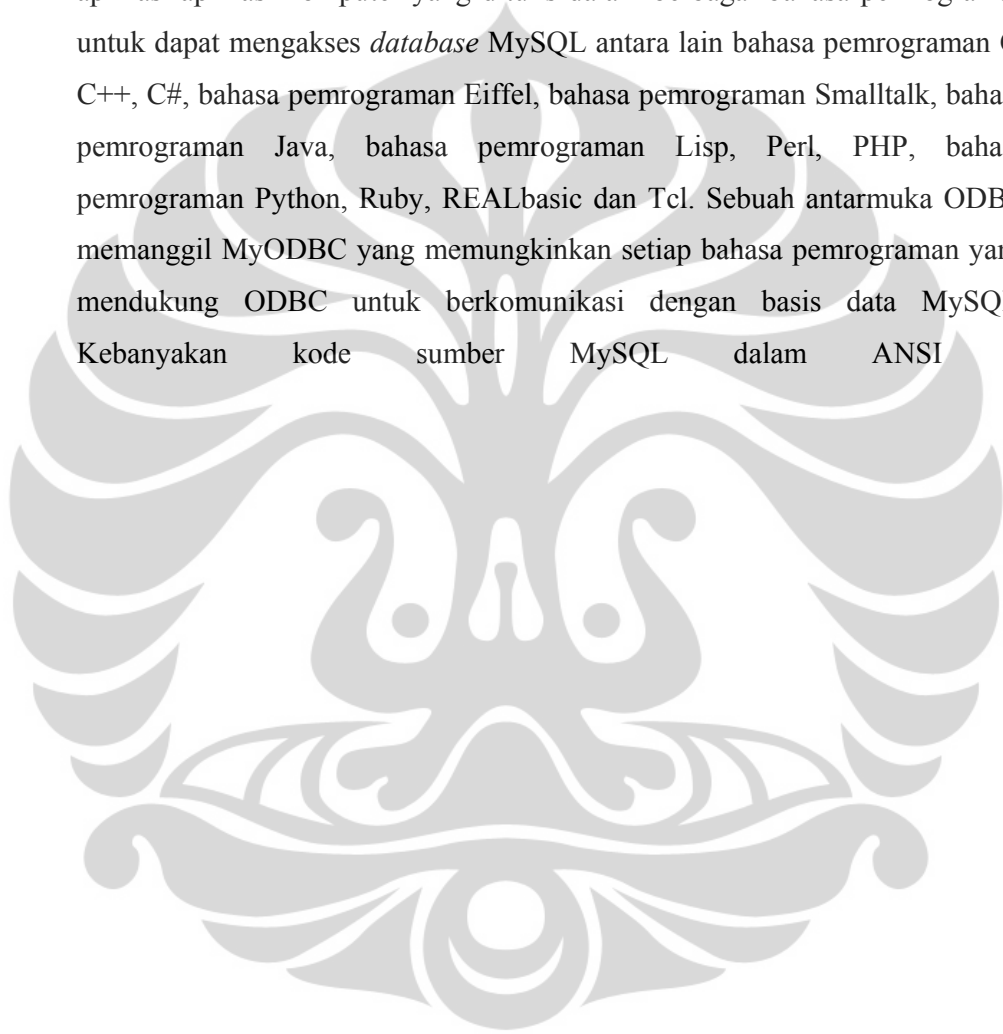
2.5.1 Database MySQL

Pada penelitian ini digunakan basis data MySQL. MySQL adalah sebuah perangkat lunak sistem manajemen basis data *Structured Query Language* (SQL) atau *Database Management Sistem* (DBMS) yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.[12]

Sebagai *database* server, MySQL dapat dikatakan lebih unggul dibandingkan *database* server lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase.[12] Selain itu MySQL juga memiliki beberapa keistimewaan, antara lain:

- *Portability*, yaitu MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
- *Multiuser*, yaitu MySQL dapat digunakan oleh beberapa pengguna atau *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
- *Performance tuning*, yaitu MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memroses lebih banyak SQL per satuan waktu.
- *Security*, yaitu MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta *password* terenkripsi.

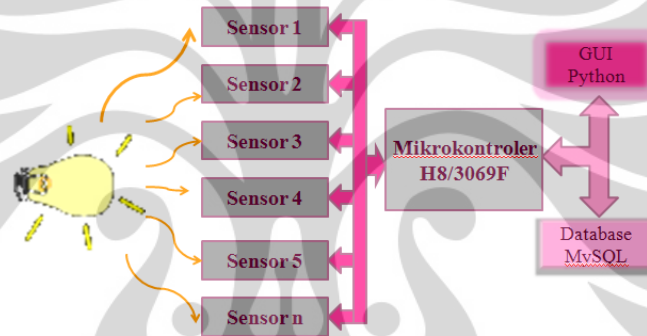
- *Scalability* dan *limits*, yaitu MySQL mampu menangani *database* dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
- Bahasa pemrograman, terdapat beberapa API tersedia yang memungkinkan aplikasi-aplikasi komputer yang ditulis dalam berbagai bahasa pemrograman untuk dapat mengakses *database* MySQL antara lain bahasa pemrograman C, C++, C#, bahasa pemrograman Eiffel, bahasa pemrograman Smalltalk, bahasa pemrograman Java, bahasa pemrograman Lisp, Perl, PHP, bahasa pemrograman Python, Ruby, REALbasic dan Tcl. Sebuah antarmuka ODBC memanggil MyODBC yang memungkinkan setiap bahasa pemrograman yang mendukung ODBC untuk berkomunikasi dengan basis data MySQL. Kebanyakan kode sumber MySQL dalam ANSI C



BAB 3 PERANCANGAN ALAT

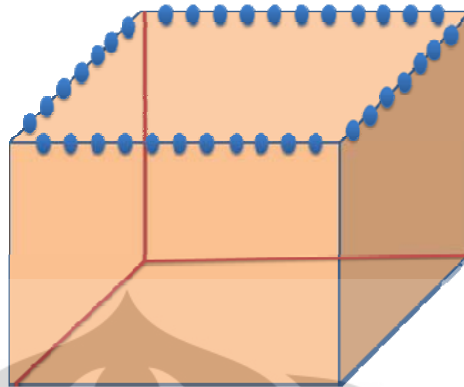
3.1 PERANGKAT KERAS

Dalam perancangan alat penentu sumber panas dengan menggunakan tomografi ini sistem yang digunakan adalah sistem yang sangat sederhana yaitu terdiri atas sensor, pengolah data, dan komputer untuk mengendalikan dan merekonstruksi citra. Gambar 3.1 berikut merupakan blok diagram sistem yang telah dibuat:



Gambar 3.1 Blok Diagram Sistem

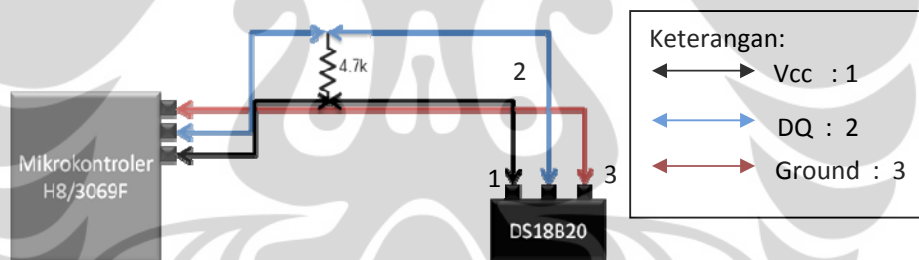
Pada perancangan alat ini peneliti menggunakan kotak inkubator berdimensi 60 x 40 x 60 cm, dimana pada kotak inkubator tersebut diletakan 36 sensor termometer digital DS18B20 di sepanjang tepi permukaan atasnya dengan jarak 5 cm antar satu sensor ke sensor yang lain. Kotak inkubator ini hanya digunakan sebagai media atau sistem agar data distribusi yang diperoleh merupakan nilai distribusi yang hanya berasal dari dalam inkubator tanpa dipengaruhi oleh temperatur dari luar. Ilustrasi peletakan sensor pada tepi permukaan atas kotak inkubator ini ditunjukkan oleh Gambar 3.2.



Gambar 3.2 Ilustrasi peletakan sensor pada tepi permukaan atas kotak inkubator

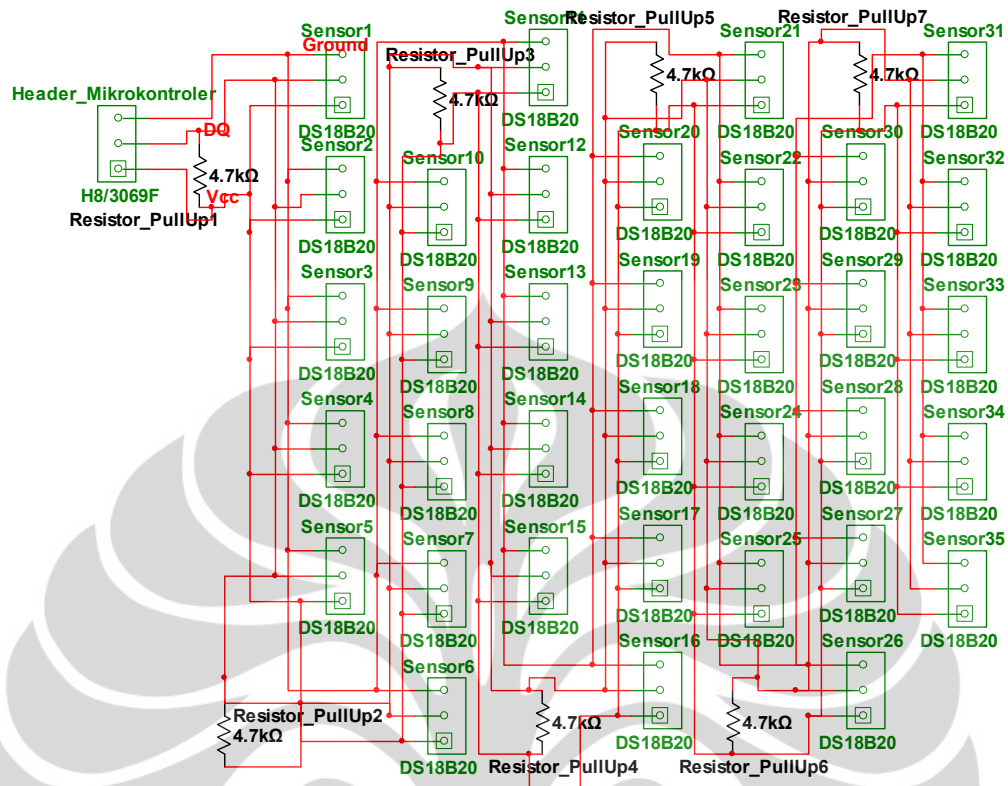
3.1.1. Pengoperasian Sensor DS18B20

Pada penelitian ini pengoperasian DS18B20 menggunakan mode *power supply* dari luar sehingga pin vcc pada DS18B20 dihubungkan ke vcc(mikrokontroler H8/3069F), pin DQ dihubungkan ke pin I/O (Pin 1 pada Port 1 mikrokontroler H8/3069F), dan pin *ground* DS18B20 dihubungkan ke *ground* mikrokontroler H8/3069F seperti yang ditunjukkan oleh Gambar 3.3:



Gambar 3.3 Rangkaian Operasi Sensor DS18B20 ke Mikrokontroler H8/3069F

Pada penelitian ini DS18B20 yang digunakan adalah sebanyak 36 sensor. Seperti yang telah diketahui bahwa DS18B20 ini berkomunikasi secara *1 wire* dalam pengoperasiannya maka hanya 1 pin I/O mikrokontroler H8/3069F saja yang digunakan. Untuk mengoperasikan DS18B20 ini secara bersamaan maka harus dipastikan bahwa setiap DS18B20 memperoleh arus yang cukup untuk beroperasi, oleh sebab itu peneliti menggunakan beberapa resistor pull up agar arusnya tidak turun seperti yang ditunjukkan oleh Gambar 3.4.



Gambar3.4 Rangkaian Operasi Seluruh Sensor ke Mikrokontroler H8/3069F

DS18B20 merupakan sensor yang harus diberikan perintah terlebih dahulu untuk beroperasi, sehingga DS18B20 tidak mungkin beroperasi tanpa mikrokontroler. Untuk mengoperasikan DS18B20 menggunakan mikrokontroler H8/3069F maka peneliti harus mengisikan sebuah program yang memerintahkan DS18B20 ke dalam mikrokontroler H8/3069F tersebut. Mikrokontroler H8/3069F hanya dapat diisi oleh program C. Oleh sebab itu peneliti harus membuat program C untuk mengisi mikrokontroler H8/3069F agar dapat mengoperasikan DS18B20. Gambar 3.5 Berikut ini merupakan langkah-langkah untuk mengisikan sebuah program pada mikrokontroler H8/3069F agar DS18B20 dapat beroperasi.

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\ciTyWePuNyA>E:
E:\>cd tugas akhir/program/tiwi-edit1
E:\TUGAS AKHIR\PROGRAM\tiwi-edit1>ls
Makefile  ds1820.c      interrupt.h  onewire.h   start.S     tiwi.mot   uart.h
delay.c   ds1820.h      ioh83069.h  rom3068f.ld tiwi.c      tiwi.out   vects.c
delay.h   h8write.exe   onewire.c   romcode.h   tiwi.map    uart.c

E:\TUGAS AKHIR\PROGRAM\tiwi-edit1>h8write.exe -3069 -f20 tiwi.mot con9
H8/3069F is ready! 2002/5/20 Yukio Mituiwa.
writing
WARNING:This Line dosen't start with"S".
Address Size seems wrong
WARNING:This Line dosen't start with"S".
Address Size seems wrong
.....
EEPROM Writing is succeeded.
E:\TUGAS AKHIR\PROGRAM\tiwi-edit1>_

```

Gambar 3.5 Langkah-langkah saat melakukan *writing* program pengoperasian DS18B20 modul mikrokontroler H8/3069F

Sebelum melakukan *writing* hasil kompilasi ke modul mikrokontroler H8/3069F, peneliti mengatur posisi *DIP switch* ke mode-3 seperti pada Tabel 3.1 sebagai berikut[13]:

Tabel 3.1 Posisi DIP Switch Mode-3

	Pin 1	Pin 2	Pin 3	Pin 4
ON	O	O		O
OFF			O	

Penggunaan sensor sebanyak 36 sensor ini membuat peneliti harus harus mengisikan sebuah program *1 wire* untuk mendapatkan kode family atau ROM Code masing-masing DS18B20 sebelum program mengoperasikan DS18B20 dijalankan. Berikut ini merupakan langkah-langkah untuk mengisikan sebuah program pada mikrokontroler H8/3069F agar kode family atau ROM Code masing-masing DS18B20 dapat terbaca.


```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\ciTyWePuNyA>E:
E:\>cd tugas akhir/program\lwire
E:\TUGAS AKHIR\PROGRAM\lwire>ls
Makefile  ds1820.c      interrupt.h  main.lss   main.out   rom3068f.ld  uart.h
delay.c   ds1820.h      ioH83069.h  main.map  onewire.c  start.S     vects.c
delay.h   h8write.exe   main.c      main.mot  onewire.h  uart.c

E:\TUGAS AKHIR\PROGRAM\lwire>h8write.exe -3069 -f20 main.mot com9
H8/3069F is ready!  2002/5/20 Yukio Mituiwa.
writing
WARNING:This Line dosen't start with"S".
Address Size seems wrong
WARNING:This Line dosen't start with"S".
Address Size seems wrong
.....
EEPROM Writing is succeeded.
E:\TUGAS AKHIR\PROGRAM\lwire>

```

Gambar 3.6 Langkah-langkah saat melakukan writing program pembacaan ROM Code DS18B20 ke modul mikrokontroler H8/3069F

Setelah melakukan *writing* ubahlah posisi *DIP switch* ke mode-7 untuk melakukan *reading* seperti yang ditunjukkan pada Tabel 3.2[13]

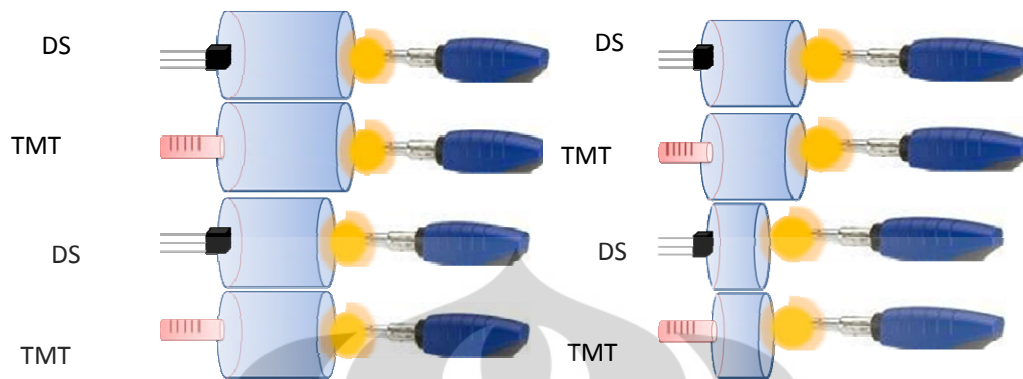
Tabel 3.2 Posisi DIP Switch Mode-7

	Pin 1	Pin 2	Pin 3	Pin 4
ON	O	O	O	
OFF				O

3.1.2. Proses Kalibrasi Sensor DS18B20

Kalibrasi merupakan proses verifikasi bahwa suatu akurasi alat ukur sesuai dengan rancangannya.[14] Kalibrasi dilakukan dengan membandingkan suatu standar yang terhubung dengan standar nasional maupun internasional. Pada penelitian ini peneliti membandingkan data temperatur yang terbaca oleh DS18B20 dengan temperatur yang terbaca oleh termometer raksa. Proses kalibrasi ini dilakukan dengan menggunakan sumber panas yang sama dengan memvariasikan jarak untuk mendapatkan variasi temperatur. Agar temperatur yang terukur oleh kedua sensor tersebut stabil maka peneliti menggunakan selubung yang panjangnya divariasikan sesuai dengan jarak antara sensor dan sumber panas. Gambar 3.7 berikut ini merupakan ilustrasi saat

proses pengkalibrasian dilakukan.



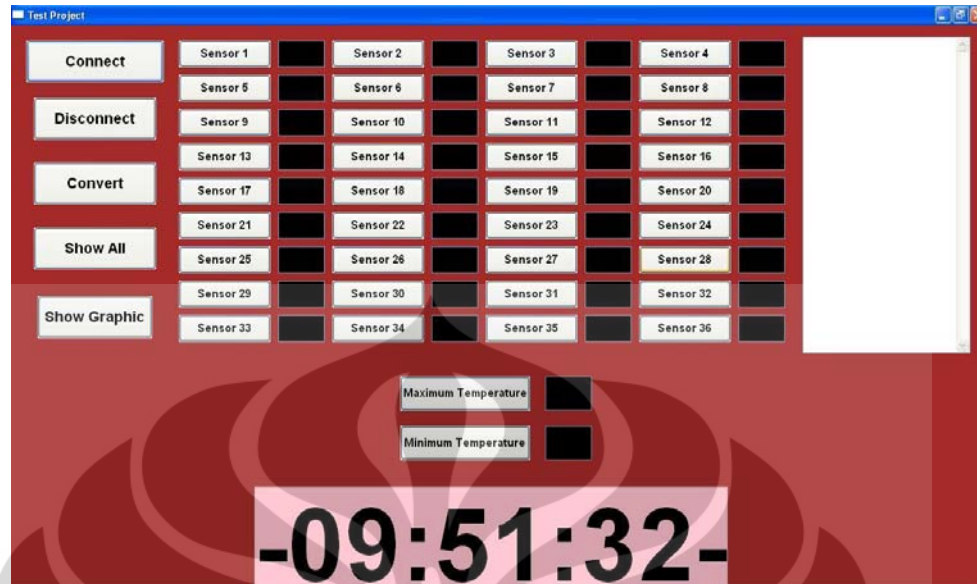
Gambar 3.7 Ilustrasi proses pengkalibrasian DS18B20 dengan termometer raksa

3.2 Perangkat Lunak

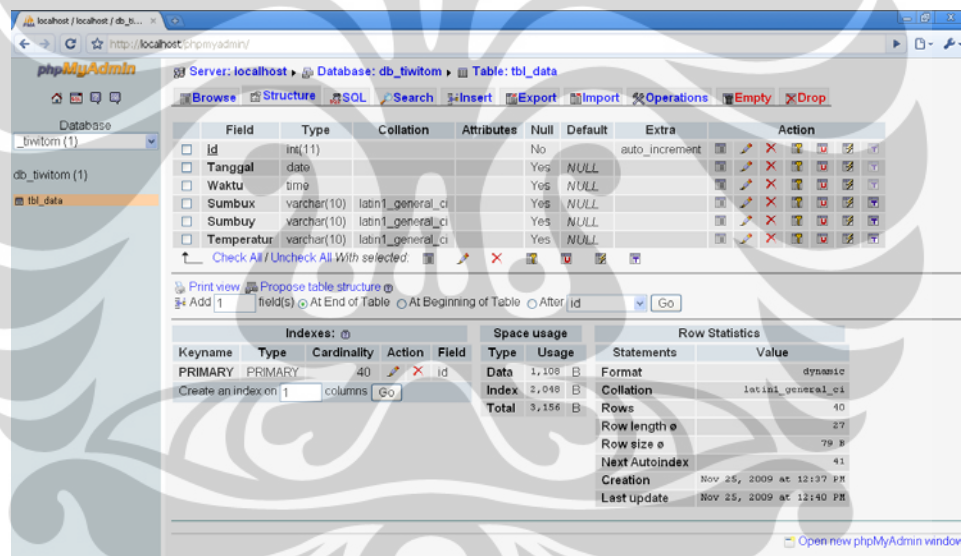
Pada penelitian ini digunakan perangkat lunak pada mikrokontroler dan komputer. Perangkat lunak pada mikrokontroler digunakan untuk mengaktifkan dan mengoperasikan sensor DS18B20. Sedangkan perangkat lunak pada komputer digunakan untuk menampilkan data seluruh sensor (baik satu persatu maupun secara langsung), mengolah data, menampilkan grafik distribusi temperatur, dan menampilkan letak sumber panas.

3.2.1. Tampilan Data pada GUI dan Database

Tampilan yang dibuat berupa GUI (*Graphical User Interface*). Data yang dikirim mikrokontroler H8/3069F terlebih dahulu diolah, kemudian ditampilkan pada GUI dan disimpan pada *database*. Perangkat lunak pada GUI penelitian ini menggunakan bahasa Python. *Library* Python yang digunakan untuk membuat tampilan GUI pada penelitian ini adalah wxPython. *Library* Python yang digunakan untuk membaca data dari *database* pada penelitian ini adalah MySQL-Python. *Database* yang digunakan untuk menampung data-data dari sensor adalah MySQL. Data yang diterima dicatat waktu pengambilan datanya serta data *real* dari tiap sensor. Hasil yang sudah ditampung pada *database* dapat yang langsung segera ditampilkan pada GUI dan dianalisis. Gambar 3.8 merupakan tampilan GUI sebelum komputer dihubungkan dengan mikrokontroler H8/3069F melalui serial, saat mengkonversi temperatur baik satu persatu atau seluruh sensor, dan saat mengambil data temperatur sensor DS18B20.



Gambar 3.8 Tampilan GUI sebelum di hubungkan ke mikrokontroler



Gambar 3.9 Tampilan home database

3.2.2. Pengolahan Data Temperatur dengan Algoritma Finite Difference

Untuk mendapatkan distribusi temperatur peneliti menggunakan algoritma persamaan diferensial parsial, selanjutnya akan dipersingkat menjadi PDP yang difokuskan dalam formulasi *Finite-Difference*. PDP dapat dibagi menjadi 3 jenis, yaitu persamaan diferensial eliptik, parabolik dan hiperbolik. Pada penelitian ini

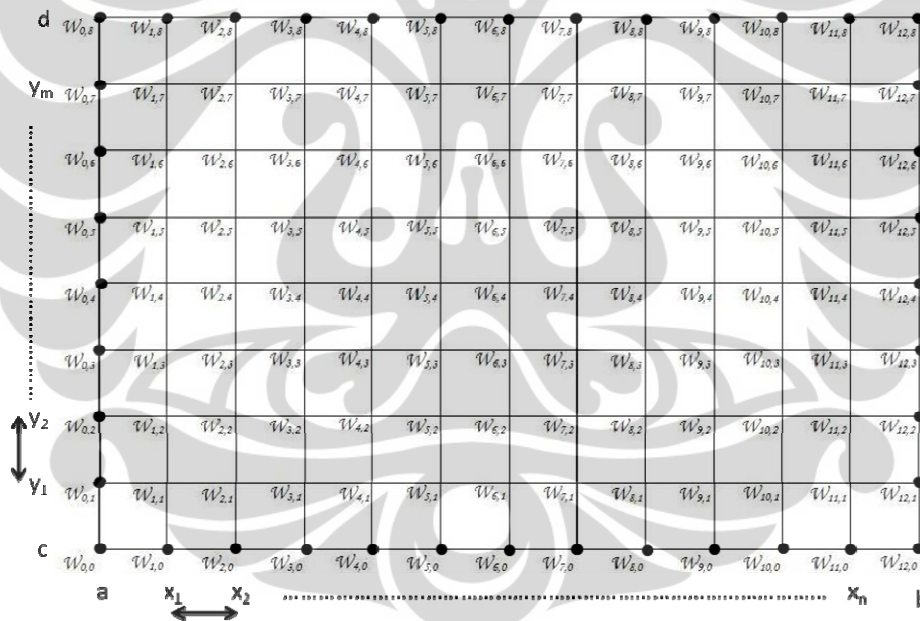
peneliti menggunakan PDP eliptik untuk mendapatkan distribusi temperaturnya. Persamaan PDP eliptik dinyatakan dalam persamaan 3.1 sebagai berikut:

$$\frac{\partial^2 u}{\partial x^2}(x, y) + \frac{\partial^2 u}{\partial y^2}(x, y) = f(x, y) \quad (3.1)$$

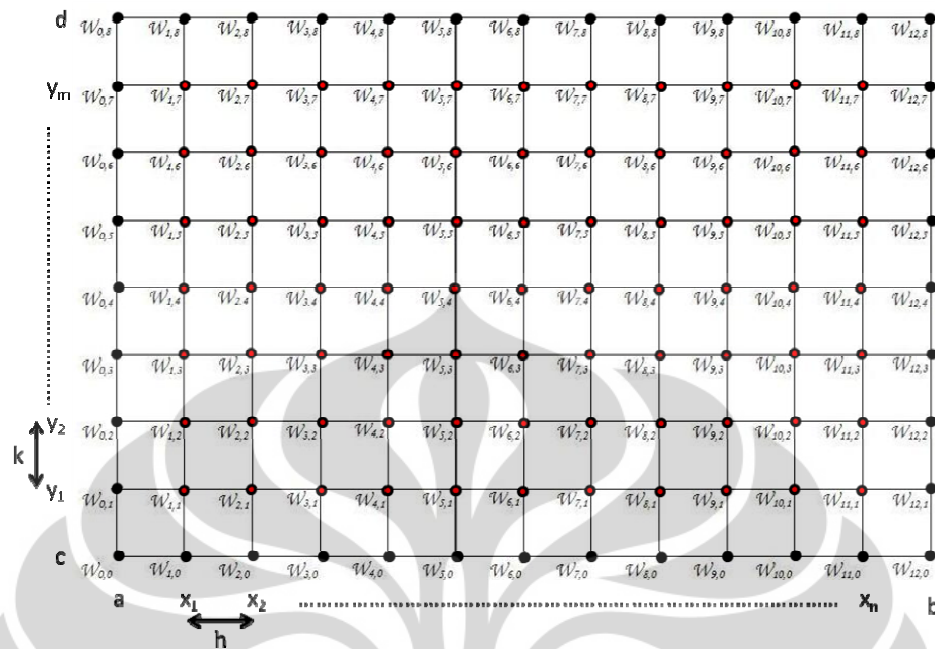
Dalam bidang fisika, persamaan (3.1) dikenal sebagai *persamaan poisson*. Jika $f(x, y) = 0$ [15], maka diperoleh persamaan yang lebih sederhana seperti persamaan berikut:

$$\frac{\partial^2 u}{\partial x^2}(x, y) + \frac{\partial^2 u}{\partial y^2}(x, y) = 0 \quad (3.2)$$

Persamaan (3.2) biasa disebut sebagai *persamaan laplace*. Dimana $R = [(x, y)|a < x < b, c < y < d]$. Maksudnya, variasi titik-titik x berada di antara a dan b. Demikian pula dengan variasi titik-titik y, dibatasi mulai dari c sampai d seperti yang terlihat pada Gambar 3.10. [15]



Gambar 3.10: Susunan *syarat batas* untuk mendapatkan distribusi temperatur pada sisi permukaan inkubator



Gambar 3.11: Skema *grid lines* dan *mesh points* pada penentuan distribusi temperatur

Dimana:

$W_{1,0}$ = Sensor 1	$W_{12,2}$ = Sensor 13	$W_{5,8}$ = Sensor 25
$W_{2,0}$ = Sensor 2	$W_{12,3}$ = Sensor 14	$W_{4,8}$ = Sensor 26
$W_{3,0}$ = Sensor 3	$W_{12,4}$ = Sensor 15	$W_{3,8}$ = Sensor 27
$W_{4,0}$ = Sensor 4	$W_{12,5}$ = Sensor 16	$W_{2,8}$ = Sensor 28
$W_{5,0}$ = Sensor 5	$W_{12,6}$ = Sensor 17	$W_{1,8}$ = Sensor 29
$W_{6,0}$ = Sensor 6	$W_{12,7}$ = Sensor 18	$W_{0,7}$ = Sensor 30
$W_{7,0}$ = Sensor 7	$W_{11,8}$ = Sensor 19	$W_{0,6}$ = Sensor 31
$W_{8,0}$ = Sensor 8	$W_{10,8}$ = Sensor 20	$W_{0,5}$ = Sensor 32
$W_{9,0}$ = Sensor 9	$W_{9,8}$ = Sensor 21	$W_{0,4}$ = Sensor 33
$W_{10,0}$ = Sensor 10	$W_{8,8}$ = Sensor 22	$W_{0,3}$ = Sensor 34
$W_{11,0}$ = Sensor 11	$W_{7,8}$ = Sensor 23	$W_{0,2}$ = Sensor 35
$W_{12,1}$ = Sensor 12	$W_{6,8}$ = Sensor 24	$W_{0,1}$ = Sensor 36

Jika h adalah jarak interval antar titik yang saling bersebelahan pada titik-titik dalam rentang horizontal a dan b , maka titik-titik variasi di antara a dan b dapat diketahui melalui rumus berikut ini:

$$x_i = a + ih, \text{ dimana } i = 1, 2, \dots, n$$

dimana a adalah titik awal pada sumbu horisontal x . Demikian pula pada sumbu y . Jika k adalah jarak interval antar titik yang bersebelahan pada titik-titik dalam rentang vertikal c dan d , maka titik-titik variasi di antara c dan d dapat diketahui melalui rumus berikut:[15]

$$y_j = c + jk, \text{ dimana } j = 1, 2, \dots, m$$

dimana c adalah titik awal pada sumbu vertikal y . Perhatikan Gambar 3.2, garis-garis yang sejajar sumbu horisontal, $y = y_i$ dan garis-garis yang sejajar sumbu vertikal, $x = x_i$ disebut *grid lines*. Sementara titik-titik perpotongan antara garis-garis horisontal dan vertikal dinamakan *mesh points*. [15]

Turunan kedua dari persamaan (3.1) dapat dinyatakan dalam rumus *centered-difference* sebagai berikut:

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j) = \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j))}{h^2} - \frac{h^2}{12} \frac{\partial^4 u}{\partial x^4}(\xi_i, y_j) \quad (3.3)$$

$$\frac{\partial^2 u}{\partial y^2}(x_i, y_j) = \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1}))}{k^2} - \frac{k^2}{12} \frac{\partial^4 u}{\partial y^4}(\xi_i, \eta_j) \quad (3.4)$$

Metode *Finite-Difference* biasanya mengabaikan suku yang terakhir, sehingga cukup dinyatakan sebagai berikut:

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j) = \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j))}{h^2} \quad (3.5)$$

$$\frac{\partial^2 u}{\partial y^2}(x_i, y_j) = \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1}))}{k^2} \quad (3.6)$$

Pengabaian suku terakhir otomatis menimbulkan *error* yang dinamakan *truncation error*. Jadi, ketika suatu persamaan diferensial diolah secara numerik dengan metode *Finite-Difference*, maka solusinya pasti meleset alias keliru "sedikit", dikarenakan adanya *truncation error* tersebut. Akan tetapi, nilai *error* tersebut dapat ditolerir hingga batas-batas tertentu. Setelah itu persamaan (3.5) dan (3.6) disubstitusi ke persamaan (3.1), [15] hasilnya adalah

$$\frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j))}{h^2} + \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1}))}{k^2} = f(x_i, y_j) \quad (3.7)$$

Pengertian syarat batas disini adalah bagian tepi atau bagian pinggir dari susunan *mesh points*. Pada penelitian ini yang menjadi syarat batas adalah nilai temperatur yang terbaca oleh sensor termometer digital DS18B20. Jumlah syarat batas atau sensor yang diletakan di tepi permukaan inkubator dengan panjang kali

lebar 60cm x 40 cm adalah 36 buah. Sebelumnya peneliti nyatakan bahwa $y(x_{i+1}) = w_{i+1}$ dan $y(x_i) = w_i$ serta $y(x_{i-1}) = w_{i-1}$. [15] Sehingga persamaan (3.7) dinyatakan dalam notasi w sebagai berikut:

$$\frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{h^2} + \frac{w_{i,j+1} - 2w_{i,j} + w_{i,j-1}}{k^2} = f(x_i, y_j) \quad (3.8)$$

$$w_{i+1,j} - 2w_{i,j} + w_{i-1,j} + \frac{h^2}{k^2} (w_{i,j+1} - 2w_{i,j} + w_{i,j-1}) = h^2 f(x_i, y_j) \quad (3.9)$$

$$w_{i+1,j} - 2w_{i,j} + w_{i-1,j} + \frac{h^2}{k^2} w_{i,j+1} - 2\frac{h^2}{k^2} w_{i,j} + \frac{h^2}{k^2} w_{i,j-1} = h^2 f(x_i, y_j) \quad (3.10)$$

$$2 \left[1 + \frac{h^2}{k^2} \right] w_{i,j} - (w_{i+1,j} + w_{i-1,j}) - \frac{h^2}{k^2} (w_{i,j+1} + w_{i,j-1}) = -h^2 f(x_i, y_j) \quad (3.11)$$

Dengan h dan k sama dengan 5 cm maka persamaan akhirnya menjadi:

$$4w_{i,j} - w_{i+1,j} - w_{i-1,j} - w_{i,j+1} - w_{i,j-1} = 0 \quad (3.12)$$

dimana $i = 1, 2, \dots, n - 1$ dan $j = 1, 2, \dots, m - 1$, dengan syarat batas sebagai berikut:

$$w_{0,j} = g(x_0, y_j) \quad w_{n,j} = g(x_n, y_j) \quad j = 1, 2, \dots, m - 1;$$

$$w_{i,0} = g(x_i, y_0) \quad w_{i,m} = g(x_i, y_m) \quad i = 1, 2, \dots, n - 1.$$

Lalu kita akan memperoleh sistem persamaan linear yang selanjutnya dapat dinyatakan sebagai bentuk operasi matrik berikut [15]:

$$Aw = b$$

dimana A adalah matrik tridiagonal dengan orde $N \times N$ sesuai dengan jumlah syarat batas yang digunakan maka pada penelitian ini nilai i divariasikan dari $i = 1$ sampai dengan $i = 11$ dan dari $j = 1$ sampai $j = 7$. Berdasarkan persamaan (3.12), yang merupakan rumusan akhir metode *finite difference* untuk PDP Eliptik maka diperoleh [15]:

Pada $i = 1, j = 1$

$$4w_{1,1} - w_{2,1} - w_{0,1} - w_{1,0} - w_{1,2} = 0$$

$$4w_{1,1} - w_{2,1} - w_{1,2} = w_{0,1} + w_{1,0} \quad (3.13)$$

Pada $i = 1, j = 2$

$$\begin{aligned} 4w_{1,2} - w_{2,2} - w_{0,2} - w_{1,1} - w_{1,3} &= 0 \\ 4w_{1,2} - w_{2,2} - w_{1,1} - w_{1,3} &= w_{0,2} \end{aligned} \quad (3.14)$$

Pada $i = 1, j = 3$

$$\begin{aligned} 4w_{1,3} - w_{2,3} - w_{0,3} - w_{1,2} - w_{1,4} &= 0 \\ 4w_{1,3} - w_{2,3} - w_{1,2} - w_{1,4} &= w_{0,3} \end{aligned} \quad (3.15)$$

Pada $i = 1, j = 4$

$$\begin{aligned} 4w_{1,4} - w_{2,4} - w_{0,4} - w_{1,3} - w_{1,5} &= 0 \\ 4w_{1,4} - w_{2,4} - w_{1,3} - w_{1,5} &= w_{0,4} \end{aligned} \quad (3.16)$$

Pada $i = 1, j = 5$

$$\begin{aligned} 4w_{1,5} - w_{2,5} - w_{0,5} - w_{1,4} - w_{1,6} &= 0 \\ 4w_{1,5} - w_{2,5} - w_{1,4} - w_{1,6} &= w_{0,5} \end{aligned} \quad (3.17)$$

Pada $i = 1, j = 6$

$$\begin{aligned} 4w_{1,6} - w_{2,6} - w_{0,6} - w_{1,5} - w_{1,7} &= 0 \\ 4w_{1,6} - w_{2,6} - w_{1,5} - w_{1,7} &= w_{0,6} \end{aligned} \quad (3.18)$$

Pada $i = 1, j = 7$

$$\begin{aligned} 4w_{1,7} - w_{2,7} - w_{0,7} - w_{1,6} - w_{1,8} &= 0 \\ 4w_{1,7} - w_{2,7} - w_{1,6} &= w_{0,7} + w_{1,8} \end{aligned} \quad (3.19)$$

Pada $i = 2, j = 1$

$$\begin{aligned} 4w_{2,1} - w_{3,1} - w_{1,1} - w_{2,0} - w_{2,2} &= 0 \\ 4w_{2,1} - w_{3,1} - w_{1,1} - w_{2,2} &= w_{2,0} \end{aligned} \quad (3.20)$$

Pada $i = 2, j = 2$

$$4w_{2,2} - w_{3,2} - w_{1,2} - w_{2,1} - w_{1,3} = 0 \quad (3.21)$$

Pada $i = 2, j = 3$

$$4w_{2,3} - w_{3,3} - w_{1,3} - w_{2,2} - w_{2,4} = 0 \quad (3.22)$$

Pada $i = 2, j = 4$

$$4w_{2,4} - w_{3,4} - w_{1,4} - w_{2,3} - w_{2,5} = 0 \quad (3.23)$$

Pada $i = 2, j = 5$

$$4w_{2,5} - w_{3,5} - w_{1,5} - w_{2,4} - w_{2,6} = 0 \quad (3.24)$$

Pada $i = 2, j = 6$

$$4w_{2,6} - w_{3,6} - w_{1,6} - w_{2,5} - w_{2,7} = 0 \quad (3.25)$$

Pada $i = 2, j = 7$

$$\begin{aligned} 4w_{2,7} - w_{3,7} - w_{1,7} - w_{2,6} - w_{2,8} &= 0 \\ 4w_{2,7} - w_{3,7} - w_{1,7} - w_{2,6} &= w_{2,8} \end{aligned} \quad (3.26)$$

Pada $i = 3, j = 1$

$$\begin{aligned} 4w_{3,1} - w_{4,1} - w_{2,1} - w_{3,0} - w_{3,2} &= 0 \\ 4w_{3,1} - w_{4,1} - w_{2,1} - w_{3,2} &= w_{3,0} \end{aligned} \quad (3.27)$$

Pada $i = 3, j = 2$

$$4w_{3,2} - w_{4,2} - w_{2,2} - w_{3,1} - w_{3,3} = 0 \quad (3.28)$$

Pada $i = 3, j = 3$

$$4w_{3,3} - w_{4,3} - w_{2,3} - w_{3,2} - w_{3,4} = 0 \quad (3.29)$$

Pada $i = 3, j = 4$

$$4w_{3,4} - w_{4,4} - w_{2,4} - w_{3,3} - w_{3,5} = 0 \quad (3.30)$$

Pada $i = 3, j = 5$

$$4w_{3,5} - w_{4,5} - w_{2,5} - w_{3,4} - w_{3,6} = 0 \quad (3.31)$$

Pada $i = 3, j = 6$

$$4w_{3,6} - w_{4,6} - w_{2,6} - w_{3,5} - w_{3,7} = 0 \quad (3.32)$$

Pada $i = 3, j = 7$

$$\begin{aligned} 4w_{3,7} - w_{4,7} - w_{2,7} - w_{3,6} - w_{3,8} &= 0 \\ 4w_{3,7} - w_{4,7} - w_{2,7} - w_{3,6} &= w_{3,8} \end{aligned} \quad (3.33)$$

Pada $i = 4, j = 1$

$$4w_{4,1} - w_{5,1} - w_{3,1} - w_{4,0} - w_{4,2} = 0$$

$$4w_{4,1} - w_{5,1} - w_{3,1} - w_{4,2} = w_{4,0} \quad (3.34)$$

Pada $i = 4, j = 2$

$$4w_{4,2} - w_{5,2} - w_{3,2} - w_{4,1} - w_{4,3} = 0 \quad (3.35)$$

Pada $i = 4, j = 3$

$$4w_{4,3} - w_{5,3} - w_{3,3} - w_{4,2} - w_{4,4} = 0 \quad (3.36)$$

Pada $i = 4, j = 4$

$$4w_{4,4} - w_{5,4} - w_{3,4} - w_{4,3} - w_{4,5} = 0 \quad (3.37)$$

Pada $i = 4, j = 5$

$$4w_{4,5} - w_{5,5} - w_{3,5} - w_{4,4} - w_{4,6} = 0 \quad (3.38)$$

Pada $i = 4, j = 6$

$$4w_{4,6} - w_{5,6} - w_{3,6} - w_{4,5} - w_{4,7} = 0 \quad (3.39)$$

Pada $i = 4, j = 7$

$$4w_{4,7} - w_{5,7} - w_{3,7} - w_{4,6} - w_{4,8} = 0$$

$$4w_{4,7} - w_{5,7} - w_{3,7} - w_{4,6} = w_{4,8} \quad (3.40)$$

Pada $i = 5, j = 1$

$$4w_{5,1} - w_{6,1} - w_{4,1} - w_{5,0} - w_{5,2} = 0$$

$$4w_{5,1} - w_{6,1} - w_{4,1} - w_{5,2} = w_{5,0} \quad (3.41)$$

Pada $i = 5, j = 2$

$$4w_{5,2} - w_{6,2} - w_{4,2} - w_{5,1} - w_{5,3} = 0 \quad (3.42)$$

Pada $i = 5, j = 3$

$$4w_{5,3} - w_{6,3} - w_{4,3} - w_{5,2} - w_{5,4} = 0 \quad (3.43)$$

Pada $i = 5, j = 4$

$$4w_{5,4} - w_{6,4} - w_{4,4} - w_{5,3} - w_{5,5} = 0 \quad (3.44)$$

Pada $i = 5, j = 5$

$$4w_{5,5} - w_{6,5} - w_{4,5} - w_{5,4} - w_{5,6} = 0 \quad (3.45)$$

Pada $i = 5, j = 6$

$$4w_{5,6} - w_{6,6} - w_{4,6} - w_{5,5} - w_{5,7} = 0 \quad (3.46)$$

Pada $i = 5, j = 7$

$$4w_{5,7} - w_{6,7} - w_{4,7} - w_{5,6} - w_{5,8} = 0$$

$$4w_{5,7} - w_{6,7} - w_{4,7} - w_{5,6} = w_{5,8} \quad (3.47)$$

Pada $i = 6, j = 1$

$$4w_{6,1} - w_{7,1} - w_{5,1} - w_{6,0} - w_{6,2} = 0$$

$$4w_{6,1} - w_{7,1} - w_{5,1} - w_{6,2} = w_{6,0} \quad (3.48)$$

Pada $i = 6, j = 2$

$$4w_{6,2} - w_{7,2} - w_{5,2} - w_{6,1} - w_{6,3} = 0 \quad (3.49)$$

Pada $i = 6, j = 3$

$$4w_{6,3} - w_{7,3} - w_{5,3} - w_{6,2} - w_{6,4} = 0 \quad (3.50)$$

Pada $i = 6, j = 4$

$$4w_{6,4} - w_{7,4} - w_{5,4} - w_{6,3} - w_{6,5} = 0 \quad (3.51)$$

Pada $i = 6, j = 5$

$$4w_{6,5} - w_{7,5} - w_{5,5} - w_{6,4} - w_{6,6} = 0 \quad (3.52)$$

Pada $i = 6, j = 6$

$$4w_{6,6} - w_{7,6} - w_{5,6} - w_{6,5} - w_{6,7} = 0 \quad (3.53)$$

Pada $i = 6, j = 7$

$$4w_{7,7} - w_{8,7} - w_{6,7} - w_{7,6} - w_{7,8} = 0$$

$$4w_{7,7} - w_{8,7} - w_{6,7} - w_{7,6} = w_{7,8} \quad (3.54)$$

Pada $i = 7, j = 1$

$$4w_{7,1} - w_{8,1} - w_{6,1} - w_{7,0} - w_{7,2} = 0$$

$$4w_{7,1} - w_{8,1} - w_{6,1} - w_{7,2} = w_{7,0} \quad (3.55)$$

Pada $i = 7, j = 2$

$$4w_{7,2} - w_{8,2} - w_{6,2} - w_{7,1} - w_{7,3} = 0 \quad (3.56)$$

Pada $i = 7, j = 3$

$$4w_{7,3} - w_{8,3} - w_{6,3} - w_{7,2} - w_{7,4} = 0 \quad (3.57)$$

Pada $i = 7, j = 4$

$$4w_{7,4} - w_{8,4} - w_{6,4} - w_{7,3} - w_{7,5} = 0 \quad (3.58)$$

Pada $i = 7, j = 5$

$$4w_{7,5} - w_{8,5} - w_{6,5} - w_{7,4} - w_{7,6} = 0 \quad (3.59)$$

Pada $i = 7, j = 6$

$$4w_{7,6} - w_{8,6} - w_{6,6} - w_{7,5} - w_{7,7} = 0 \quad (3.60)$$

Pada $i = 7, j = 7$

$$\begin{aligned} 4w_{7,7} - w_{8,7} - w_{6,7} - w_{7,6} - w_{7,8} &= 0 \\ 4w_{7,7} - w_{8,7} - w_{6,7} - w_{7,6} &= w_{7,8} \end{aligned} \quad (3.61)$$

Pada $i = 8, j = 1$

$$\begin{aligned} 4w_{8,1} - w_{9,1} - w_{7,1} - w_{8,0} - w_{8,2} &= 0 \\ 4w_{8,1} - w_{9,1} - w_{8,1} - w_{8,2} &= w_{8,0} \end{aligned} \quad (3.62)$$

Pada $i = 8, j = 2$

$$4w_{8,2} - w_{9,2} - w_{7,2} - w_{8,1} - w_{8,3} = 0 \quad (3.63)$$

Pada $i = 8, j = 3$

$$4w_{8,3} - w_{9,3} - w_{7,3} - w_{8,2} - w_{8,4} = 0 \quad (3.64)$$

Pada $i = 8, j = 4$

$$4w_{8,4} - w_{9,4} - w_{7,4} - w_{8,3} - w_{8,5} = 0 \quad (3.65)$$

Pada $i = 8, j = 5$

$$4w_{8,5} - w_{9,5} - w_{7,5} - w_{8,4} - w_{8,6} = 0 \quad (3.66)$$

Pada $i = 8, j = 6$

$$4w_{8,6} - w_{9,6} - w_{7,6} - w_{8,5} - w_{8,7} = 0 \quad (3.67)$$

Pada $i = 8, j = 7$

$$\begin{aligned} 4w_{8,7} - w_{9,7} - w_{7,7} - w_{8,6} - w_{8,8} &= 0 \\ 4w_{8,7} - w_{9,7} - w_{7,7} - w_{8,6} &= w_{8,8} \end{aligned} \quad (3.68)$$

Pada $i = 9, j = 1$

$$4w_{9,1} - w_{10,1} - w_{8,1} - w_{9,0} - w_{9,2} = 0$$

$$4w_{9,1} - w_{10,1} - w_{8,1} - w_{9,2} = w_{9,0} \quad (3.69)$$

Pada $i = 9, j = 2$

$$4w_{9,2} - w_{10,2} - w_{8,2} - w_{9,1} - w_{9,3} = 0 \quad (3.70)$$

Pada $i = 9, j = 3$

$$4w_{9,3} - w_{10,3} - w_{8,3} - w_{9,2} - w_{9,4} = 0 \quad (3.71)$$

Pada $i = 9, j = 4$

$$4w_{9,4} - w_{10,4} - w_{8,4} - w_{9,3} - w_{9,5} = 0 \quad (3.72)$$

Pada $i = 9, j = 5$

$$4w_{9,5} - w_{10,5} - w_{8,5} - w_{9,4} - w_{9,6} = 0 \quad (3.73)$$

Pada $i = 9, j = 6$

$$4w_{9,6} - w_{10,6} - w_{8,6} - w_{9,5} - w_{9,7} = 0 \quad (3.74)$$

Pada $i = 9, j = 7$

$$4w_{9,7} - w_{10,7} - w_{8,7} - w_{9,6} - w_{9,8} = 0$$

$$4w_{9,7} - w_{10,7} - w_{8,7} - w_{9,6} = w_{9,8} \quad (3.75)$$

Pada $i = 10, j = 1$

$$4w_{10,1} - w_{11,1} - w_{9,1} - w_{10,0} - w_{10,2} = 0$$

$$4w_{10,1} - w_{11,1} - w_{9,1} - w_{10,2} = w_{10,0} \quad (3.76)$$

Pada $i = 10, j = 2$

$$4w_{10,2} - w_{11,2} - w_{9,2} - w_{10,1} - w_{10,3} = 0 \quad (3.77)$$

Pada $i = 10, j = 3$

$$4w_{10,3} - w_{11,3} - w_{9,3} - w_{10,2} - w_{10,4} = 0 \quad (3.78)$$

Pada $i = 10, j = 4$

$$4w_{10,4} - w_{11,4} - w_{9,4} - w_{10,3} - w_{10,5} = 0 \quad (3.79)$$

Pada $i = 10, j = 5$

$$4w_{10,5} - w_{11,5} - w_{9,5} - w_{10,4} - w_{10,6} = 0 \quad (3.80)$$

Pada $i = 10, j = 6$

$$4w_{10,6} - w_{11,6} - w_{9,6} - w_{10,5} - w_{10,7} = 0 \quad (3.81)$$

Pada $i = 10, j = 7$

$$\begin{aligned} 4w_{10,7} - w_{11,7} - w_{9,7} - w_{10,6} - w_{10,8} &= 0 \\ 4w_{10,7} - w_{11,7} - w_{9,7} - w_{10,6} &= w_{10,8} \end{aligned} \quad (3.82)$$

Pada $i = 11, j = 1$

$$\begin{aligned} 4w_{11,1} - w_{12,1} - w_{10,1} - w_{11,0} - w_{11,2} &= 0 \\ 4w_{11,1} - w_{10,1} - w_{11,2} &= w_{12,1} + w_{11,0} \end{aligned} \quad (3.83)$$

Pada $i = 11, j = 2$

$$\begin{aligned} 4w_{11,2} - w_{12,2} - w_{10,2} - w_{11,1} - w_{11,3} &= 0 \\ 4w_{11,2} - w_{10,2} - w_{11,1} - w_{11,3} &= w_{12,2} \end{aligned} \quad (3.84)$$

Pada $i = 11, j = 3$

$$\begin{aligned} 4w_{11,3} - w_{12,3} - w_{10,3} - w_{11,2} - w_{11,4} &= 0 \\ 4w_{11,3} - w_{10,3} - w_{11,2} - w_{11,4} &= w_{12,3} \end{aligned} \quad (3.85)$$

Pada $i = 11, j = 4$

$$\begin{aligned} 4w_{11,4} - w_{12,4} - w_{10,4} - w_{11,3} - w_{11,5} &= 0 \\ 4w_{11,4} - w_{10,4} - w_{11,3} - w_{11,5} &= w_{12,4} \end{aligned} \quad (3.86)$$

Pada $i = 11, j = 5$

$$\begin{aligned} 4w_{11,5} - w_{12,5} - w_{10,5} - w_{11,4} - w_{11,6} &= 0 \\ 4w_{11,5} - w_{10,5} - w_{11,4} - w_{11,6} &= w_{12,5} \end{aligned} \quad (3.87)$$

Pada $i = 11, j = 6$

$$\begin{aligned} 4w_{11,6} - w_{12,6} - w_{10,6} - w_{11,5} - w_{11,7} &= 0 \\ 4w_{11,6} - w_{10,6} - w_{11,5} - w_{11,7} &= w_{12,6} \end{aligned} \quad (3.88)$$

Pada $i = 11, j = 7$

$$\begin{aligned} 4w_{11,7} - w_{12,7} - w_{10,7} - w_{11,6} - w_{11,8} &= 0 \\ 4w_{11,7} - w_{10,7} - w_{11,6} &= w_{12,7} + w_{11,8} \end{aligned} \quad (3.89)$$

Setelah dilakukan penginisialisasian berdasarkan persamaan (3.13) sampai persamaan (3.89) maka diperoleh matrik A berukuran 77×77 , dimana matrik yang dihasilkan adalah sebagai berikut:

$$\begin{bmatrix} 4. -1. 0. \dots, 0. 0. 0. \\ -1. 4. -1. \dots, 0. 0. 0. \\ 0. -1. 4. \dots, 0. 0. 0. \\ \dots \\ 0. 0. 0. \dots, 4. -1. 0. \\ 0. 0. 0. \dots, -1. 4. -1. \\ 0. 0. 0. \dots, 0. -1. 4. \end{bmatrix}$$

Matrik b merupakan matrik yang nilainya ditentukan berdasarkan syarat batas seperti matrik berikut ini:

$$\begin{bmatrix} w_{0,1} + w_{1,0} \\ w_{0,2} \\ w_{0,3} \\ \dots \\ w_{12,5} \\ w_{12,6} \\ w_{12,7} + w_{11,8} \end{bmatrix}$$

Maka matrik w yang merupakan nilai distribusi temperatur yang ingi dicari didapatkan sesuai dengan matrik berikut:

$$\begin{bmatrix} w_{1,1} \\ w_{1,2} \\ w_{2,1} \\ \dots \\ w_{11,5} \\ w_{11,6} \\ w_{11,7} \end{bmatrix}$$

BAB 4

ANALISIS HASIL PENELITIAN

4.1 Perangkat Keras

Untuk memastikan sistem yang telah dibuat bekerja dengan baik sesuai dengan yang diinginkan, maka perlu dilakukan beberapa pengujian dan pengambilan data.

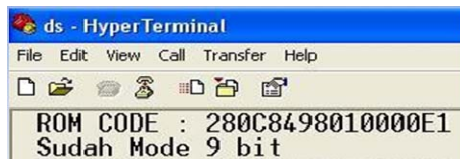
4.1.1 Hasil dan Analisis Data Rom Code DS18B20

Pada penelitian ini digunakan 36 sensor DS18B20 yang dijadikan sebagai syarat batas untuk menentukan distribusi temperatur. DS18B20 menggunakan komunikasi *1 wire* untuk tatap muka dengan mikrokontroler. Komunikasi *1 wire* dirancang untuk transfer data yang rendah dan menggabungkan sinyal data dan *power* dalam satu jalur. Kelebihannya terletak pada jalur data yang digunakan yakni hanya satu jalur *power*, data, dan satu jalur *ground*. Model komunikasi *1 wire* ini adalah *master-slave*, dimana mikrokontroler H8/3069F merupakan masternya dan DS18B20 adalah *device 1 wire* yang menjadi *slavenya*. Untuk membedakan satu *slave* dengan yang lainnya maka setiap *slave* memiliki *slave address* yang berbeda (meskipun *device slavenya* sama).

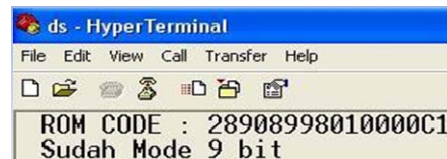
Untuk membedakan DS18B20 satu dengan yang lainnya DS18B20 memiliki kode family yang berbeda-beda yang tersimpan di Rom Codenya. Gambar 4.1(a),(b),(c),(d),(e), dan (f) merupakan contoh hasil data rom code pada 6 buah sensor DS18B20.

Terlihat bahwa semua DS18B20 memiliki kode family yang berbeda-beda. Hasil data rom code pada semua DS18B20 ini dimasukkan ke dalam program untuk penginisialisasian. Hal ini dilakukan untuk penamaan untuk masing-masing sensor agar dapat dibedakan. Selain mencari data rom code, dengan menggunakan program yang sama Pada Gambar 4.1(a), (b), (c), (d), (e), dan (f) terlihat tulisan "*Sudah Mode 9 bit*".

Hal ini berarti resolusi ADC (internal DS18B20) pada semua sensor DS18B20 semuanya dirubah ke dalam mode 9 bit. Hal ini juga untuk memastikan bahwa semua sensor DS18B20 membutuhkan waktu konversi yang sama.



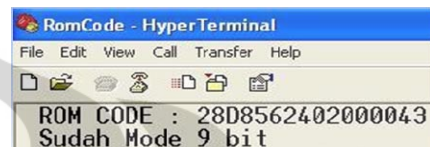
Gambar 4.1(a)



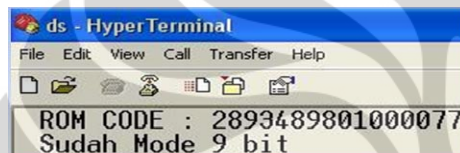
Gambar 4.1(b)



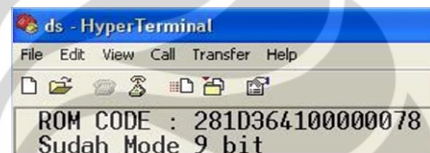
Gambar 4.1(c)



Gambar 4.1(d)



Gambar 4.1(e)



Gambar 4.1(f)

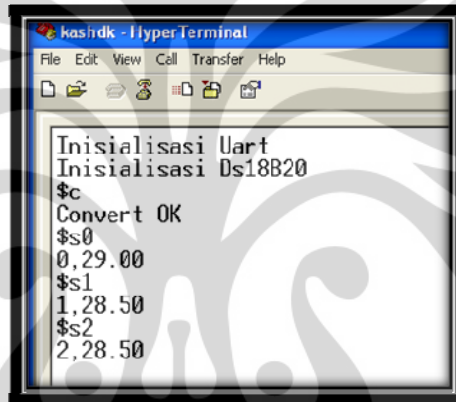
Gambar 4.1(a), (b), (c), (d), (e), dan (f) Data rom code pada hiperterminal

4.1.2 Hasil dan Analisis Data Temperatur

Setelah mendapatkan kode *family* semua sensor maka semua sensor sudah dapat diambil data temperaturnya. Data awal yang diperoleh adalah dalam bentuk hexadecimal lalu dirubah ke dalam biner lalu langsung dapat dikonversi kedalam data temperatur, perintah untuk mengkonversinya adalah menggunakan perintah 44h. Untuk pengujian pertama (sebelum diuji di GUI Python) peneliti menampilkan di hyperterminal terlebih dahulu. Pada perancangan sistem ini dibuat agar data tidak terus melakukan *looping*. Hal ini dilakukan agar pengambilan data dapat di perintah sesuai keinginan kita. Sehingga data temperatur yang akan diolah oleh finite difference merupakan data temperatur yang memang ingin kita dapatkan pada saat yang kita inginkan.

Gambar 4.2 menunjukkan potongan data yang ditampilkan di hyperterminal, untuk melakukan konversi temperatur maka kita harus menuliskan '\$c' terlebih dahulu, jika tidak ada kesalahan, baik dalam pemasangan pinnya ataupun kondisi sensor yang tidak baik. Pada data yang tampil di hyperterminal terlihat bahwa sebelum tampil data temperatur ada satu perintah '\$s0', perintah ini

berarti bahwa kita memerintahkan sensor 0 untuk menampilkan data temperatur di komputer, jika ada perintah '\$s1' perintah ini berarti bahwa kita memerintahkan sensor 0 untuk menampilkan data temperatur, begitupun untuk perintah-perintah selanjutnya. Angka 0, 1, dan selanjutnya menunjukkan nomor sensor yang sebelumnya sudah diberi nomor sesuai dengan data rom code yang telah diperoleh sebelumnya. Berdasarkan data temperatur yang terbaca di hyperterminal, maka dapat dipastikan bahwa program DS18B20 yang telah dibuat sudah dapat dipakai untuk membaca temperatur. Dapat dipastikan pula bahwa program yang dibuat telah dapat membedakan ke-36 sensor yang akan digunakan. Hal ini dapat dibuktikan dengan mengetest sensor satu persatu dan merubah temperaturnya.



```

kashdk - Hyper Terminal
File Edit View Cell Transfer Help
Inisialisasi Uart
Inisialisasi Ds18B20
$c
Convert OK
$s0
0,29.00
$s1
1,28.50
$s2
2,28.50

```

Gambar 4.2 Cara mengkonversi dan data yang tampil di hyperterminal

4.1.3. Hasil dan Analisis Data Kalibrasi Sensor DS18B20

Pada proses kalibrasi ini peneliti ingin menyetarakan atau menyesuaikan besaran-besaran hasil pengukuran dengan harga standar. Tujuannya adalah untuk mendapatkan hubungan antara keluaran dari sensor (tranducer) dengan harga yang sebenarnya dari *measurand*. Pada proses pengkalibrasian ini peneliti membandingkan data temperatur yang terbaca oleh DS18B20 dengan temperatur yang terbaca oleh termometer raksa. Proses kalibrasi ini dilakukan dengan menggunakan sumber panas (ujung solder) yang sama dengan memvariasikan jarak untuk mendapatkan variasi temperatur. Agar temperatur yang terukur oleh kedua sensor tersebut stabil maka peneliti menggunakan selubung yang panjangnya divariasikan sesuai dengan jarak antara sensor dan sumber panas

seperti yang telah dijelaskan pada bab 3. Tabel 4.1 merupakan data temperatur yang terukur oleh sensor DS18B20 dengan memvariasikan selubung (yang digambarkan pada bab 3) dimulai pada jarak 10 cm sampai 1 cm dimana masing-masing jarak dilakukan pengambilan data sebanyak 10 kali. Pengukuran yang dilakukan 10 kali ini dilakukan untuk mendapatkan nilai kepresisian sensor. Data pada Tabel 4.1 ini dapat dibandingkan dengan data pada Tabel 4.2 karena pada kedua tabel ini data yang didapatkan merupakan data yang diambil pada waktu, tempat, dan suhu ruang yang sama. Data-data ini diperoleh saat suhu ruang menunjukkan angka 27⁰ C.

Tabel 4.1 Data temperatur sensor DS18B20 (jarak antara sensor dan sumber panas semakin didekatkan)

Jarak (cm)	Temperatur Terukur di DS18B20 (Derajat Celsius)									
	Pengukuran Ke-									
	1	2	3	4	5	6	7	8	9	10
10	29	29	29	29	29	28.5	28.5	29	28.5	29
9	30	30	30	30	30	30	30	30	30	30
8	30	30.5	30.5	30.5	30.5	30.5	30.5	30.5	30.5	30
7	31	31	31	31	31	31	31	31	31	31
6	32.5	32.5	32.5	32.5	32.5	32.5	32.5	32.5	32.5	32.5
5	33.5	33.5	33.5	33.5	33.5	33.5	33.5	33.5	33.5	33.5
4	36.5	36.5	36.5	36.5	36.5	36.5	36.5	36.5	36.5	36.5
3	41	41	41	41	41	41	41	41	41	41
2	49.5	49.5	49.5	49.5	49.5	49.5	49.5	49.5	49.5	49.5
1	79	78.5	78.5	78.5	78.5	78.5	78.5	78.5	78.5	79

Tabel 4.2 Data temperatur sensor termometer raksa (jarak antara sensor dan sumber panas semakin didekatkan)

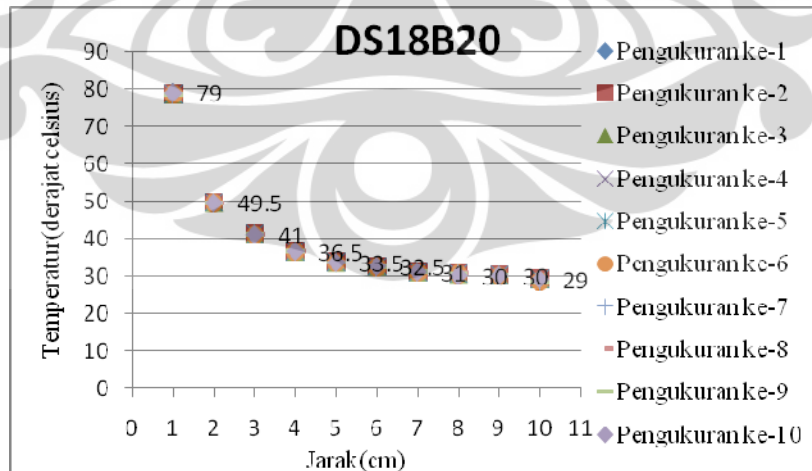
Jarak (cm)	Temperatur Terukur di Termometer Raksa (Derajat Celsius)									
	Pengukuran Ke-									
	1	2	3	4	5	6	7	8	9	10
10	28.5	28.5	28.5	28.5	28.5	29	29	29	29	29
9	30	30	30	30	30	30	30	30	30	30
8	31	31	31	31	30.5	30.5	31	31	31	31
7	31.5	31.5	31.5	31.5	31.5	31.5	31.5	31.5	31.5	31.5
6	32	32	32	32	32	32	32	32	32	32
5	34	34	34	34	34	34	34	34	34	34
4	37	37	37	37	37	37	37	37	37	37
3	40.5	40.5	40.5	40.5	40.5	41	40.5	40.5	40.5	40.5

2	49	49	49.5	49.5	49.5	49.5	49.5	49.5	49.5	49.5
1	78.5	78.5	79	79	79	79	79	79	79	79

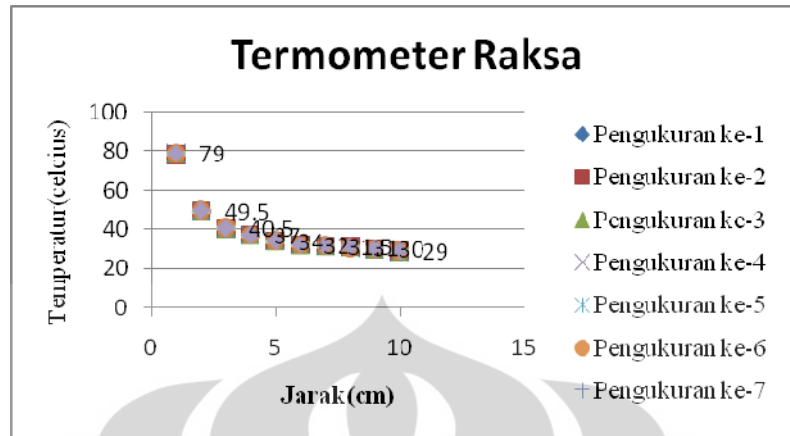
Tabel 4.3 Nilai rata-rata data temperatur sensor DS18B20 dibandingkan dengan termometer raksa

Jarak(cm)	Nilai Rata-rata pada DS18B20	Nilai Rata-rata pada Termometer
10	28.85	28.75
9	30	30
8	30.4	30.9
7	31	31.5
6	32.5	32
5	33.5	34
4	36.5	37
3	41	40.55
2	49.5	49.4
1	78.6	78.9

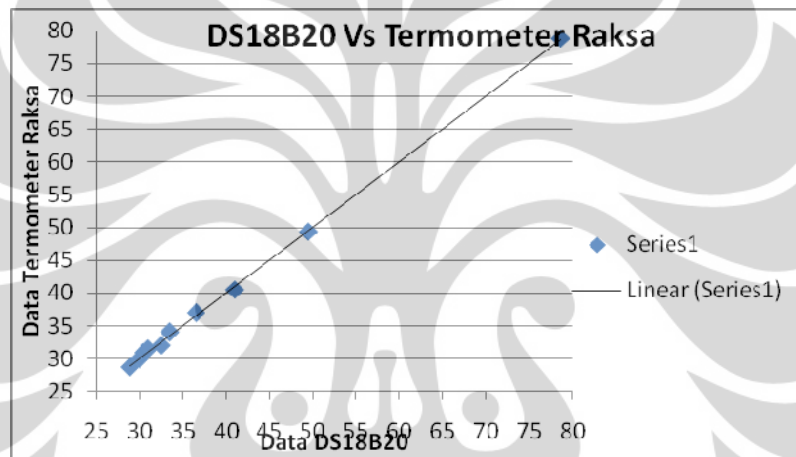
Berdasarkan data yang diperoleh dapat ditentukan beberapa karakteristik pada sensor tersebut, yaitu nilai sensitivitas, linieritas, presisi, akurasi, resolusi, dan histerisisnya. Karakteristik-karakteristik ini (kecuali histerisis) dapat kita ketahui berdasarkan hasil data yang diperoleh atau dengan melihat gambar grafik yang terdapat pada Gambar 4.3, 4.4, dan Gambar 4.5.



Gambar 4.3 Grafik hubungan antara jarak dan temperatur yang terukur pada DS18B20



Gambar 4.4 Grafik hubungan antara jarak dan temperatur yang terukur pada termometer raksa



Gambar 4.5 Grafik hubungan temperatur yang terukur pada DS18B20 dan termometer raksa

Untuk mendapatkan nilai histerisisnya maka peneliti harus melakukan kalibrasi yang arahnya merupakan kebalikan dari arah kalibrasi sebelumnya dengan tetap menggunakan objek, sumber panas, sensor, dan suhu ruang yang sama. Hal ini dapat didapatkan dengan merubah variasi jarak, jika sebelumnya pengukuran dilakukan dari jarak 10 sampai 1 cm, maka untuk mendapatkan histerisisnya penulis melakukan pengukuran dengan variasi data dimulai dari 1 sampai 10 cm.

Tabel 4.4 merupakan data temperatur yang terukur oleh sensor DS18B20 dengan memvariasikan selubung (yang digambarkan pada bab 3) dimulai pada jarak 1 cm sampai 10 cm dimana masing-masing jarak dilakukan pengambilan data sebanyak 10 kali. Sedangkan Tabel 4.5 merupakan data temperatur yang

terukur oleh sensor termometer raksa dengan memvariasikan selubung(yang digambarkan pada bab 3) dimulai pada jarak 1 cm sampai 10 cm dimana masing-masing jarak dilakukan pengambilan data sebanyak 10 kali.

Tabel 4.4 Data temperatur sensor DS18B20 (jarak antara sensor dan sumber panas semakin dijauhkan)

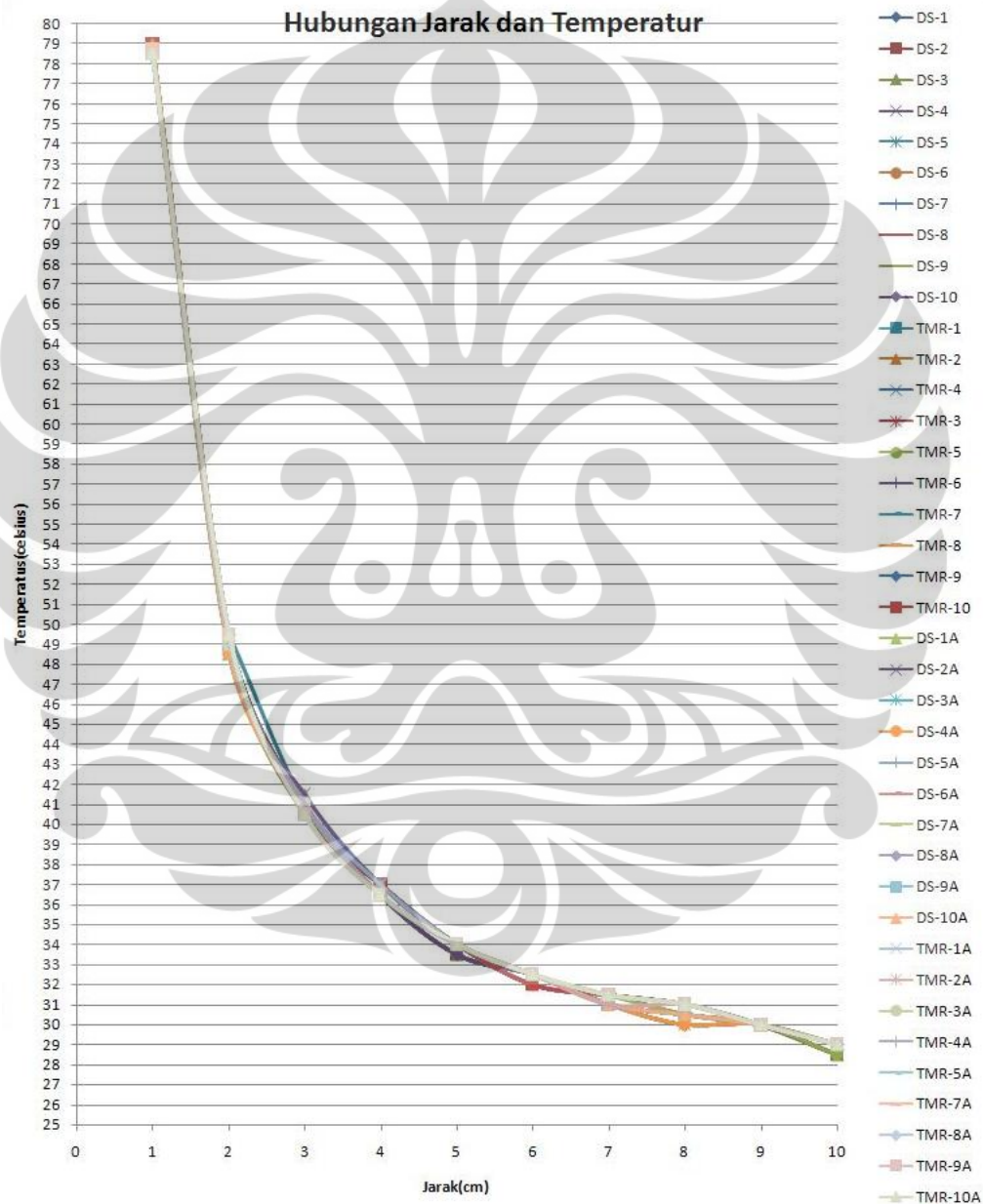
Jarak (cm)	Temperatur Terukur di DS18B20 (Derajat Celsius)									
	Pengukuran Ke-									
	1	2	3	4	5	6	7	8	9	10
1	79	78.5	78.5	78.5	78.5	78.5	78.5	78.5	78.5	78.5
2	48.5	48.5	48.5	48.5	49	49	49	49	49	49
3	41.5	41.5	41	41	41	41	41	41	41	41
4	37	37	37	36.5	36.5	36.5	36.5	36.5	36.5	36.5
5	34	34	34	34	34	34	34	34	34	34
6	32.5	32.5	32.5	32.5	32.5	32.5	32.5	32.5	32.5	32.5
7	31.5	31	31	31	31	31	31	31	31	31
8	30.5	30.5	30	30	30.5	30.5	30.5	30.5	30.5	30.5
9	30	30	30	30	30	30	30	30	30	30
10	29	29	29	29	29	29	29	29	29	29

Tabel 4.5 Data temperatur sensor termometer raksa (jarak antara sensor dan sumber panas semakin dijauhkan)

Jarak (cm)	Temperatur Terukur di Termometer Raksa (Derajat Celsius)									
	Pengukuran Ke-									
	1	2	3	4	5	6	7	8	9	10
1	78.5	78.5	78.5	78.5	78.5	79	79	78.5	78.5	78.5
2	49	49	49	49	49.5	49.5	49.5	49.5	49.5	49.5
3	41	41	41	41	40.5	40.5	40.5	40.5	40.5	40.5
4	37	36.5	36.5	36.5	36.5	36.5	36.5	36.5	36.5	36.5
5	34	34	34	34	34	34	34	34	34	34
6	32.5	32.5	32.5	32.5	32.5	32.5	32.5	32.5	32.5	32.5
7	31	31	31.5	31.5	31.5	31.5	31.5	31.5	31.5	31.5
8	31	31	31	31	31	31	31	31	31	31
9	30	30	30	30	30	30	30	30	30	30
10	29	29	29	29	29	29	29	29	29	29

Berdasarkan data-data yang terdapat pada Tabel 4.4 dan 4.5 grafik yang diperoleh juga tidak jauh berbeda dengan gambar grafik yang diperoleh

berdasarkan data-data yang terdapat pada Tabel 4.1 dan 4.2. Untuk mempermudah melihat karakteristik kedua sensor berdasarkan semua data yang diperoleh maka dibuat sebuah grafik yang dibuatkan berdasarkan semua data-data yang diperoleh baik data pada DS18B20 maupun data pada termometer raksa. Gambar 4.6 merupakan gambar sebuah grafik yang dibuatkan berdasarkan semua data-data yang diperoleh baik data pada DS18B20 maupun data pada termometer raksa.



Gambar 4.6 Gambar grafik berdasarkan semua data-data yang diperoleh baik data pada DS18B20 maupun data pada termometer raksa.

Keterangan series:

DS 1	:	Series data pengukuran ke-1 pada DS18B20 dari jarak 10-1 cm
DS 2	:	Series data pengukuran ke-2 pada DS18B20 dari jarak 10-1 cm
DS 3	:	Series data pengukuran ke-3 pada DS18B20 dari jarak 10-1 cm
DS 4	:	Series data pengukuran ke-4 pada DS18B20 dari jarak 10-1 cm
DS 5	:	Series data pengukuran ke-5 pada DS18B20 dari jarak 10-1 cm
DS 6	:	Series data pengukuran ke-6 pada DS18B20 dari jarak 10-1 cm
DS 7	:	Series data pengukuran ke-7 pada DS18B20 dari jarak 10-1 cm
DS 8	:	Series data pengukuran ke-8 pada DS18B20 dari jarak 10-1 cm
DS 9	:	Series data pengukuran ke-9 pada DS18B20 dari jarak 10-1 cm
DS 10	:	Series data pengukuran ke-10 pada DS18B20 dari jarak 10-1 cm
TMR 1	:	Series data pengukuran ke-1 pada termometer dari jarak 10-1 cm
TMR 2	:	Series data pengukuran ke-2 pada termometer dari jarak 10-1 cm
TMR 3	:	Series data pengukuran ke-3 pada termometer dari jarak 10-1 cm
TMR 4	:	Series data pengukuran ke-4 pada termometer dari jarak 10-1 cm
TMR 5	:	Series data pengukuran ke-5 pada termometer dari jarak 10-1 cm
TMR 6	:	Series data pengukuran ke-6 pada termometer dari jarak 10-1 cm
TMR 7	:	Series data pengukuran ke-7 pada termometer dari jarak 10-1 cm
TMR 8	:	Series data pengukuran ke-8 pada termometer dari jarak 10-1 cm
TMR 9	:	Series data pengukuran ke-9 pada termometer dari jarak 10-1 cm
TMR10	:	Series data pengukuran ke-10 pada termometer dari jarak 10-1 cm
DS-1A	:	Series data pengukuran ke-1 pada DS18B20 dari jarak 1-10 cm
DS-2A	:	Series data pengukuran ke-2 pada DS18B20 dari jarak 1-10 cm
DS-3A	:	Series data pengukuran ke-3 pada DS18B20 dari jarak 1-10 cm
DS-4A	:	Series data pengukuran ke-4 pada DS18B20 dari jarak 1-10 cm
DS-5A	:	Series data pengukuran ke-5 pada DS18B20 dari jarak 1-10 cm
DS-6A	:	Series data pengukuran ke-6 pada DS18B20 dari jarak 1-10 cm
DS-7A	:	Series data pengukuran ke-7 pada DS18B20 dari jarak 1-10 cm
DS-8A	:	Series data pengukuran ke-8 pada DS18B20 dari jarak 1-10 cm
DS-9A	:	Series data pengukuran ke-9 pada DS18B20 dari jarak 1-10 cm
DS-10A	:	Series data pengukuran ke-10 pada DS18B20 dari jarak 1-10 cm
TMR-1A	:	Series data pengukuran ke-1 pada termometer dari jarak 1-10 cm
TMR-2A	:	Series data pengukuran ke-2 pada termometer dari jarak 1-10 cm
TMR-3A	:	Series data pengukuran ke-3 pada termometer dari jarak 1-10 cm
TMR-4A	:	Series data pengukuran ke-4 pada termometer dari jarak 1-10 cm
TMR-5A	:	Series data pengukuran ke-5 pada termometer dari jarak 1-10 cm
TMR-6A	:	Series data pengukuran ke-6 pada termometer dari jarak 1-10 cm
TMR-7A	:	Series data pengukuran ke-7 pada termometer dari jarak 1-10 cm
TMR-8A	:	Series data pengukuran ke-8 pada termometer dari jarak 1-10 cm
TMR-9A	:	Series data pengukuran ke-9 pada termometer dari jarak 1-10 cm
TMR-10A	:	Series data pengukuran ke-10 pada termometer dari jarak 1-10 cm

4.1.4. Hasil dan Analisis Data dan Grafik Distribusi Temperatur

Untuk mendapatkan data dan grafik distribusi temperatur maka sebelumnya kita harus mendapatkan data seluruh sensor. Data temperatur seluruh sensor dapat ditampilkan pada GUI yang tampilannya telah dijelaskan di bab 3. Tampilan GUI data seluruh sensor ini merupakan tampilan GUI yang harus dihubungkan ke mikrokontroler H8/3069F melalui serial baik saat mengkonversi temperatur satu per satu atau seluruh sensor dan saat mengambil data temperatur sensor DS18B20.



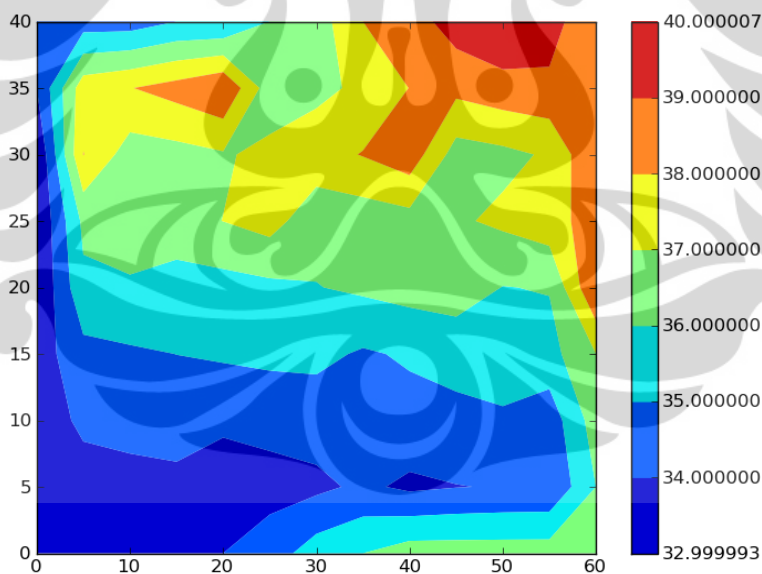
Gambar 4.7 Tampilan GUI Python saat menampilkan nilai temperatur yang terbaca 36 sensor DS18B20, temperatur tertinggi, dan temperatur terendah

Setelah mendapatkan seluruh data temperatur, maka kita dapat memasukkannya ke dalam program *finite difference* yang telah dibuat, yang telah sesuai dengan 77 persamaan turunan differensial. Berdasarkan 77 persamaan turunan differensial tersebut dapat kita rubah menjadi suatu perkalian matrik $Aw = b$. Matrik A dan b dapat diinisialisasi Berdasarkan 77 persamaan turunan differensial tersebut, maka matrik w dapat ditentukan.

Setelah melakukan penginisialisasian, maka didapatkan matrik A sebagai berikut:

$$\begin{bmatrix} 4. -1. 0...., 0. 0. 0. \\ -1. 4. -1...., 0. 0. 0. \\ 0. -1. 4...., 0. 0. 0. \\ \dots \\ 0. 0. 0...., 4. -1. 0. \\ 0. 0. 0...., -1. 4. -1. \\ 0. 0. 0...., 0. -1. 4. \end{bmatrix}$$

Berdasarkan syarat batas yang ditentukan berdasarkan data temperatur yang terbaca oleh sensor maka matrik b dan w dapat ditentukan. Lalu dengan menekan tombol *Show Graphic* maka akan muncul sebuah grafik kontur yang menggambarkan temperatur yang terbaca oleh sensor dan hasil distribusi yang telah diolah oleh algoritma finite difference. Gambar 4.8 merupakan Gambar grafik kontur hasil pengolahan berdasarkan data yang tertera pada Gambar 4.7:

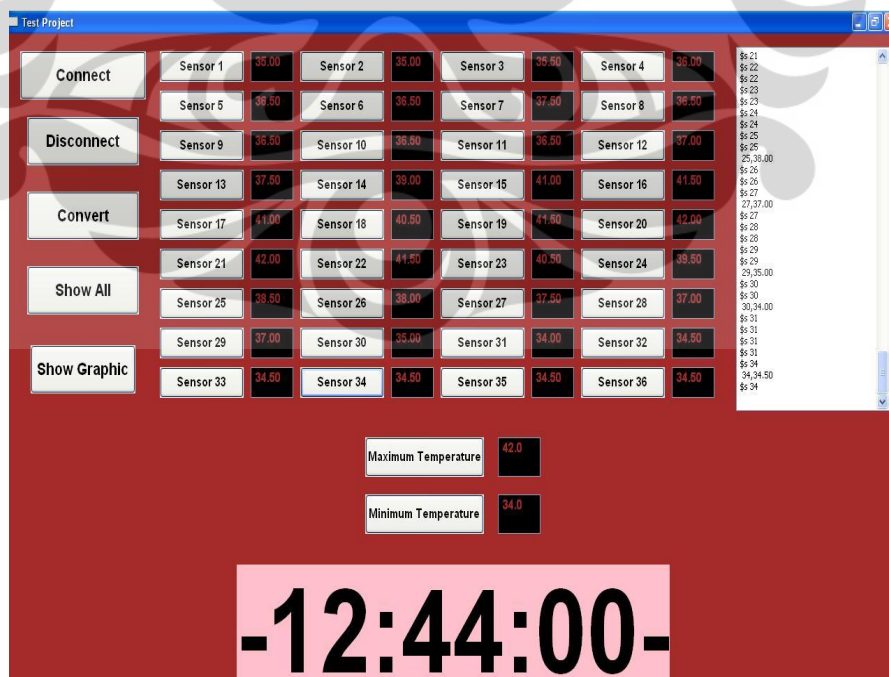


Gambar 4.8 Gambar grafik kontur hasil pengolahan berdasarkan data yang tertera pada Gambar 4.7

Setelah didapatkan grafik konturnya peneliti melakukan perbandingan data temperatur yang terbaca oleh grafik kontur dengan yang terukur secara manual.

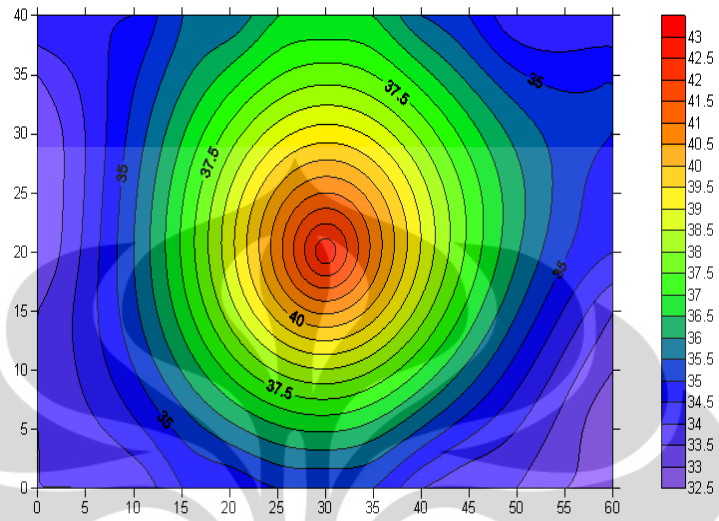
Saat sumber panas diletakan dipojokan incubator temperatur yang terukur dengan yang terbaca oleh grafik kontur hasilnya cukup sesuai, namun saat sumber diletakan ditengah incubator hasilnya kurang karena temperatur yang terbaca dan terukur datanya sangat berbeda. Olehsebab itu peneliti mencoba mencari metode lain untuk mendapatkan nilai distribusi berdasarkan data yang terukur oleh ke-36 sensor DS18B20. Maka peneliti mencoba mempelajari pengolahan data menggunakan metode interpolasi.

Interpolasi adalah suatu metode atau fungsi matematika yang menduga nilai pada lokasi-lokasi yang datanya tidak tersedia. Interpolasi spasial mengasumsikan bahwa atribut data bersifat kontinu di dalam ruang (*space*) dan atribut ini saling berhubungan (*dependence*) secara spasial. Kedua asumsi tersebut mengindikasikan bahwa pendugaan atribut data dapat dilakukan berdasarkan lokasi-lokasi di sekitarnya dan nilai pada titik-titik yang berhubungan. Ada beberapa metode interpolasi data yang dikenal, antara lain metode krigging dan invers distance. Metode krigging pada dasarnya adalah metode rata-ran terbobot dari setiap nilai contoh lokasi. Software surfer merupakan salah satu software yang salah satu pengolahan datanya menggunakan interpolasi krigging. Maka pada penelitian ini kami mencoba menjadikan keluaran yang terbaca oleh ke-36 sensor DS18B20 menjadi masukan untuk software surfer.



Gambar 4.9 Tampilan GUI Python saat menampilkan nilai temperatur yang terbaca 36 sensor DS18B20, temperatur tertinggi, dan temperatur terendah

Dan menghasilkan grafik kontur sebagai berikut:



Gambar 4.10 Gambar grafik kontur menggunakan software surfer

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah dilakukan penelitian ini baik dalam perancangan alat maupun pembuatan *software*, maka dapat disimpulkan bahwa:

1. Metode tomografi dapat diterapkan pada penentuan sumber panas dengan cara meletakkan sejumlah sensor di tepi sisi inkubator yang berfungsi sebagai syarat batas bagi perhitungan numerik
2. Sensor DS18B20 dapat digunakan sebagai sensor untuk pengukuran pada beberapa titik hanya dalam satu pin mikrokontroler
3. Dengan menggunakan metode numerik *finite difference* distribusi yang diperoleh hasilnya hanya cukup sesuai saat sumber panas terdapat pada tepi permukaan saja
4. Dengan menggunakan metode interpolasi kriging hasil distribusi yang diperoleh cukup sesuai dengan meletakkan sensor tidak di tepi saja
5. GUI Python dan database Python dapat menampilkan dan menyimpan data temperatur grafik distribusi temperatur, dan letak sumber panas

5.2 Saran

Penelitian lebih lanjut untuk menambah hasil-hasil yang telah dilakukan pada penelitian ini yang dapat dilakukan dengan :

1. Melakukan pengukuran pada bidang 3 dimensi
2. Penerapan *software* perancangan alat pada OS yang *open source*, seperti Linux.

DAFTAR ACUAN

- [1] <http://physicsnext.wordpress.com/2009/04/18/tomografi/> diakses pada tanggal 5 Agustus 2009 pukul 14.42 WIB
- [2] <http://www.fisikanet.lipi.go.id/utama.cgi?cetakarartikel&1167182875> diakses pada tanggal 5 Agustus 2009 pukul 14.51 WIB
- [3] <http://www.fisikanet.lipi.go.id/utama.cgi?cetakarartikel&1076033084> diakses pada tanggal 5 Agustus 2009 pukul 15.17 WIB
- [4] <http://www.erakomputer.com/node/42> diakses pada tanggal 5 Agustus 2009 pukul 15.29 WIB
- [5] V. Kolehmainen, J.P. Kaipio, and H.R.B. Orlande. 2007. “*a Reconstruction of thermal conductivity and heat capacity using a tomographic approach*”, Department of Mechanical Engineering, Universidade Federal de Rio de Janeiro, Rio de Janeiro, Brazil.
- [6] David Halliday and Robert Resnick. 1966. *Physics Combined Edition*. New York: John Willey & Sons, Inc.
- [7] Dalas semiconductor “*DS18B20 Programmable Resolution 1 wire Digital Thermometer*”, diakses dari <http://pdfserv.maxim-ic.com/en/ds/DS18B20.pdf> pada tanggal 30 september 2009 pukul 20.00 WIB
- [8] Renesas Solutions Corp. 2005. *H8/3068F-ZTAT™ Hardware Manual*. Renesas Technology Corp., Japan
- [9] Putra, Muchtar Suhari., et.al., 2008, *Sistem kontrol inkubator melalui jaringan LAN menggunakan mikrokontroller H8/3069F 16-bit*, Annual Meeting on Testing and Quality, Puspitek LIPI Serpong
- [10] Backer, A, 2002. *Computational Physics Education with Python*. Institut fur Theoretische Physik, TU Dresden, D-01062 Dresden. Germany.
- [11] Norton, Peter, Alex Samuel and friends, D.A, 2005. *Beginning Python*. Wiley Publishing, Inc. Indiana.
- [12] _____, 2009, [Overview of the MySQL Database Management System](http://dev.mysql.com/doc/refman/5.0/en/what-is-mysql.html) diakses dari <http://dev.mysql.com/doc/refman/5.0/en/what-is-mysql.html> pada tanggal 4 Oktober 2009.
- [13] Tim H8, 2007, *Modul Praktikum Renesas H8/3069F*, Departemen Fisika

FMIPA UI, Depok.

[14] Sukirno Santoso. 2005. “Karakteristik Sensor”. Depok : Departemen Fisika, FMIPA UI.

[15] Suparno Supriyanto, “Komputasi untuk Sains dan Teknik”, Edisi ke-III, Departemen Fisika, FMIPA UI, hlm 144-152, 2009



LAMPIRAN

No	X (cm)	Y (cm)	Suhu Kontur (C)	Suhu Terukur (C)	Kesalahan Relatif(%)
1	0	0	26.5	26	1.89
2	0	5	26.5	26	1.89
3	0	10	26.5	26	1.89
4	0	15	26.5	26	1.89
5	0	20	26.5	26	1.89
6	0	25	26.5	26	1.89
7	0	30	26.5	26	1.89
8	0	35	26.5	26	1.89
9	0	40	26.5	26	1.89
10	5	0	26.5	26	1.89
11	5	5	26.5	26	1.89
12	5	10	26.5	26	1.89
13	5	15	26.5	26	1.89
14	5	20	26.5	26	1.89
15	5	25	26.5	26	1.89
16	5	30	26.5	26	1.89
17	5	35	26.5	26	1.89
18	5	40	26.5	26	1.89
19	10	0	26.5	26	1.89
20	10	5	26.5	26	1.89
21	10	10	26.5	26	1.89
22	10	15	26.5	26	1.89
23	10	20	26.5	26	1.89
24	10	25	26.5	26	1.89
25	10	30	26.5	26	1.89
26	10	35	26.5	26	1.89
27	10	40	26.5	26	1.89
28	15	0	26.5	26	1.89
29	15	5	26.5	26	1.89
30	15	10	26.5	26	1.89
31	15	15	26.5	26	1.89
32	15	20	26.5	26	1.89
33	15	25	26.5	26	1.89
34	15	30	26.5	26	1.89

35	15	35	26.5	26	1.89
36	15	40	26.5	26	1.89
37	20	0	26.5	26	1.89
38	20	5	26.5	26	1.89
39	20	10	26.5	26	1.89
40	20	15	26.5	26	1.89
41	20	20	26.5	26	1.89
42	20	25	26.5	26	1.89
43	20	30	26.5	26	1.89
44	20	35	26.5	26	1.89
45	20	40	26.5	26	1.89
46	25	0	26.5	26	1.89
47	25	5	26.5	26	1.89
48	25	10	26.5	26	1.89
49	25	15	26.5	26	1.89
50	25	20	26.5	26	1.89
51	25	25	26.5	26	1.89
52	25	30	26.5	26	1.89
53	25	35	26.5	26	1.89
54	25	40	26.5	26	1.89
55	30	0	26.5	26	1.89
56	30	5	26.5	26	1.89
57	30	10	26.5	26	1.89
58	30	15	26.5	26	1.89
59	30	20	26.5	26	1.89
60	30	25	26.5	26	1.89
61	30	30	26.5	26	1.89
62	30	35	26.5	26	1.89
63	30	40	26.5	26	1.89
64	35	0	26.5	26	1.89
65	35	5	26.5	26	1.89
66	35	10	26.5	26	1.89
67	35	15	26.5	26	1.89
68	35	20	26.5	26	1.89
69	35	25	26.5	26	1.89
70	35	30	26.5	26	1.89
71	35	35	26.5	26	1.89
72	35	40	26.5	26	1.89
73	40	0	26.5	26	1.89
74	40	5	26.5	26	1.89
75	40	10	26.5	26	1.89

76	40	15	26.5	26	1.89
77	40	20	26.5	26	1.89
78	40	25	26.5	26	1.89
79	40	30	26.5	26	1.89
80	40	35	26.5	26	1.89
81	40	40	26.5	26	1.89
82	45	0	26.5	26	1.89
83	45	5	26.5	26	1.89
84	45	10	26.5	26	1.89
85	45	15	26.5	26	1.89
86	45	20	26.5	26	1.89
87	45	25	26.5	26	1.89
88	45	30	26.5	26	1.89
89	45	35	26.5	26	1.89
90	45	40	26.5	26	1.89
91	50	0	26.5	26	1.89
92	50	5	26.5	26	1.89
93	50	10	26.5	26	1.89
94	50	15	26.5	26	1.89
95	50	20	26.5	26	1.89
96	50	25	26.5	26	1.89
97	50	30	26.5	26	1.89
98	50	35	26.5	26	1.89
99	50	40	26.5	26	1.89
100	55	0	26.5	26	1.89
101	55	5	26.5	26	1.89
102	55	10	26.5	26	1.89
103	55	15	26.5	26	1.89
104	55	20	26.5	26	1.89
105	55	25	26.5	26	1.89
106	55	30	26.5	26	1.89
107	55	35	26.5	26	1.89
108	55	40	26.5	26	1.89
109	60	0	26.5	26	1.89
110	60	5	26.5	26	1.89
111	60	10	26.5	26	1.89
112	60	15	26.5	26	1.89
113	60	20	26.5	26	1.89
114	60	25	26.5	26	1.89
115	60	30	26.5	26	1.89
116	60	35	26.5	26	1.89

117	60	40	26.5	26	1.89
-----	----	----	------	----	------

No	X(cm)	Y(cm)	Suhu(Kontur)	Suhu(Ukur)	Kesalahan Relatif(%)
1	0	0	33.5	-	-
2	0	5	33.5	-	-
3	0	10	33.5	-	-
4	0	15	33.5	-	-
5	0	20	33	-	-
6	0	25	33	-	-
7	0	30	33	-	-
8	0	35	33.5	-	-
9	0	40	34.5	-	-
10	5	0	33.5	-	-
11	5	5	33.5	33	1.49
12	5	10	33.5	33	1.49
13	5	15	33.5	33	1.49
14	5	20	33.5	33	1.49
15	5	25	33.5	33	1.49
16	5	30	33.5	33	1.49
17	5	35	33.5	33	1.49
18	5	40	34.5	-	-
19	10	0	33.5	-	-
20	10	5	34	33	2.94
21	10	10	34.5	34	1.45
22	10	15	34.5	34	1.45
23	10	20	34.5	34	1.45
24	10	25	34.5	34	1.45
25	10	30	34.5	34	1.45
26	10	35	34.5	34	1.45
27	10	40	34.5	34	1.45
28	15	0	34.5	34	1.45
29	15	5	35	34	2.86
30	15	10	36	35	2.78
31	15	15	36	35	2.78
32	15	20	36.5	36	1.37
33	15	25	36.5	36	1.37
34	15	30	36.5	36	1.37
35	15	35	36	36	0
36	15	40	35.5	-	-

37	20	0	34.5	-	-
38	20	5	35.5	35	1.41
39	20	10	36.5	36	1.37
40	20	15	37	36	2.7
41	20	20	38	37	2.63
42	20	25	38	37	2.63
43	20	30	37.5	37	1.33
44	20	35	35.5	35	1.41
45	20	40	35.5	35	1.41
46	25	0	35	35	0
47	25	5	36	35	2.78
48	25	10	36.5	36	1.37
49	25	15	38.5	37	3.9
50	25	20	39	38	2.56
51	25	25	38.5	38	1.3
52	25	30	37.5	38	-1.3
53	25	35	36	36	0
54	25	40	36.5	36	1.37
55	30	0	35	36	-2.9
56	30	5	36	35	2.78
57	30	10	38	37	2.63
58	30	15	40.5	40	1.23
59	30	20	43	43	0
60	30	25	39.5	40	-1.3
61	30	30	38	38	0
62	30	35	37	37	0
63	30	40	36.5	36	1.37
64	35	0	35	36	-2.9
65	35	5	-	-	-
66	35	10	-	-	-
67	35	15	-	-	-
68	35	20	-	-	-
69	35	25	-	-	-
70	35	30	-	-	-
71	35	35	-	-	-
72	35	40	36.5	-	-
73	40	0	34	-	-
74	40	5	-	-	-
75	40	10	-	-	-
76	40	15	-	-	-
77	40	20	-	-	-

78	40	25	-	-	-
79	40	30	-	-	-
80	40	35	-	-	-
81	40	40	35.5	-	-
82	45	0	33.5	-	-
83	45	5	-	-	-
84	45	10	-	-	-
85	45	15	-	-	-
86	45	20	-	-	-
87	45	25	-	-	-
88	45	30	-	-	-
89	45	35	-	-	-
90	45	40	35	-	-
91	50	0	33.5	-	-
92	50	5	-	-	-
93	50	10	-	-	-
94	50	15	-	-	-
95	50	20	-	-	-
96	50	25	-	-	-
97	50	30	-	-	-
98	50	35	-	-	-
99	50	40	34	-	-
100	55	0	33	-	-
101	55	5	-	-	-
102	55	10	-	-	-
103	55	15	-	-	-
104	55	20	-	-	-
105	55	25	-	-	-
106	55	30	-	-	-
107	55	35	-	-	-
108	55	40	34	-	-
109	60	0	33	-	-
110	60	5	32.5	-	-
111	60	10	33	-	-
112	60	15	33.5	-	-
113	60	20	34.5	-	-
114	60	25	34.5	-	-
115	60	30	35	-	-
116	60	35	35	-	-
117	60	40	34	-	-

PROGRAM

```
#Main program GUI#

from __future__ import division

import wx

import wx.grid

import random

from random import randint as rand

import time

import serial, threading

from numpy import *

from modul_tiwu import *

import wx

from matplotlib.patches import Patch

from matplotlib import pyplot as P

from matplotlib.colors import LogNorm

from pylab import *

from numpy import *

import db_tabel as tbl

SERIALRX = wx.NewEventType()

# bind to serial data receive events

EVT_SERIALRX = wx.PyEventBinder(SERIALRX, 0)

RETURN = chr(13)

CARRIAGE = chr(10)

class SerialRxEvent(wx.PyCommandEvent):

    eventType = SERIALRX
```

```

def __init__(self, windowID, data):
    wx.PyCommandEvent.__init__(self, self.eventType, windowID)
    self.data = data

    def Clone(self):
        self.__class__(self.GetId(), self.data)

class FrmMain(wx.Frame):
    def __init__(self, parent, id):
        wx.Frame.__init__(self, parent, id, title='Test
Project')

        color = wx.SystemSettings.GetColour(wx.SYS_COLOUR_BTNFACE)
        self.SetBackgroundColour('Pink')

        pnl = wx.Panel(self, -1)
        pnl.SetBackgroundColour('Brown')

        btn1 = wx.Button(pnl, -1, 'Connect', size = (145,55))
        btn2 = wx.Button(pnl, -1, 'Disconnect', size = (150,55))
        btn3 = wx.Button(pnl, -1, 'Convert', size = (150,55))
        btn4 = wx.Button(pnl, -1, 'Show All', size = (150,55))
        btn5 = wx.Button(pnl, -1, 'Show Graphic', size = (150,55))

        abtn = wx.BoxSizer(wx.VERTICAL)

        abtn.Add(btn1, 0, wx.EXPAND | wx.ALL, 5)
        abtn.Add(btn2, 0, wx.EXPAND | wx.ALL, 15)
        abtn.Add(btn3, 0, wx.EXPAND | wx.ALL, 15)
        abtn.Add(btn4, 0, wx.EXPAND | wx.ALL, 15)
        abtn.Add(btn5, 0, wx.EXPAND | wx.ALL, 20)

```

```

self.data = wx.TextCtrl(pnl, -1, '0')

self.data.Hide() #disembunyikan

self.btn8-43 =wx.Button(pnl, -1, 'Sensor 1-36', size = (120,35))
self.data1-36 = wx.TextCtrl(pnl, -1, '', size = (61,35))

bbtn = wx.FlexGridSizer(0,8,0,0) #(kolom,baris,gap)
self.Bind(wx.EVT_BUTTON, self.OnBtn1, btn1)
self.Bind(wx.EVT_BUTTON, self.OnBtn2, btn2)
self.Bind(wx.EVT_BUTTON, self.OnBtn3, btn3)
self.Bind(wx.EVT_BUTTON, self.OnBtn4, btn4)
self.Bind(wx.EVT_BUTTON, self.OnBtn5, btn5)

self.log = wx.TextCtrl(pnl, -1, '', style = wx.TE_MULTILINE |
wx.TE_PROCESS_ENTER,size =(220,60))

self.btn44 =wx.Button(pnl, -1, 'Maximum Temperature'
self.data37 = wx.TextCtrl(pnl, -1, '', size = (61,35))
self.btn45 =wx.Button(pnl, -1, 'Minimum Temperature'self.data38 =
wx.TextCtrl(pnl, -1, '', size = (61,35))

cbtn = wx.FlexGridSizer(0,2,0,0) #(kolom,baris,gap)
cbtn.Add(self.btn44, 0, wx.EXPAND | wx.ALL, 10)
cbtn.Add(self.data37, 0, wx.EXPAND | wx.ALL, 10)
cbtn.Add(self.btn45, 0, wx.EXPAND | wx.ALL, 10)
cbtn.Add(self.data38, 0, wx.EXPAND | wx.ALL, 10)

self.jam = wx.StaticText(self, -1, "hh:mm:ss")
self.jam.SetFont(h3)

```

```
self.Timer = wx.Timer(self)

self.Bind(wx.EVT_TIMER, self.OnTimerna, self.Timer)

self.Timer.Start(1000)

bord1 = wx.BoxSizer(wx.VERTICAL)
bord1.Add(cbtn, 0, wx.ALIGN_CENTER_HORIZONTAL | wx.ALL, 5)
bord1.Add(self.jam, 1, wx.ALIGN_CENTER_HORIZONTAL | wx.ALL, 20)

bord2 = wx.BoxSizer(wx.HORIZONTAL)
bord2.Add(abtn, 0, wx.EXPAND | wx.ALL, 5)
bord2.Add(bbtn, 1, wx.EXPAND | wx.ALL, 5)
bord2.Add(self.log, 0, wx.EXPAND | wx.ALL, 5)

bord3 = wx.BoxSizer(wx.VERTICAL)
bord3.Add(bord2, 0, wx.EXPAND | wx.ALL, 5)
bord3.Add(bord1, 0, wx.EXPAND | wx.ALL, 5)
xbord = wx.BoxSizer()
xbord.Add(bord3, 1, wx.EXPAND | wx.ALL, 5)
pnl.SetSizer(xbord)

box = wx.BoxSizer(wx.VERTICAL)
box.Add(pnl, 1, wx.EXPAND | wx.ALL, 0)

border = wx.BoxSizer()
border.Add(box, 1, wx.EXPAND | wx.ALL, 0)

self.SetSizer(border)

self.Layout()
```



```
#Initial value
self.ser = serial.Serial()
self.ser.timeout = 0.5
self.thread = None
self.alive = threading.Event()

self.timer = wx.Timer(self)
self.Bind(wx.EVT_TIMER, self.OnTimer, self.timer)

#Event Serial
self.Bind(EVT_SERIALRX, self.OnSerialRead)

def Waktuna(self):
    t = time.localtime(time.time())
    st = time.strftime("%I:%M:%S-",t)
self.jam.SetLabel(st)

def OnTimerna(self, event):
    self.Waktuna()

def Waktu(self):
    self.log.WriteText('Membaca semua sensor...\n')
    self.ConvertSensor()
    self.Delay(0.05)

def OnTimer(self, event):
    pass
```

```
def StartThread(self):  
    """Start the receiver thread"""  
    self.thread = threading.Thread(target=self.ComPortThread)  
    self.thread.setDaemon(1)  
    self.alive.set()  
    self.thread.start()  
  
def StopThread(self):  
    """Stop the receiver thread, wait until it's  
finished."""  
    if self.thread is not None:  
        self.alive.clear  
        self.thread.join  
        self.thread = None  
  
def ComPortThread(self):  
    while self.alive.isSet():  
        text = self.ser.read(1)  
        n = self.ser.inWaiting()  
        if n:  
            text = text + self.ser.read(n)  
            text = text.replace('\r', '')  
            event = SerialRxEvent(self.GetId(), text)  
  
self.GetEventHandler().AddPendingEvent(event)  
  
def OnSerialRead(self, event):  
    self.data.WriteText(text)  
  
self.data_asli = self.data.GetValue()  
self.data.SetValue('')
```

```

temp = self.data_asli.replace(chr(13), '')
temp = temp.replace(chr(10), '')
temp = temp.split(' ')
try:
    data_mentah = temp[1]
    cek = data_mentah.split(',')[0]
    self.referensi = float(data_mentah.split(',')[1])
    if cek == '0-35':
        self.data_asli0-35 = data_mentah.split(',')[1]
        self.data1-36.SetValue('%s' % self.data_asli0-
35)
self.data_jadi1-36 = float(self.data_asli1-36)

self.sumbux =
[0,5,10,15,20,25,30,35,40,45,50,55,60,0,60,0,60,0,60,0,60,0,60,0,60,0,60,0,60,0,60,0,5,10,15,20,25,30,35,40,45,50,55,60]

self.sumbuy =
[0,0,0,0,0,0,0,0,0,0,0,0,0,5,5,10,10,15,15,20,20,25,25,30,30,35,35,40,40,40,40,40,40,40,40,40,40,40,40,40,40,40,40]

self.databasnya = [self.data_jadi1,self.data_jadi1,
self.data_jadi1, self.data_jadi2, self.data_jadi3,
self.data_jadi4, self.data_jadi5, self.data_jadi6,
self.data_jadi7, self.data_jadi9, self.data_jadi9,
self.data_jadi10, self.data_jadi10, self.data_jadi34,
self.data_jadi11, self.data_jadi34, self.data_jadi12,
self.data_jadi34, self.data_jadi13, self.data_jadi31,
self.data_jadi14, self.data_jadi31, self.data_jadi15,
self.data_jadi30, self.data_jadi16, self.data_jadi29,
self.data_jadi17, self.data_jadi28, self.data_jadi28,
self.data_jadi27, self.data_jadi26,
self.data_jadi25,self.data_jadi24,self.data_jadi23,self.data_jadi2
2,self.data_jadi21,self.data_jadi20,self.data_jadi19,self.data_jad
i18,self.data_jadi18]

for i in range(0,40):

```

```
tbl.InsertData('', '',str(self.sumbux[i]),str(self.sumbuy[i]),
str(self.databasesnya[i]))

Sensornya = [self.data_jadi1, self.data_jadi2, self.data_jadi3,
self.data_jadi4, self.data_jadi5, self.data_jadi6,
self.data_jadi7, self.data_jadi9, self.data_jadi10,
self.data_jadi11, self.data_jadi12, self.data_jadi13,
self.data_jadi14, self.data_jadi15, self.data_jadi16,
self.data_jadi17, self.data_jadi18, self.data_jadi19,
self.data_jadi20, self.data_jadi21, self.data_jadi22,
self.data_jadi23, self.data_jadi24, self.data_jadi25,
self.data_jadi26, self.data_jadi27, self.data_jadi28,
self.data_jadi29, self.data_jadi30, self.data_jadi31,
self.data_jadi34]

self.M = max(Sensornya)

self.data37.SetValue('%s' % self.M)

self.N = min(Sensornya)
self.data38.SetValue('%s' % self.N)

except IndexError:
    pass
    self.log.AppendText(text)

def OnBtn1(self, event):
    self.log.WriteText('Connect to serial com9...\n')

    try:
        self.ser = serial.Serial('com9')
        self.StartThread()
        self.timer.Start(250)

    except serial.SerialException:
```

```
warn = wx.MessageDialog(self, 'Tidak dapat membuka port...',
                        'TEST', wx.OK | wx.ICON_STOP)

warn.ShowModal()

warn.Destroy()

def OnBtn2(self, event):

    self.log.WriteText('Disconnect serial...\n')

    self.StopThread()

    self.ser.close()

def Delay(self, detik):

    time.sleep(detik)

def ConvertSensor(self):

    self.log.WriteText('Mengkonversi...\n')

    self.ser.write('$c')

    self.ser.write(RETURN)

def GetDataSensor(self, nomor):

    sensor = nomor - 1

    no_sensor = nomor

    self.ser.write('$s ' + str(sensor))

    self.ser.write(RETURN)

    self.Delay(0.05)

def OnBtn3(self, event):

    self.ConvertSensor()

def OnBtn4(self, event):

    self.log.WriteText('Membaca semua sensor...\n')

    self.ConvertSensor()

    self.Delay(0.05)
```

```

self.GetDataSensor(1-36)

def OnBtn5(self, event):

    x = array([0, 5, 10 , 15, 20 , 25, 30, 35, 40, 45, 50,
55,60])

    y = array([0, 5, 10 , 15, 20 , 25, 30, 35,40])

z = array([[self.data_jadil, self.data_jadil,
self.data_jadil,self.data_jadi2 , self.data_jadi3, self.data_jadi4
, self.data_jadi5, self.data_jadi6, self.data_jadi7,
self.data_jadi9, self.data_jadi9, self.data_jadi10,
self.data_jadi10], [self.data_jadi34, self.dis1, self.dis2 ,
self.dis3, self.dis4 , self.dis5, self.dis6, self.dis7, self.dis8,
self.dis9, self.dis10, self.dis11, self.data_jadi11])

X,Y = meshgrid(x,y)
contourf(X, Y, z)
ax = subplot(111)
axis([0, 60, 0, 40])
colorbar(1)
show()

def OnSensor1-36(self, event):
    self.GetDataSensor(1-36)

```

Program C untuk DS18B20

```

/*****
*****
File untuk mengukur suhu
Command line :
    $s [no sensor] -> untuk mengambil suhu sensor
    $c -> untuk memerintahkan sensor mengkonversi
suhu pada semua sensor

Contoh :
    $s 1      Tekan Enter

```

```

Program Status : Completed
*****
*****/

#include "ioh83069.h"
#include "uart.h"
#include "ds1820.h"
#include "delay.h"
#include "romcode.h"
#include "interrupt.h"
#include <stdlib.h>

//const unsigned char rom [ROM_LEN] =
{0x28,0xFA,0x6F,0x98,0x01,0x00,0x00,0x8B};
//unsigned char rom [ROM_LEN];
unsigned char sp [SCP_LEN], cmdln, nsensor;
char data, buf [10], cmdbuf[10];

//Deklarasi
//Fungsi untuk menangani interrupt handler
void ISR_RXI1 (void)
{
    data = RDR1;
    SSR1 &= ~(1 << RDRF);

    if (data != 13) {

        uart_ch(data);
        cmdbuf[cmdln] = data;
        cmdln++;

    } else {

        uart_strP("\r\n");
        cmdbuf[cmdln] = '\0';

        if (cmdbuf[0] != '$') {

            uart_strP("Invalid Command \r\n");

        } else { //akhir 0

            if (cmdbuf[1] == 's' || cmdbuf[1] ==
'S') { //Get nilai suhu sensor

                if (cmdln > 2) {

```

```

                                nsensor = (unsigned char)
atoi(&cmdbuf[2]);

    ds1820_match_rom(&rom[nsensor][0]);
                                ds1820_get_scratchpad(&sp[0]);

    ds1820_print_suhu(&buf[0],&sp[0]);
                                uart_str(&cmdbuf[2]);
                                uart_strP(",");
                                uart_str(&buf[0]);
                                uart_strP("\r\n");

                                } else {
                                    uart_strP("No : 0 - 34\r\n");
                                }

                                } else if (cmdbuf[1] == 'c' || cmdbuf[1]
== 'C') { //Start konversi suhu

                                ds1820_convert_all();
                                _delay_us(100); //tunggu sampe 100
us
                                uart_strP("Convert OK\r\n");

                                } else { //akhir 1
                                    uart_strP("Invalid Command \r\n");
                                }
                                }

                                cmdln = 0;
                                }
                                }

int main (void)
{

    uart_init();
    uart_strP("Inisialisasi Uart\r\n");
    ds1820_init();
    uart_strP("Inisialisasi Ds18B20\r\n");
    cmdln = 0;
    sei();
    while(1);
    return 0;
}

```