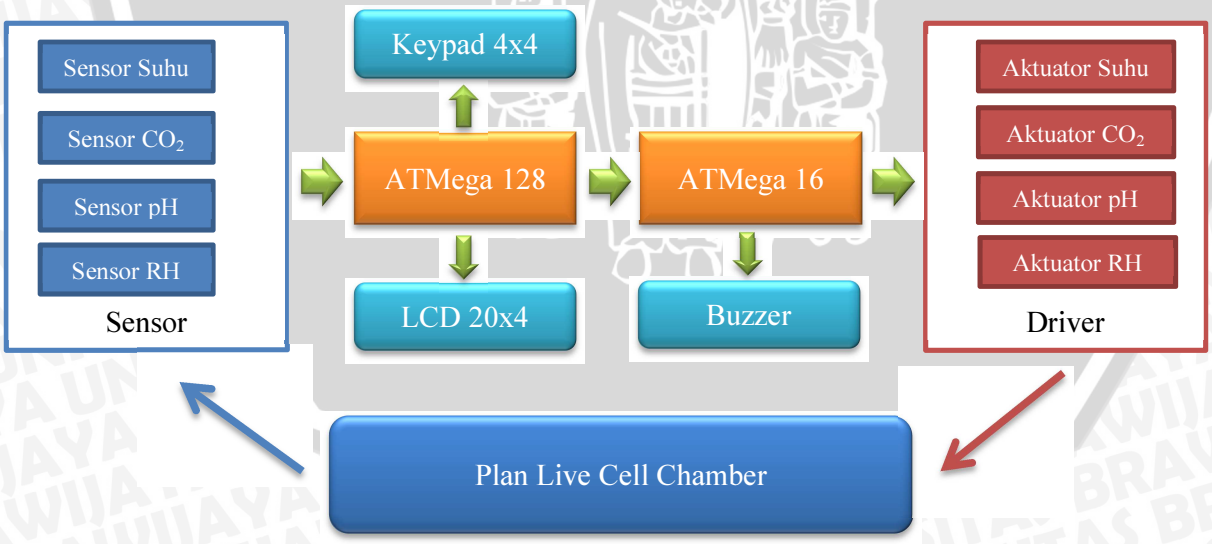
BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

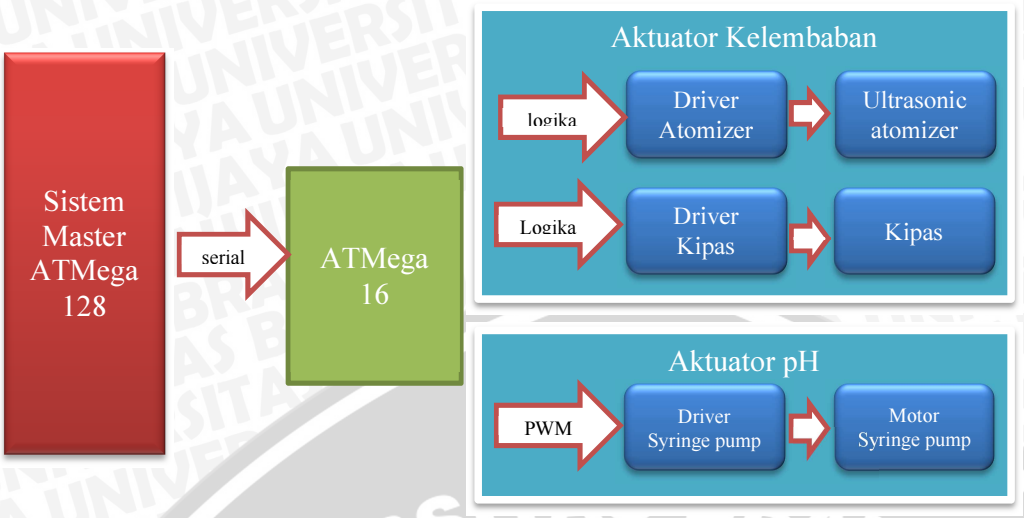
Bab ini menjelaskan tentang perancangan dan pembuatan *system Live cell Chamber* mulai dari diagram blok sistem, desain mekanik, perancangan perangkat keras, dan perancangan perangkat lunak. Perancangan dan pembuatan dilakukan secara bertahap dan sistematis, sehingga nantinya akan memudahkan dalam analisis sistem.

4.1 Diagram Blok Sistem

Sistem *live cell chamber* ini terdiri dari 2 buah mikrokontroler yang memiliki tugas tersendiri. Pada sistem mikrokontroler *master* yang memakai ATMega 128 digunakan untuk menangani sensor yang ada yaitu sensor suhu, pH, kelembaban dan sensor CO_2, menerima masukan dari keypad 4x4, serta menampilkan hasil pembacaan sensor pada LCD 20x4. Sedangkan untuk mikrokontroler *slave* digunakan untuk mengontrol semua aktuator yang ada dari perintah yang dikirimkan oleh mikrokontroler *master*. Selain itu terdapat *buzzer* yang digunakan sebagai peringatan kepada operator. Sistem *live cell chamber* secara keseluruhan ditunjukkan dalam Gambar 4.1. dan pada Gambar 4.2 menunjukkan diagram blok sistem aktuator pH dan kelembaban pada sistem *Live Cell Chamber.*



Gambar 4. 1 Diagram Blok Sistem *Live Cell Chambere*



Gambar 4.2 blok diagram aktuator pH dan kelembaban

Pada sistem aktuator pH dan kelembaban ini terdapat 2 jenis aktuator yang digunakan yaitu aktuator pH yang berupa *syringe pump* yang berisi cairan *buffer* pH sodium bikarbonat dan aktuator kelembaban yang berupa *ultrasonic atomizer*.

Pada aktuator pH digunakan *syringe pump* yang pada dasarnya terdiri dari unit kontrol, motor dc dan *syring* atau alat suntik. Alat ini bekerja dengan pertama memasang suntik 50 ml yang berisi cairan *buffer* kemudian mengunci di tempatnya. Saat dinyalakan sistem akan menunggu perintah dari mikrokontroler master ATMega 128 mengenai kapan sistem harus dinyalakan dan berapakah kecepatan yang harus diberikan untuk mendorong *Syringe* karena, kecepatan gerak motor yang dikontrol melalui PWM mempengaruhi berapa banyak volume yang diberikan dan mempengaruhi kenaikan pH yang diinginkan.

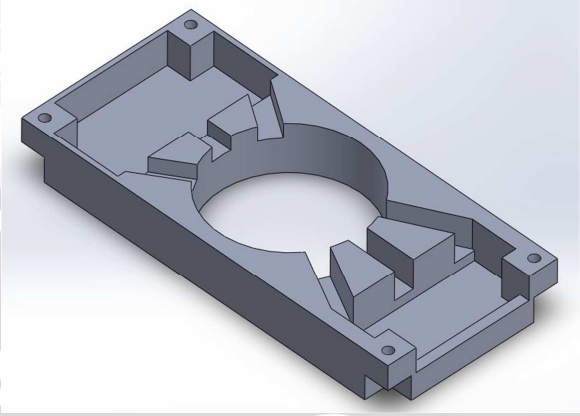
Pada *ultrasonic atomizer,* kita pertama harus menyiapkan air pada transduser *ultrasonik atomizer* hingga air memenui batas yang ditentukan. Kemudian setelah transduser dihubungkan ke driver dan catu daya, sistem akan menunggu perintah dari mikrokontroler ATMega 16 kapan sistem harus dinyalakan dan berapa waktu yang dibutuhkan sesuai dari perintah mikrokontroler ATmega 128 ke ATmega 16. Dan saat atomizer telah menyala dan menghasilkan embun, kipas bekerja untuk mengalirkan embun tersebut ke *chamber.* Selain dengan kipas, embun tersebut juga dapat dialirkan dengan gas CO_2 karena saluran gas sejalur dengan saluran kelembaban.

4.2 Perancangan Perangkat Keras

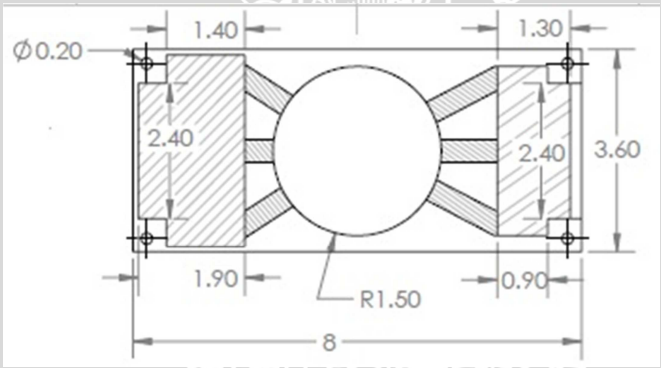
4.2.1 Perancangan Mekanik

Mekanik dari sistem ini akan terbagi dari 4 buah bagian yaitu bagian unit kontrol yang berisi sistem mikroprosesor ATMega 16, kipas, *syringe pump* dan driver *ultrasonic*

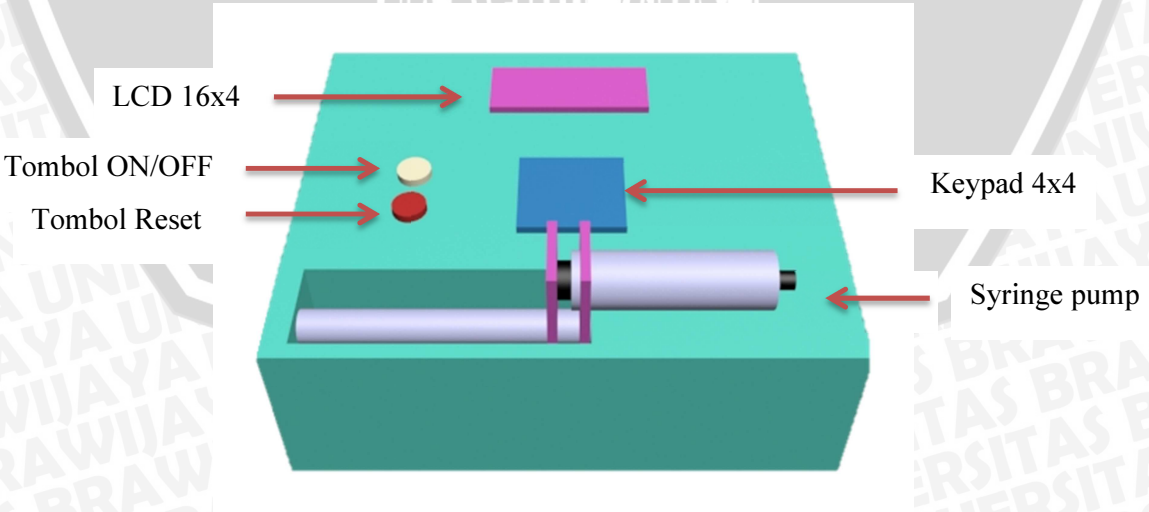
atomizer. Bagian tangki air yang berisi air dan transduser dari *ultrasonic atomizer*. Bagian *Syringe pump* yang berisi driver motor, motor DC, serta pendukung sistem *syringe pump*. Serta bagian power supply yang berisi Swiching Power supply unit. Selain itu juga sistem *Chamber* tempat sel dijaga kondisinya.



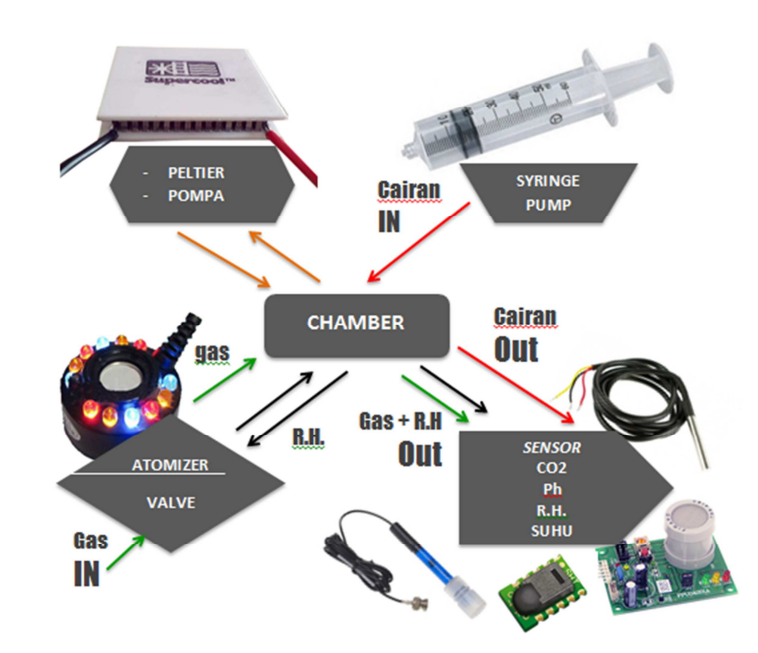
Gambar 4.3 Design Mekanik *Chamber* Sel



Gambar 4.4 *Chamber* sel tampak atas



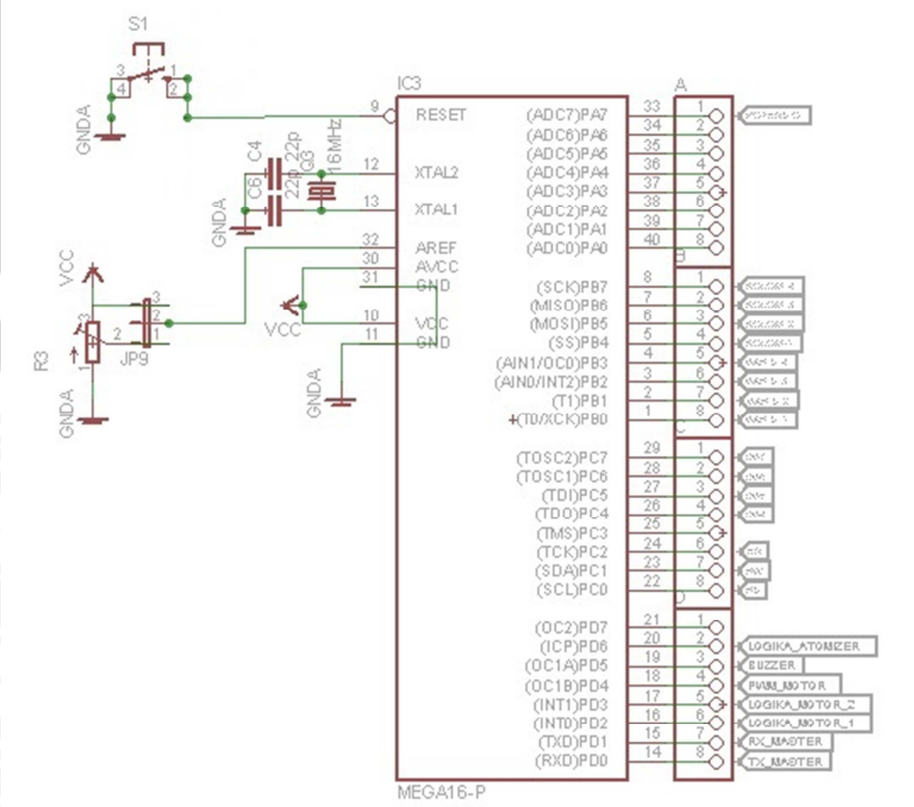
Gambar 4.5 Design kotak elektrik



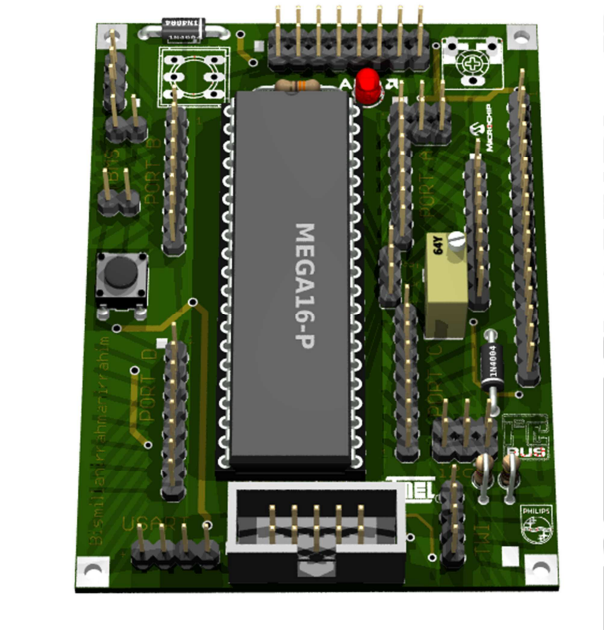
Gambar 4.6 Diagram sambungan sistem Live Cell Chamber

4.2.2 Perancangan Rangkaian Elektrik

4.2.2.1 Perancangan Rangkaian Sistem Minimum ATMega 16



Gambar 4.7 Perancangan Sistem minimum ATMega 16



Gambar 4.8 Design 3D sistem minimum ATMega 16

Mikrokontroler Atmega16 yang digunakan memiliki 4 *port* 8 *bits bidirectional input output* yang dapat diprogram, yaitu *Port* A, *Port* B, *Port* C dan *Port* D. Pengaturan dan penggunaan pin mikrokontroler dalam perancangan alat ini adalah:

1. Port A
    - Pin A.0
    - Pin A.1
    - Pin A.2 dihubungkan ke buzzer
    - Pin A.3 (terpakai)
    - Pin A.4 dihubungkan ke logika kipas atomizer
    - Pin A.5 dihubungkan ke logika driver *Ultrasonic Atomizer*
    - Pin A.6 (terpakai)
    - Pin A.7 (terpakai)
2. Port B
    - Pin B.0
    - Pin B.1
    - Pin B.2
    - Pin B.3
    - Pin B.4
    - Pin B.5
    - Pin B.6

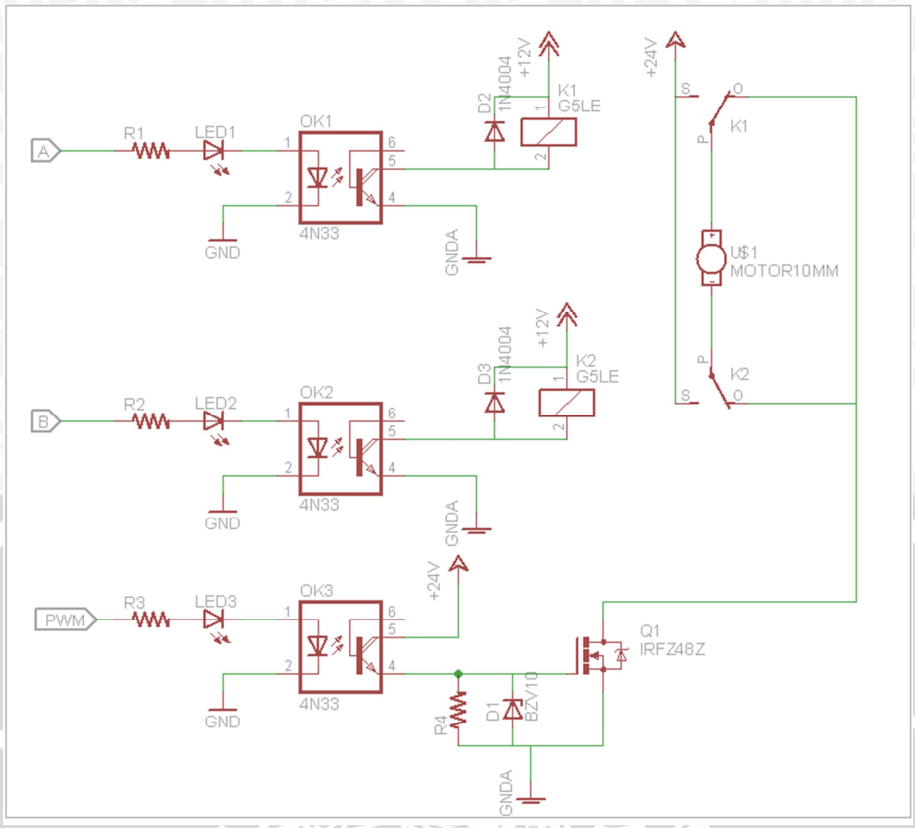
- Pin B.7

3. Port C
    - Pin C.0
    - Pin C.1
    - Pin C.2
    - Pin C.3
    - Pin C.4
    - Pin C.5
    - Pin C.6
    - Pin C.7
4. Port D
    - Pin D.0 dihubungkan ke pin TX mikrokontroler Master
    - Pin D.1 dihubungkan ke pin RX mikrokontroler Master
    - Pin D.2 dihubungkan ke logika driver motor *syringe pump*
    - Pin D.3 dihubungkan ke logika driver motor *syringe pump*
    - Pin D.4 dihubungkan ke PWM driver motor *syringe pump*
    - Pin D.5 (terpakai)
    - Pin D. 6 (terpakai)
    - Pin D.7 (terpakai)

4.2.2.2 Perancangan Driver Motor DC *Syringe pump*

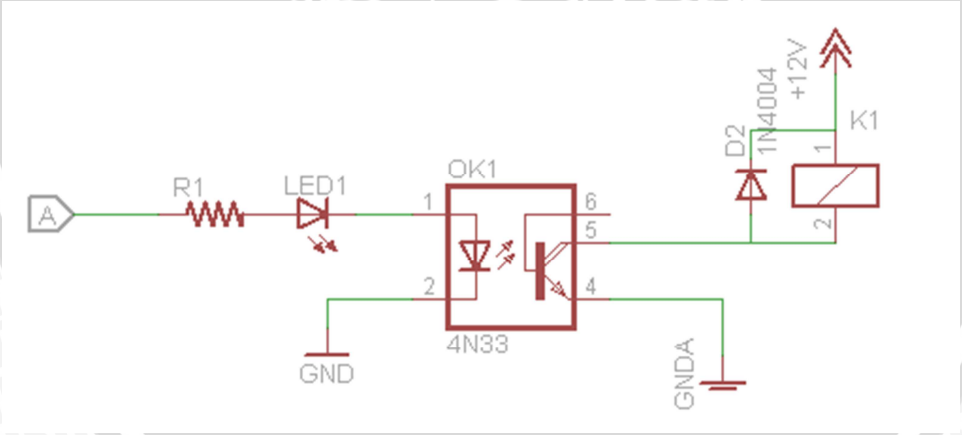
Pada sistem *syringe pump* ini mengggunakan driver relay untuk mengontrol pergerakan motor yang mengatur pendorong *syringe* atau suntik. Driver relay ini terdiri atas 2 buah relay yang bekerja berlawanan yang menyebabkan arah motor bisa berubah ke arah maju atau mundur sesuai kombinasi input relay yang diberikan. Dan untuk mengontro kecepatan dari motor digunakan control PWM dengan menggunakan E-MOSFET Kanal N. dan sebagai pengaman dengan rangkaian di depannya dipasang isolasi fisik dengan ditambahkan optocoupler.

Rangkaian keseluruhan dari driver pengatur kecepatan motor dan arah pergerakan motor *syringr pump* dapat dilihat pada gambar 4.9



Gambar 4.9 Rangkaian Driver *Syringe pump*

4.2.2.3 Perhitungan Nilai Komponen Relay *Syringe pump*



Gambar 4.10 Rangkaian relay *syringe pump*

Pada driver *syringe pump* ini menggunakan relay buatan OMRON dengan tipe G5LE yang memiliki spesifikasi

- V_R = 12 volt
- I_R = 33,3 mA
- *Coil resistance* =360 Ω

Dan *optocoupler* yang digunakan adalah 4N33 dengan spesifikasi

- V_F = 1,2 V
- I_{Fmax} = 80 mA
- turn off time 2 µs
- turn on time 100 µs
- I_{Cmax} = 150 mA
- V_{CEsat} = 1 V
- CTR_{min} = 500 %

Dari spesifikasi di atsa, maka dicari arus pada bagian transmiter atau I_F dengan menggunakan persamaan

$$CTR = \frac{I_C}{I_F} x\ 100\%$$

Dengan nilai CTR = CTR_{min} = 500% dan I_C = 33,3 mA, maka

$$500\% = \frac{33{,}3\ mA}{I_F} x\ 100\%$$

$$I_F = 6{,}66\ mA$$

Untuk perancangan nilai resistor LED optocoupler sebagai berikut

$$R_{led} = \frac{V_A - V_{Led} - V_F}{I_F}$$

$$= \frac{5V - 2V - 1{,}2V}{6{,}66\ X\ 10^{-3}}$$

$$= \frac{1{,}8V}{6{,}66\ X\ 10^{-3}}$$

$$= 270{,}27\ \Omega$$

R_{led} yang digunakan dalam perancangan ini adalah 270 Ω.

$$I_F = \frac{V_A - V_{led} - V_F}{R_{led}}$$

$$= \frac{5V - 2V - 1{,}2V}{270\Omega}$$

$$= \frac{1{,}8V}{270\Omega}$$

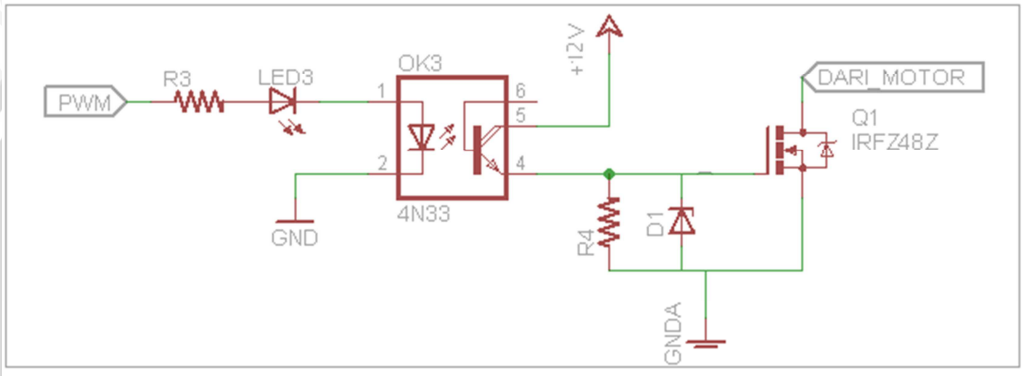
$$= 6{,}6\ mA$$

Jadi arus yang mengalir dalam LED optocoupler sebesar 15 mA. Masih lebih besar arus maksimal output mikrokontroler yang sebesar 20 mA.

Arus yang dibutuhkan *relay* untuk aktif sebesar 33,3 mA. Nilai ini masih dibawah nilai arus maksimal keluaran transistor optocoupler sebesar 150 mA.

4.2.2.4 Perancangan Pengontrol Kecepatan Driver *Syringe Pump*

Pada perancangan skripsi ini digunakan komponen E-MOSFET kanal N dengan masukan berupa sinyal PWM dari mikrokontroler. Gambar menunjukkan rangkaian pengontrol kecepatan motor.



Gambar 4.11 rangkaian pengontrol kecepatan *syrnge pump*

Jenis FET yang digunakan adalah IRFZ44N dengan spesifikasi

- $V_{DS\ max}$ = 60V
- $I_{D\ max}$ = 35A
- $R_{DS\ on\ max}$ = 0.028 Ω
- $V_{GS\ max}$ = 20V
- $V_{GS\ threshold\ max}$ = 4V

IRFZ44N digunakan karena IRFZ44N memiliki $I_{D\ max}$ yang cukup besar dan $R_{DS\ on\ max}$ lebih kecil dibandingkan beberapa jenis E-MOSFET kanal N yang lain. Hal ini mengakibatkan IRFZ44N tidak cepat menjadi panas ketika dilewati arus yang besar.

Untuk aplikasi pengontrol kecepatan motor, FET selalu dikondisikan dalam keadaan saturasi atau *cut off*-nya. Hal ini dimaksudkan agar tidak terlalu banyak daya yang terbuang dalam FET itu sendiri.

Untuk E-MOSFET kanal N syarat agar komponen dalam kondisi *cut off* adalah ketika $V_{GS} < V_{threshold}$. Dengan $V_{threshold\ max}$ IRFZ44N = 4V,maka V_{GS} *cut off* yang digunakan kurang dari 4V. V_{GS} yang digunakan adalah 0V.

Syarat agar E-MOSFET kanal N dalam kondisi aktif saturasi adalah ketika $V_{GS} > V_{threshold}$, dan $V_{DS} > (V_{GS} - V_{threshold})$. Menggunakan V_{DS} sebesar 12V, maka $V_{GS} < 16V$. Karena $V_{GS\ max} = 20V$, maka $4V < V_{GS} < 20V$. V_{GS} saturasi yang digunakan sebesar 10V. Untuk membatasi V_{GS} saturasi ini digunakan dioda zener 10V.

Perhitungan nilai resistor untuk tegangan masukan E-MOSFET sebagai berikut

$$R4 = \frac{12V - V_{CE\ sat}}{I_C}$$

$$= \frac{12V - 1V}{150\ X\ 10^{-3}\ A}$$

$$= 73{,}33\Omega$$

Nilai R4 yang digunakan sebesar 100 Ω.

$$I_C = \frac{12V - V_{CE\ sat}}{R_4}$$

$$= \frac{12V - 1V}{120}$$

$$= 91{,}66.10^{-3} = 91{,}66\ mA$$

Nilai ini masih dibawah nilai arus maksimal keluaran transistor *optocoupler* sebesar 150 mA. Arus masukan *optocoupler* I_F dapat dihitung dengan

$$CTR = \frac{I_C}{I_F} x\ 100\%$$

$$I_F = \frac{I_C}{CTR} x\ 100\%$$

$$I_F = \frac{91{,}66\ mA}{500\%} x\ 100\%$$

$$I_F = 18{,}33\ mA$$

Jadi arus yang mengalir dalam LED *optocoupler* sebesar 18,33 mA. Masih lebih besar arus maksimal output mikrokontroler yang sebesar 20 mA. Untuk perancangan nilai resistor LED *optocoupler* sebagai berikut

$$R_{led} = \frac{V_A - V_F - V_{LED}}{I_f}$$

$$= \frac{5V - 1{,}2V - 2{,}2V}{18{,}3\ X\ 10^{-3}}$$

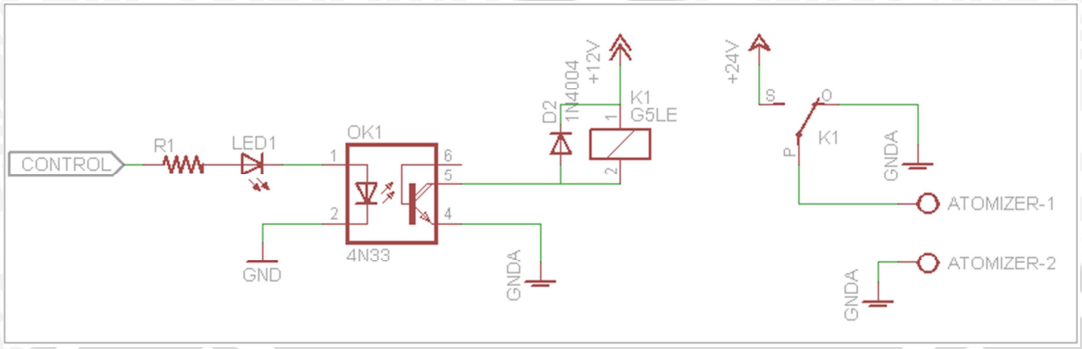
$$= \frac{1{,}6V}{18{,}33\ X\ 10^{-3}}$$

$$= 97{,}97\ \Omega$$

R_{led} yang digunakan dalam perancangan ini adalah 91 Ω.

4.2.2.5 Perancangan Driver *Ultrasonic Atomizer*

Pada driver *ultrasonic atomizer* menggunakan relay yang digunakan untuk mengantur kondisi mati atau menyala dari *ultrasonic atomizer.* Pada driver ini, saat mikrokontroler mengirimkan sinyal kontrol, maka *optocoupler* akan aktif dan menyalakan relay yang mengontrol *ultrasonic*



Gambar 4.12 rangkaian pengontrol *ultrasonic Atomizer*

Pada driver *ultrasonic atomizer* ini menggunakan relay buatan OMRON dengan tipe G5LE yang memiliki spesifikasi

- V_R = 12 volt
- I_R = 33.3 mA
- *Coil resistance* = 360 Ω

Dari spesifikasi di atas, maka dicari arus pada bagian transmiter atau I_F dengan menggunakan persamaan

$$CTR = \frac{I_C}{I_F} x\ 100\%$$

Dengan nilai CTR = CTR_{min} = 500% dan I_C = 33.3 mA, maka

$$500\% = \frac{33.3mA}{I_F} x\ 100\%$$

$$I_F = 6.66\ mA$$

Untuk perancangan nilai resistor LED optocoupler sebagai berikut

$$R_{led} = \frac{V_A - V_{led} - V_F}{I_F}$$

$$= \frac{5V - 2V - 1{,}2V}{6{,}66\ X\ 10^{-3}}$$

$$= \frac{1{,}8V}{6{,}66\ X\ 10^{-3}}$$

$$= 270{,}27\ \Omega$$

R_{led} yang digunakan dalam perancangan ini adalah 270 Ω.

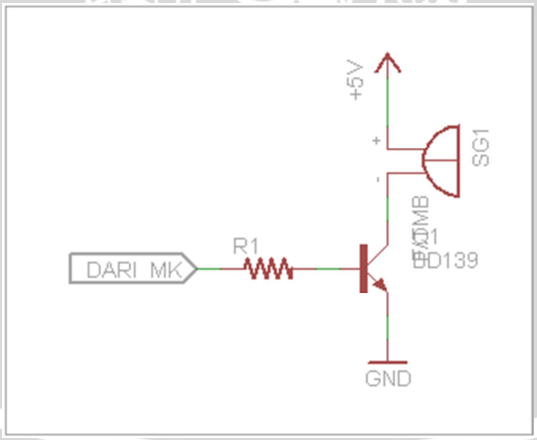
$$I_F = \frac{V_A - V_{led} - V_F}{R_{led}}$$
$$= \frac{5V - 2V - 1{,}2V}{270\Omega}$$
$$= \frac{1{,}8V}{270\Omega}$$
$$= 6{,}6\ mA$$

Jadi arus yang mengalir dalam LED optocoupler sebesar 6,6 mA. Masih lebih besar arus maksimal output mikrokontroler yang sebesar 20 mA.

Arus yang dibutuhkan *relay* untuk aktif sebesar 33,3 mA. Nilai ini masih dibawah nilai arus maksimal keluaran transistor optocoupler sebesar 150 mA.

4.2.2.6 Perancangan Driver *Buzzer*

Pada driver *buzzer*, digunakan driver transistor untuk menyalakan dan mematikan *buzzer* yang digunakan. Driver transistor ini menggunakan prinsip transistor sebagai saklar yaitu dengan mengaktifkan transistor saat kondisi saturasi untuk saklar tertutup dan kondisi *cut off* untuk saklar terbuka. Dengan memakai kondisi ini, maka saat driver transistor menerima masukan logika dari mikrokontroler driver akan dalam kondisi saturasi yang berarti *buzzer* akan menyala. Begitu juga sebaliknya saat masukan logika dari mikrokontroler diputus, driver akaan mengalami kondisi *cut off* sehingga *buzzer* mati.



Gambar 4.13 rangkaian driver *Buzzer*

Pada rangkaian ini menggunakan *buzzer* dengan tipe AT-1224-TWT-R yang memiliki spesifikasi

- *Rated Voltage* = 5 V
- *Operating voltage* = 3-8 V

- *Rated Current(max)* = 40 mA
- *Coil resistance* = 47 ± 9,4 ohm
- I_{Cmax} = 150 mA
- *Resonanct frequency* 2.400 ± 500 Hz

Dan transistor yang digunakan adalah BD 139 dengan spesifikasi

- V_{CE} sat = 0,5 V
- V_{BE} = 1 V
- hFE = 63
- I_{Cmax} = 1,5 A

Pada rangkaian ini transistor difungsikan sebagai saklar yang sehingga transistor harus difungsikan dalam keadaan saturasi (jenuh). Maka $I_{C(sat)}$ dapat diketahui dengan persamaan berikut

$$I_{c(sat)} = \frac{V_{cc} - V_{CE(sat)}}{R_{buzzer}}$$

$$I_{c(sat)} = \frac{5\,V - 0.5\,V}{47\,\Omega}$$

$$I_{c(sat)} = 95.7\,mA$$

Dari nilai $I_{c(sat)}$ maka dapat diketahui I_B

$$I_B = \frac{I_{c(sat)}}{h_{FE}}$$

$$I_B = \frac{95.7\,mA}{63}$$

$$I_B = 1.52\,mA$$

Maka

$$R_B = \frac{V_{in} - V_{BE(sat)}}{I_B}$$

$$R_B = \frac{5\,V - 1\,V}{1.52\,mA}$$

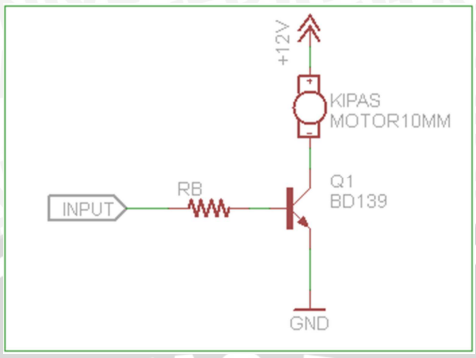
$$R_B = 2.58\,K\Omega$$

Karena nilai R_B =2.52 KΩ tidak termasuk resistor standart maka digunakan nilai R_B = 2.4 KΩ

4.2.2.7 Perancangan Driver Kipas

Rangkaian driver kipas menggunakan driver transistor yang difungsikan dalam kondisi sebagai saklar. Saat transistor dikondisikan sebagai saklar, maka transistor bekerja

pada daerah saturasi dan *cut off*. Saat driver diberi masukan dari mikrokontroler, maka transistor dalam kondidi saturasi yaitu seperti saklar tertutup sehingga kipas menyala.



Gambar 4.14 rangkaian driver kipas

Pada rangkaian ini menggunakan kipas *brushless* memiliki spesifikasi

- *Rated Voltage* = 12 V
- *Rated Current(max)* = 0.08 mA

Dan transistor yang digunakan adalah BD 139 dengan spesifikasi

- V_{CE} sat = 0,5 V
- V_{BE} = 1 V
- hFE = 63
- I_{Cmax} = 1,5 A

Pada rangkaian ini transistor difungsikan sebagai saklar yang sehingga transistor harus difungsikan dalam keadaan saturasi (jenuh). Maka $I_{C(sat)}$

Dari nilai $I_{c(sat)}$ maka dapat diketahui I_B

$$I_B = \frac{I_{c(sat)}}{h_{FE}}$$

$$I_B = \frac{80\ mA}{63}$$

$$I_B = 1.27\ mA$$

Maka

$$R_B = \frac{V_{in} - V_{BE(sat)}}{I_B}$$

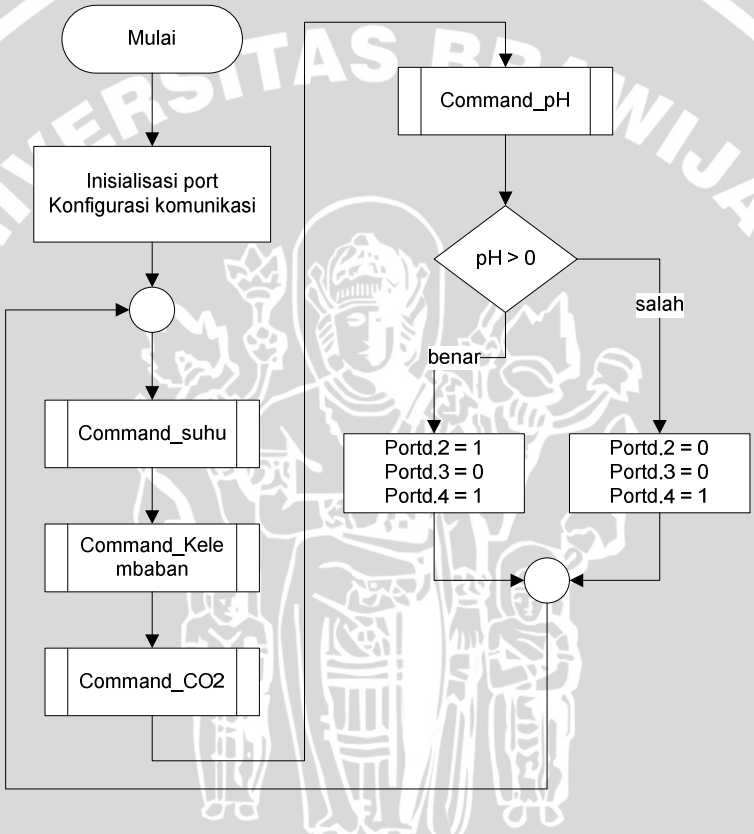
$$R_B = \frac{12\ V - 1\ V}{1.27\ mA}$$

$$R_B = 8.66\ K\Omega$$

Karena nilai R_B =8.66 KΩ tidak termasuk resistor standart maka digunakan nilai R_B = 8,2 KΩ

4.3 Perancangan Perangat Lunak

Tujuan dari program utama adalah mengatur urutan kerja sistem sehingga sistem mampu menjalankan fungsinya dengan baik. Secara umum tugas yang harus dikerjakan oleh mikrokontroler *slave* meliputi pengontrolan aktuator dari sistem *live cell chamber* seperti aktuator kelembaban yaitu *ultrasonic atomizer* dan kipas, aktuator suhu yaitu *peltier* dan pompa, aktuator CO_2 yaitu valve gas, aktuator pH yaitu *syringe pump.* Selain itu mikrokontroler slave juga menangani komunikasi serial dengan mikrokontroler master. Diagram alir program utama mikrokontroler slave ditunjukkan dalam Gambar 4.15.



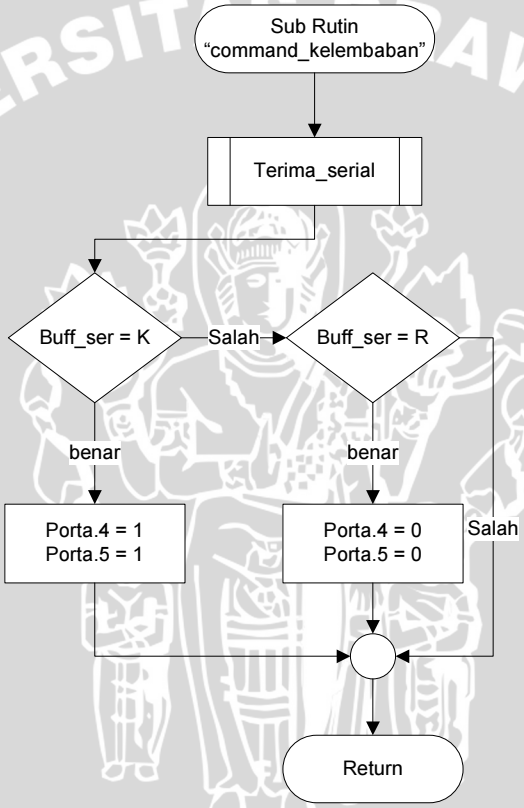
Gambar 4.15 Flow chart algoritma program utama mikrokontroler slave

Saat sistem *ON,* mikrokontroler akan melakukan inisialisasi sistem seperti untuk mengatur port yang digunakan, komunikasi USART, dan lain lain yang akan digunakan. Kemudian sistem akan menjalankan sub rutin command_suhu yang akan membaca data serial yang dikirimkan oleh mikrokontroler master dan data yang didapat akan mengaktifkan aktuator yang ada. Setelah itu akan menjalankan sub rutin command_kelembaban yang meminta data juga dari mikrokontroler master untuk mengaktifkan aktuator yang ada. Selanjutnya sistem akan menjalankan sub rutin command CO_2 dan kemudian menjalankan sub rutin command_ph yang berfungsi menghitung data

error pH yang dikirimkan oleh mikrokontroler master yang digunakan sebagai data nilai pH. Bila nilai pH > dari 0 maka aktuator pH yaitu syringe pump akan aktif sedang bila kurang dari 0 maka akan berhenti. Kemudian program akan mengulang proses yang ada hingga sistem dimatikan.

4.3.1 Algoritma Sub Rutin Command_Kelembaban

Pada sub rutin command_kelembaban digunakan untuk mengaktifkan aktuator kelembaban yaitu *ultrasonic atomizer* dan untuk membantu sirkulasi udara lembab tersebut digunakan kipas berdasarkan masukan data dari mikrokontroler master. Diagram alir dari sub rutin command_kelembaban ini ditunjukkan pada Gambar 4.16.

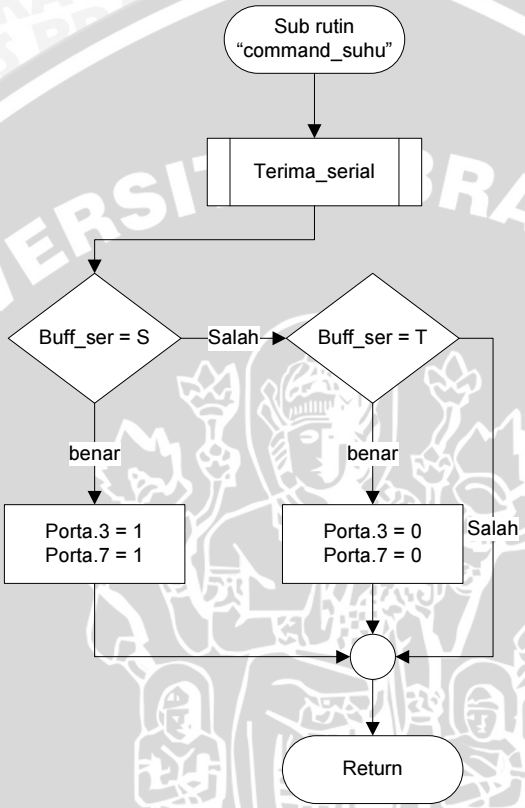


Gambar 4.16 Flowchart algoritma sub rutin command_kelembaban

Pada proses awal dari sub rutin ini, program akan memanggil sub rutin terima serial yang berfungsi menerima data serial dari mikrokontroler master. Setelah menjalankan sub rutin terima_serial, data buff_ser yang didapat dicek apakah sama dengan "K" apabila benar maka PORT A.4 yaitu logika driver kipas dan PORT A.5 yaitu logika driver *ultrasonic atomizer* akan diaktifkan. Sedangkan bila data buff_ser sama dengan "R" maka PORT A.4 dan A.5 akan dimatikan. Bila program telah selesai maka program akan kembali ke program awal.

4.3.2 Algoritma Sub Rutin command_Suhu

Sub rutin command_suhu berguna untuk mengaktifkan actuator suhu yaitu *peltier* dan pompa air yang digunakan untuk mengalirkan air panas yang ada**.** Sub rutin ini juga mengandalkan data dari mikrokontroler master untuk menjalankan tugasnya. Diagram alir dari sub rutin command_suhu dapa dilihat pada Gambar 4.17.

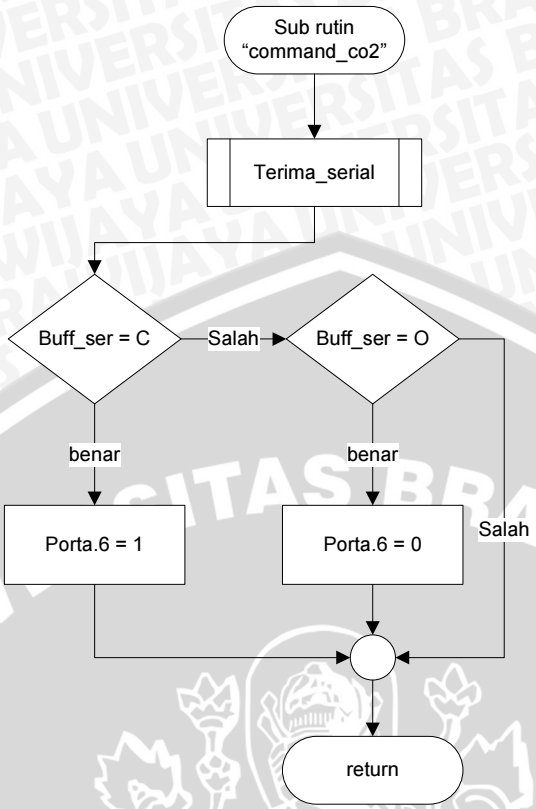


Gambar 4.17 Flow chart algoritma sub rutin command_suhu

Proses awal dari sub rutin ini adalah memanggil sub rutin terima_serial untuk mendapatkan daa yang dikirimkan oleh mikrokontroler master. Setelah data buff_ser didapat dilihat data tersebut apakah sama dengan ”S”. Bila benar maka PORT A.3 yaitu logika pompa dan PORT A.7 yaitu logika Peltier akan diaktifkan. Sedangka bila data buff_ser sama dengan ”T” maka PORT A.3 dan PORT A.7 akan dimatikan. Bila data buff_ser bukan kedua nilai tersebut program akan kembali ke program awal.

4.3.3 Algoritma Sub Rutin Command_CO_2

Sub rutin command_ CO_2 ini digunakan untuk mengaktifkan actuator CO_2 yaitu valve gas CO_2 sesuai data yang diterima dari mikrokontroler master. Diagram alir sub rutin command_CO2 ini dapat dilihat pada Gambar 4.18.

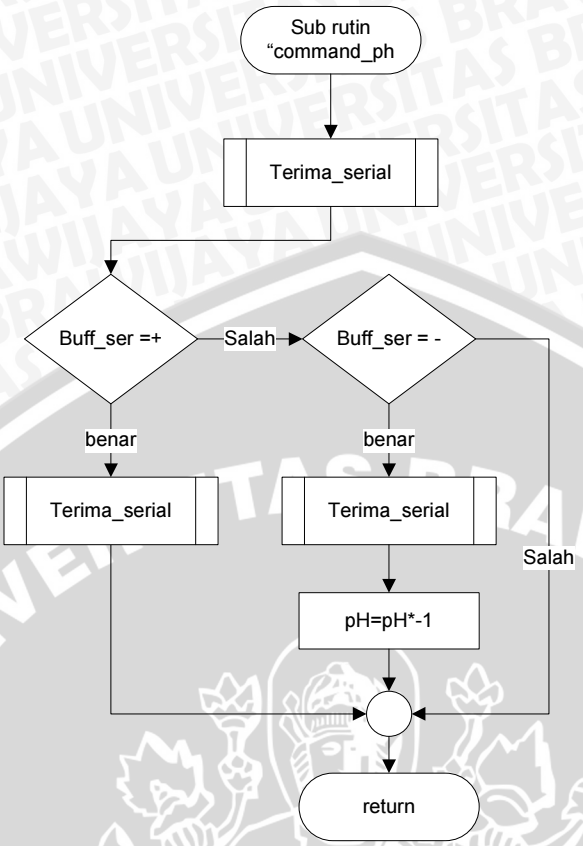


Gambar 4.18 Flow chart algoritma sub rutin command_co2

Proses awal dari sub rutin ini adalah memanggil sub rutin terima_serial untuk mendapatkan data yang dikirimkan oleh mikrokontroler master. Setelah data buff_ser didapat dilihat data tersebut apakah sama dengan ”C”. Bila benar maka PORT A.6 yaitu logika valve gas akan diaktifkan. Sedangka bila data buff_ser sama dengan ”O” maka PORT A.6 akan dimatikan. Bila data buff_ser bukan kedua nilai tersebut program akan kembali ke program awal.

4.3.4 Algoritma Sub Rutin command_pH

Pada sub rutin command_ pH ini digunakan untuk mengaktifkan actuator pH yaitu *syringe pump* sesuai data yang diterima dari mikrokontroler master. Data yang dikirim oleh mikrokontroler master juga sedikit berbeda dengan data yang lain karena data yang dikirim berupa tanda “+” atau “-“ yang menunjukan pH perlu dinaikkan atau diturunkan dan disusul dengan sutu nilai yang merupakan selisih antara set point pH dan nilai pH yang terukur. Diagram alir sub rutin command_ph ini dapat dilihat pada Gambar 4.19.

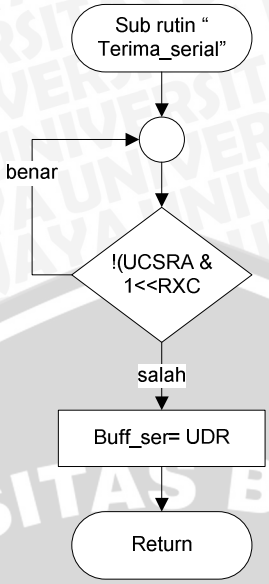


Gambar 4.19 Flow chart algoritma sub rutin command_ph

Proses awal dari sub rutin ini adalah memanggil sub rutin terima_serial untuk mendapatkan data yang dikirimkan oleh mikrokontroler master. Setelah data buff_ser didapat, dilihat data tersebut apakah sama dengan ”+”. Bila benar maka program akan memanggil sub rutin hitung_ph yang akan mengambil nilai selisih antara set point pH dengan nilai pH yang terukur. Bila nilai buff_ser yang didapat adalah ”-” maka program akan memanggil sub rutin hitung_ph dan kemudian nilai ph yang didapat dikalikan dengan -1 untuk mendapatkan nilai negatif. Bila data pada buff_ser bukan keduanya maka program akan kembali ke program awal.

4.3.5 Algoritma Sub Rutin Terima_Serial

Pada sub rutin terima_serial digunakan untuk melakukan komunikasi serial antara mikrokontroler master dan slave. Diagram alir sub rutin terima_serial ini dapat dilihat pada Gambar 4.20.

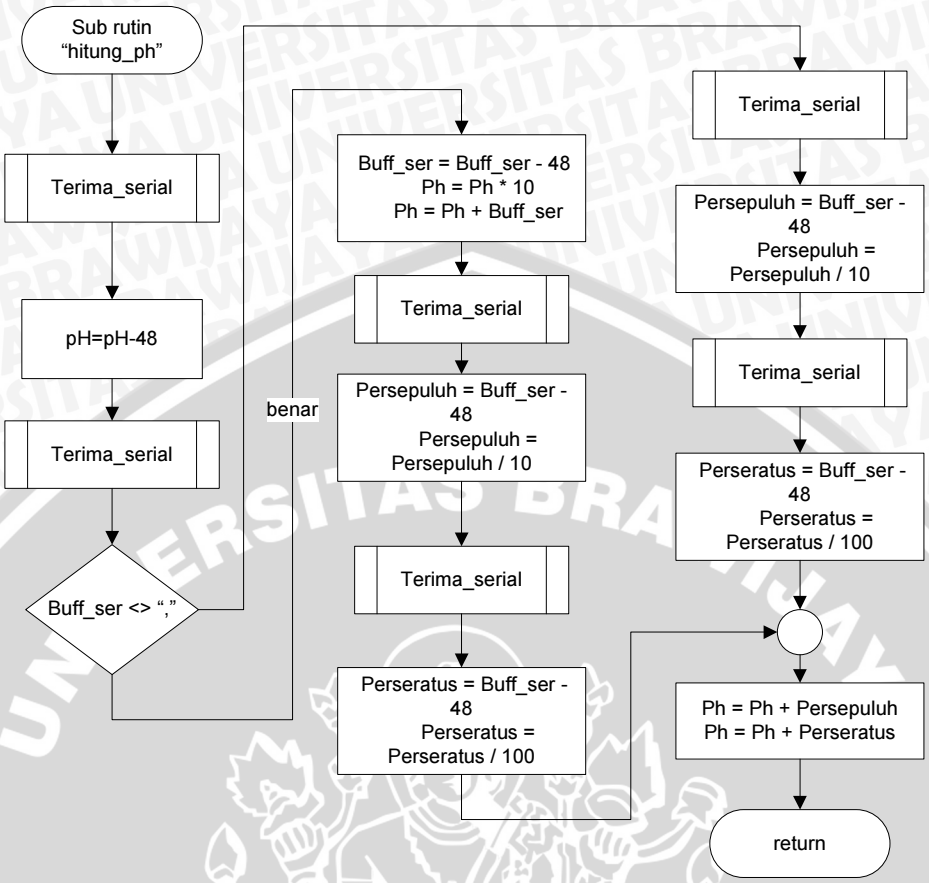


Gambar 4.20 Flow chart algoritma sub rutin terima_serial

Pada program ini, sistem akan menunggu data telah diterima dalam buffer penerima yaitu yang tersimpan pada UDR atau USART I/O Data Register dengan megecek Flag register RXC atau USART Complete Receive yang merupakan bagian dari register UCSRA atau USART Control and Status Register A. Bila RXC bernilai satu maka data sudah diterima dalam buffer penerima dan kemudian data tersebut yang dapat juga diakses melalui register UDR dimasukkan ke variable Buff_ser. Dan program akan kembali setelah melakukan proses tersebut.

4.3.6 Algoritma Sub Rutin Hitung_PH

Pada bub rutin ini berrfungsi untuk menerima data nilai selisih antara set point pH dan nilai pembacaan sensor pH. Nilai yang didapat dari sub rutin ini digunakan untuk masukan dalam logika pengontrolan *syringe pump*. Diagram alir sub rutin hitung_ph ini dapat dilihat pada Gambar 4.21.



Gambar 4.21 Flowchart algoritma sub rutin hitung_ph

Pertama program ini akan memanggil sub rutin terima_serial untuk mendapatkan data yang dikirim mikrokontroler master. Data yang diterima akan dikurangkan 48 desimal karena data yang dikirim melalui komunikasi serial menggunakan format ASCII. Setelah selesai program memanggil kembali sub rutin terima_serial. Bila nilai buff_ser sama dengan "," yang berarti nilai yang dikirm kurang dari puluhan, maka program akan lengsung menerima nilai persepuluh dan perseratusnya. Bila tidak maka nilai ph awal akan dikalikan sepuluh dan ditambahkan nilai yang berada di dalam buff_ser setelah itu baru menghitung nilai persepuluh dan per seratus dari data yang dikirim. Setelah proses tersebut nilai pH akan ditambahkan dengan nilai persepuluh dan perseratus untuk mendapatkan nilai yang lengkap. Setelah sub rutin selesai, sub rutin akan kembali ke program awal