PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

TUGAS AKHIR
AQUARIUM CONTROLLER BERBASIS ATMEGA128 SEBAGAI PENJAGA STABILITAS SISTEM AKUARIUM LAUT

Diajukan untuk memenuhi salah satu syarat
Memperoleh gelar Sarjana Teknik pada
Jurusan Teknik Elektro

Disusun oleh:
Adhipa Tri Setyawan Alim
NIM : 095114001

JURUSAN TEKNIK ELEKTRO
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA
YOGYAKARTA
2013

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

FINAL PROJECT
AQUARIUM CONTROLLER BASED ON ATMEGA128
AS STABILITY CHECKER OF MARINE AQUARIUM

Presented as Partial Fulfillment of the Requirements
To Obtain the *Sarjana Teknik* Degree
In Department of Electrical Engineering

By:
Adhipa Tri Setyawan Alim
NIM : 095114001

DEPARTMENT OF ELECTRICAL ENGINEERING
FACULTY OF SCIENCE AND TECHNOLOGY
SANATA DHARMA UNIVERSITY
YOGYAKARTA
2013

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

HALAMAN PERSETUJUAN

TUGAS AKHIR

AQUARIUM CONTROLLER BERBASIS ATMEGA128 SEBAGAI PENJAGA STABILITAS SISTEM AKUARIUM LAUT (AQUARIUM CONTROLLER BASED ON ATMEGA128 AS STABILITY CHECKER OF MARINE AQUARIUM)

Disusun oleh:
Adhipa Tri Setyawan Alim
NIM : 095114001

Telah disetujui oleh:

Pembimbing

Ir. Tjendro, M. Kom

Tanggal : 24 September 2013

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

HALAMAN PENGESAHAN

TUGAS AKHIR

AQUARIUM CONTROLLER BERBASIS ATMEGA128 SEBAGAI PENJAGA STABILITAS SISTEM AKUARIUM LAUT (AQUARIUM CONTROLLER BASED ON ATMEGA128 AS STABILITY CHECKER OF MARINE AQUARIUM)

Disusun oleh:
Adhipa Tri Setyawan Alim
NIM : 095114001

Telah dipertahankan di depan panitia penguji pada tanggal 21 Oktober 2013
Dan dinyatakan memenuhi syarat

Susunan Panitia Penguji

Nama Lengkap	Tanda Tangan
Ketua : **Ir. Th. Prima Ari Setiyani, M. T.**	.....
Sekretaris : **Ir. Tjendro, M. Kom**	.....
Anggota : **Petrus Setyo Prabowo, S.T., M. T.**	.....

Yogyakarta, 28 Oktober 2013
Fakultas Sains dan Teknologi
Universitas Sanata Dharma
Dekan,

P. H. Prima Rosa, S. Si, M. Sc.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

PERNYATAAN KEASLIAN KARYA

Saya menyatakan dengan sesungguhnya bahwa tugas akhir ini tidak memuat karya atau bagian orang lain, kecuali yang telah disebutkan dalam kutipan dan daftar pustaka, sebagaimana layaknya karya ilmiah.

Yogyakarta, 23 September 2013

Adhipa Tri Setyawan Alim

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

HALAMAN PERSEMBAHAN DAN MOTTO HIDUP

Problems are not stop signs,

They are guidelines.

~Robert H. Schuller

Skripsi ini kupersembahkan untuk:
Yesus Kristus Juru Slamat ku
Papa, Mama dan Kakakku tersayang
Raras, kekasihku tercinta
Sahabat dan teman – temanku

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Yang bertanda tangan di bawah ini, saya mahasiswa Universitas Sanata Dharma:

Nama : Adhipa Tri Setyawan Alim

Nomor Mahasiswa : 095114001

Demi pengembangan ilmu pengetahuan, saya memberikan kepada Perpustakaan Universitas Sanata Dharma karya ilmiah saya yang berjudul:

***AQUARIUM CONTROLLER* BERBASIS ATMEGA128 SEBAGAI PENJAGA STABILITAS SISTEM AKUARIUM LAUT**

Beserta perangkat yang diperlukan (bila ada). Dengan demikian saya memberikan kepada Perpustakaan Universitas Sanata Dharma hak untuk menyimpan, mengalihkan dalam bentuk media lain, mengelolanya dalam bentuk pangkalan data, mendistribusikan secara terbatas, dan mempublikasikannya di Internet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari saya maupun memberikan royalti kepada saya selama tetap mencantumkan nama saya sebagai penulis.

Demikian pernyataan ini yang saya buat dengan sebenarnya.

Yogyakarta, 23 September 2013

Adhipa Tri Setyawan Alim

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

INTISARI

Hobi akuarium laut saat ini semakin banyak peminatnya. Ikan laut dan terumbu karang memiliki warna yang beraneka ragam dan menjadi daya tarik tersendiri bagi para peminatnya. Namun demikian, tidak mudah untuk menjalani hobi ini. Banyak kendala dan hal – hal yang perlu diperhitungkan dalam menjalani hobi ini seperti kestabilan parameter air akuarium itu sendiri. Tak jarang para peminat hobi ini menyerah karena tidak adanya pengetahuan yang cukup dalam memelihara biota laut yang dipelihara.

Dengan adanya alat kontrol ini, penulis berharap para peminat hobi ini akan lebih terbantu dalam menjaga kestabilan sistem akuarium mereka. Alat kontrol yang dibuat menggunakan ATmega128 sebagai pusat kendalinya. Beberapa aspek yang dikontrol antara lain pencahayaan, penambahan bahan aditif, dan suhu. Ketiga aspek ini sangat penting di dalam memelihara akuarium laut. Pengujian alat kontrol dilakukan dengan membandingkan hasil pengujian parameter air akuarium setelah menggunakan alat kontrol dan sebelum menggunakan alat kontrol. Beberapa parameter air yang akan diuji antara lain kadar garam, kadar kalsium, kadar karbonat, dan suhu.

Dari hasil pengujian dan pengambilan data, alat kontrol ini sudah dapat bekerja. Alat yang digunakan mampu mempertahankan kadar garam, kadar kalsium dan suhu sampai dengan 95 % sedangkan kestabilan kadar karbonat pada 80 %. Alat kontrol juga dapat melakukan simulasi terjadinya *sunrise* dan *sunset.*

Kata kunci: Akuarium laut, Alat kontrol, ATmega128.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

ABSTRACT

Marine aquarium now has more and more enthusiast. Marine fishes and coral reefs had a multiform color and become attraction for their enthusiast. However, it is not easy to go trough this hobby. Many obstacles and things that must be prepared to undergo this hobby such as the stability of aquarium's water parameter itself. Many of marine aquarium enthusiast gave up because they lack of information and knowledge in maintaining their pets.

With this controller, the author hopes this hobby's enthusiasts will be helped in stabilizing their aquarium system. The controller uses ATmega128 for their central control. Some aspects are controlled, among others, the lighting, the addition of additives, and temperature. This aspects are important for maintaining marine aquarium. Testing this controller is done by comparing the result of water parameter after using controller and before that. Some of the water parameter that will be tested include specific gravity, calcium levels, carbonate levels and temperature.

From the result of testing and collecting data, this controller can already work. This tool can mantain specific gravity, calcium levels and temperature stability up to 95%, and maintain carbonate levels stability up to 80%. This tool can also perform sunrise and sunset simulation.

Key words: Marine aquarium, Controller, ATmega128.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

KATA PENGANTAR

Syukur dan terima kasih kepada Tuhan Yesus Kristus karena dengan segala rahmat dan bimbingan-Nya maka tugas akhir dengan judul "*Aquarium Controller* berbasis ATmega128 sebagai Penjaga Stabilitas Akuarium Laut" ini dapat diselesaikan dengan baik.

Selama menulis tugas akhir ini, penulis menyadari bahwa ada begitu banyak pihak yang telah memberikan bantuan dengan caranya masing – masing, sehingga tugas akhir ini bisa diselesaikan. Oleh karena itu penulis ingin mengucapkan terima kasih kepada:

1. Ibu P. H. Prima Rosa, S. Si., M. Sc., selaku Dekan Fakultas Sains dan Teknologi Universitas Sanata Dharma.
2. Bapak Ir. Tjendro, M. Kom., selaku dosen pembimbing yang dengan penuh kesabaran membimbing, memberikan saran dan kritik yang membantu penulis dalam menyelesaikan tulisan ini.
3. Ibu Ir. Theresia Prima Ari Setiyani, M. T. dan Bapak Petrus Setyo Prabowo, S. T., M. T. selaku dosen penguji.
4. Seluruh dosen prodi Teknik Elektro dan laboran yang telah memberikan ilmu pengetahuan kepada penulis selama kuliah.
5. Papa dan Mama tercinta, terima kasih untuk semua perhatian dan dukungan baik spiritual maupun material.
6. Kedua kakak, Alim Denny Kristiawan, S. Si. dan Alim Untung Widodo, S. T. yang telah memberikan masukan – masukan dan dukungan material.
7. Hayu Ajeng Anggana Raras, terima kasih karena telah memberikan motivasi dan semangat untuk segera menyelesaikan penulisan tugas akhir ini.
8. Teman – teman prodi Teknik Elektro angkatan 2009, atas kebersamaannya selama penulis menjalani masa studi.
9. Seluruh anggota *reefsforum* dan *indoreefclub* yang telah memberikan kritik, saran dan ide dalam membuat tugas akhir ini.
10. Semua pihak yang tidak bisa disebutkan satu persatu atas bantuan, bimbingan, kritik dan saran.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Dengan rendah hati penulis menyadari bahwa tugas akhir ini masih jauh dari sempurna, oleh karena itu berbagai kritik dan saran untuk perbaikan tugas akir ini sangat diharapkan. Akhir kata, semoga tugas akhir ini dapat bermanfaat bagi semua pihak. Terima kasih.

Yogyakarta, 23 September 2013

Penulis

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

DAFTAR ISI

Halaman

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

DAFTAR GAMBAR

Halaman

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

DAFTAR TABEL

Halaman

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

BAB I
PENDAHULUAN

1.1. Latar Belakang Masalah

Hobi akuarium sangatlah menyenangkan. Khususnya akuarium air laut, yang belakangan ini semakin banyak penggemarnya. Setidaknya ada 6000 orang menjadi *member* sebuah forum komunikasi di internet yang khusus membahas mengenai akuarium laut, antara lain *reefsforum.com* dan sebagainya. Hobi memelihara ikan hias air laut di dalam akuarium sudah ada sejak tahun 1880. Akuarium air laut di Indonesia mulai dikenal sejak zaman Hindia Belanda sekitar tahun 1922 [1]. Penggemar akuarium laut semakin banyak jumlahnya, karena ikan hias air laut memiliki warna yang menarik dan bervariasi.

Sebagian besar peminat hobi ini putus di tengah jalan, dikarenakan kurangnya pengetahuan mereka akan beberapa unsur penting dalam memelihara akuarium laut. Kestabilan parameter air yang sulit dijaga menjadi masalah bagi mereka yang baru memulai hobi ini. Kerugian material bagi peminatnya, juga kematian biota laut yang dipelihara sangat disayangkan. Apabila kematian biota ini terus berlangsung, ditambah banyaknya peminat hobi yang sistemnya tidak stabil, maka kelestarian lingkungan laut bisa terancam, karena biota yang dijual sebagian besar adalah hasil mengambil langsung dari alam.

Sebaliknya, di tangan peminat hobi yang serius dalam menjalani hobi ini, biota laut yang semula diambil dari alam, bisa berkembang biak di dalam akuarium. Hal ini tentu saja membantu dalam pelestarian alam itu sendiri. Mereka yang berhasil dalam memelihara kestabilan sistem akuarium laut ini biasanya menggunakan bantuan dari kontroler. Hal ini dikarenakan setiap peminat hobi ini tidak selalu ada untuk memantau akuariumnya.

Penggunaan alat kontrol akan membantu peminat hobi ini dalam menjaga parameter air akuarium mereka. Parameter air seperti suhu, salinitas, dan unsur kimia lainnya menjadi lebih stabil dengan bantuan alat ini. Alat ini juga biasa digunakan untuk mengatur kebutuhan pencahayaan akuarium yang lamanya tidak lebih dari 12 jam (siang dan malam) [2]. Dengan adanya kontrol pencahayaan, memungkinkan sumber cahaya buatan yang digunakan dalam akuarium untuk menyimulasikan terjadinya siang dan malam secara *real-time.*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Harga perangkat ini masih sangat mahal, padahal kestabilan sistem dan fungsi akuarium sangat bergantung pada alat ini. Sebagai contoh, apabila peminat hobi menggunakan sebuah kontroler suhu, setidaknya peminat hobi akan menghabiskan uang $ 134,00 (atau kurang lebih 1,3 juta Rupiah). Untuk sebuah kontroler lampu bisa menghabiskan uang $ 749.99 (atau kurang lebih 7,5 juta Rupiah) dan sebuah kontroler penambahan aditif sebesar $ 359.99 (atau kurang lebih 3,6 juta Rupiah). Dana yang sebesar ini tentu saja sangat mahal, terutama bagi mereka yang berasal dari kalangan menengah. Alat – alat kontrol ini juga masih didominasi oleh buatan luar negeri. Sangat sedikit bahkan hampir tidak ada inovasi di bidang ini yang berasal dari dalam negeri.

1.2. Tujuan dan Manfaat

Tujuan dilakukannya penelitian ini adalah:

1. Membuat alat kontrol yang dapat menjaga parameter suhu, kadar garam, kalsium dan karbonat dalam akuarium.
2. Membuat alat kontrol yang dapat mensimulasikan proses terjadinya siang dan malam melalui kontrol pencahayaan.

Manfaat dilakukannya penelitian ini adalah:

1. Membantu peminat hobi akuarium laut dalam menjaga kestabilan parameter suhu, kadar garam, kalsium dan karbonat dalam akuarium.
2. Membantu peminat hobi akuarium laut dalam mengontrol intensitas pencahayaan dan mensimulasikan terjadinya siang dan malam.

1.3. Batasan Masalah

Karena kompleksnya sistem akuarium laut, maka peneliti membataskan permasalahan hanya pada:

1. Kontrol sistem pencahayaan hanya terbatas pada sumber cahaya yang menggunakan *High Power Led (HPL)*.
2. Kontrol suhu menggunakan metode On – Off Histerisis, yang nilai titik tengah suhunya diatur oleh *user*.
3. Kontrol penambahan bahan aditif menggunakan sensor *float switch* dan sistem *timer* tergantung dari pilihan *user*.
4. Dalam pengujian alat kontrol, bahan aditif yang digunakan terbatas hanya pada penggunaan larutan *kalkwasser*.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

5. Menggunakan ATMega128 sebagai pusat kontrol.

1.4. Metode Penulisan

Metode penulisan yang digunakan adalah:

1. Pengumpulan buku – buku dan referensi tentang pembuatan ekosistem akuarium air laut.
2. Pembuatan ekosistem akuarium air laut, lengkap dengan sistem filtrasi sesuai dengan referensi terkait. Hal ini bertujuan agar ada perbandingan antara kondisi akuarium sebelum menggunakan alat kontrol, dengan sesudah menggunakan alat kontrol.
3. Pembuatan sistem kontrol akuarium air laut. Aspek yang akan dikontrol adalah sistem pencahayaan, kestabilan suhu, dan penambahan bahan aditif serta kestabilan kadar garam. Implementasi alat kontrol menggunakan bantuan dari mikrokontroler. Jenis mikrokontroler yang digunakan adalah ATmega128.
4. Pengambilan data dilakukan setelah ekosistem akuarium selesai dibuat (sebagai data sebelum menggunakan alat kontrol), sampai dengan setelah alat kontrol selesai dibuat.
5. Analisa dan pengambilan kesimpulan dilakukan dengan membandingkan kondisi kestabilan parameter suhu, salinitas, kalsium dan alkalinitas akuarium berdasarkan data sebelum menggunakan alat kontrol dan data sesudah menggunakan alat kontrol.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

BAB II
DASAR TEORI

2.1. Pencahayaan Akuarium

Pencahayaan adalah faktor utama pada keadaan fisiologi terumbu yang bersimbiosis dengan *photothropic microorganism (zooxanthellate corals)* untuk melakukan fotosintesis [3]. Hampir semua terumbu yang dipelihara pada akuarium bersimbiosis dengan mikro organisme ini. *Zooxanthallae* memproduksi oksigen dan makanan yang berguna bagi pertumbuhan terumbu dari proses fotosintesis. Pada proses fotosintesis terumbu, terdapat istilah intensitas *PAR* atau *Photosynthetically Active Radiation* yang menunjukkan seberapa besar muatan *photon* dari cahaya yang mampu diterima dan digunakan dalam proses fotosintesis. Ukuran intensitas *PAR* ini tidak dapat diturunkan menjadi ukuran intensitas cahaya (*lumen*).

Matahari adalah sumber cahaya utama bagi terumbu yang ada di laut untuk melakukan fotosintesis. Matahari mencakup panjang gelombang yang sangat lebar, dari mulai gelombang inframerah sampai gelombang ultraviolet. Pada saat mencapai laut, cahaya yang dihasilkan oleh matahari mengalami atenuasi sehingga intensitas cahaya dan nilai *PAR (Photosyntetically Active Radiation)* yang dihasilkan menjadi berkurang. Pada air yang jernih, gelombang warna biru masih bisa mencapai kedalaman 100 m di bawah permukaan air bahkan lebih, sedangkan gelombang warna merah sudah hampir tidak ada pada kedalaman 1 – 5 m [3]. Hampir semua terumbu dapat memproses gelombang warna biru, sedangkan gelombang warna merah jarang sekali digunakan dalam proses fotosintesis. Panjang gelombang warna merah juga dapat memicu mikro organisme lain yang tidak diinginkan, untuk tumbuh di akuarium seperti ganggang hijau dan *cyanobacteria* sehingga warna ini jarang digunakan pada akuarium laut.

Di alamnya, intensitas cahaya yang diterima terumbu tidaklah konstan. Intensitas cahaya yang dihasilkan matahari akan bertambah sepanjang waktu, sehingga mencapai puncaknya dan akhirnya berkurang lagi [2]. Siklus siang dan malam ini terjadi terus menerus sehingga intensitas cahaya tertinggi yang dihasilkan matahari hanya berlangsung beberapa jam saja. Pengaruh cuaca dan awan juga mempengaruhi intensitas cahaya yang sampai ke permukaan air. Sehingga jumlah intensitas cahaya yang diterima dan lamanya proses pencahayaan yang dilakukan oleh terumbu hanya beberapa jam saja.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Kebutuhan terumbu akan cahaya sangatlah beragam, tergantung dari mana dan bagaimana keadaan alam pada saat terumbu itu dipanen [2]. Semakin besar intensitas *PAR* yang dihasilkan oleh sebuah sumber cahaya, semakin besar pula aktifitas fotosintesis yang terjadi pada terumbu yang menerima cahaya tersebut. Terumbu juga dapat menyesuaikan diri dengan keadaan lingkungan sekitar, termasuk keadaan cahaya pada tempat hidupnya. Ada beberapa hal yang menunjukkan proses adaptasi terumbu terhadap intensitas cahaya yang diterimanya:

1. Terumbu dapat memanjangkan atau memendekkan tentakelnya sehingga menyebabkan luas permukaan yang menerima cahaya juga akan bertambah atau berkurang.
2. *Zooxanthallae* dapat mengubah pigmen yang dimilikinya, baik komposisinya ataupun konsentrasinya. Mereka dapat beradaptasi dalam beberapa hari atau minggu terhadap besarnya intensitas cahaya yang diterima, bahkan perubahan komposisi cahaya (perbandingan intensitas gelombang warna merah dan biru) juga dapat berpengaruh.
3. Terumbu yang berada pada kondisi kekurangan cahaya akan cenderung tumbuh secara horizontal karena mencari sumber cahaya yang lebih, sedangkan terumbu yang berada pada kondisi kelebihan cahaya akan cenderung memiliki badan yang lebih datar dan rata.

Meskipun terumbu dapat beradaptasi pada kondisi cahaya yang sangat beragam, tetapi pada sistem tertutup seperti akuarium, perubahan beberapa parameter secara serentak dapat membuat terumbu menjadi *stress* dan mati, termasuk yang disebabkan oleh perubahan intensitas cahaya. Perubahan pada sistem pencahayaan harus dilakukan secara perlahan, agar terumbu dapat beradaptasi dengan baik.

Beberapa jenis sumber cahaya yang sering digunakan pada akuarium laut antara lain lampu *fluorescent* dan lampu *metal halide.* Kedua jenis lampu ini telah lama dikenal dapat menggantikan fungsi matahari dan mampu menghasilkan intensitas *PAR* yang cukup bagi terumbu untuk melakukan fotosintesis. Penggunaan *High Power Led* sebagai pengganti matahari masih tergolong teknologi baru di hobi ini [3]. Jenis lampu ini selain memiliki spektrum warna yang luas, juga lebih efisien dan hemat listrik.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

2.2. Suhu Akuarium

Suhu adalah parameter fisika paling penting dalam memelihara terumbu. Apabila semua parameter yang lain dalam keadaan baik sedangkan parameter suhu tidak, maka terumbu tidak akan selamat untuk waktu yang lama [2]. Idealnya suhu akuarium berkisar antara 23^{o} – 26 ^{o}C. Parameter ini juga harus stabil. Fluktuasi suhu pada akuarium laut harus kurang dari dua derajat celsius. Fluktuasi suhu yang besar dapat menimbulkan *stress* pada biota terutama ikan. *Chiller* biasa digunakan untuk membantu menjaga kestabilan suhu di akuarium laut.

2.3. Kadar Garam Akuarium (Specific Gravity / SG)

Specific gravity (SG) adalah satuan yang digunakan untuk mengetahui perbandingan rata – rata salinitas yang dimiliki air laut dengan air destilasi (akuades). Akuades memiliki nilai *SG* 1.000 sedangkan air laut memiliki nilai *SG* antara 1.022 sampai 1.032. Nilai *SG* ini sangat tergantung dengan suhu air [2].

Jarang terjadi fluktuasi nilai kadar garam di laut, bahkan akibat terjadinya hujan sekalipun. Hal ini karena di laut, *volume* air yang ada sangatlah besar, sehingga tidak terlalu berpengaruh pada nilai salinitas. Pada akuarium yang memiliki *volume* air lebih kecil, efek dari penguapan bisa jadi membuat nilai kadar garam meningkat tajam. Fluktuasi yang terjadi secara terus menerus pada nilai salinitas ini dapat menyebabkan biota yang dipelihara di akuarium menjadi *stress* dan akhirnya mati.

Salinitas pada akuarium dapat meningkat seiring terjadinya penguapan pada akuarium. Air akuades dapat digunakan untuk menggantikan *volume* air yang menguap ini baik secara manual, ataupun menggunakan sebuah sistem dengan sensor ketinggian air [2]. Selain menggunakan air akuades, juga dapat digunakan larutan *kalkwasser* yang sekaligus akan menambahkan kalsium karbonat ke dalam akuarium.

2.4. Bahan Aditif (Kalsium dan Alkalinitas / C*arbonate Hardness*)

Kalsium adalah salah satu elemen utama dalam kandungan air laut. Kalsium banyak digunakan dalam struktur biologis termasuk di antaranya pada pembentukan tulang dan cangkang pada beberapa jenis alga, *scleractinians* dan beberapa invertebrata. Karbonat di sisi lain berfungsi sebagai penyeimbang proses kalsifikasi, pembentukan tulang pada koral dan invertebrata lain [3].

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Kalsium karbonat pada air laut sangatlah jenuh. Penurunan kadar kalsium di laut sangatlah jarang terjadi meskipun digunakan pada proses kalsifikasi biota – biota laut. Pada akuarium laut, proses kalsifikasi biota di dalamnya akan sangat mempengaruhi kadar kalsium dan karbonat yang ada. Kadar kalsium dan karbonat yang ada bisa bervariasi karena *volume* air yang cenderung lebih kecil. Kepadatan kadar organik dan fosfat, juga dapat mengurangi kadar kalsium dan karbonat dalam akuarium dengan cepat [3]. Kadar kalsium pada air laut biasanya berada pada rentang 390 – 430 ppm. Sedangkan kadar karbonat (alkalinitas) berada pada rentang 2 – 3 meq/L (5,6 – 8,4 dKH), namun demikian, pada akuarium laut disarankan kadar karbonat dijaga pada *level* 4 – 6 meq/L (11,2 – 16,8 dKH) [4].

Penambahan bahan aditif atau biasa disebut dengan istilah *dosing* kalsium dan karbonat hanya boleh dilakukan apabila kadarnya telah diukur terlebih dahulu secara teratur untuk menjaga agar tidak terjadi terlalu banyak perubahan pada air akuarium. Sebelum melakukan penambahan bahan aditif ini, perlu diketahui terlebih dahulu rasio penurunan kadar kalsium dan karbonat pada akuarium. Rasio penurunan ini dapat diketahui dengan cara melakukan pengukuran setiap dua sampai empat hari sekali. Pada tahap ini penggunaan bahan aditif dihentikan terlebih dahulu [3]. Berikut merupakan rumus yang digunakan untuk penambahan aditif kalsium pada akuarium laut:

$$N_{Ca,t} = \frac{(Ca_{Ca,t=0} - Ca_{Ca,t=n})}{n} \quad (2.1)\ [3]$$

Rumus tersebut hanya dapat digunakan apabila kadar kalsium pada akuarium telah sesuai dengan standar (390 - 430 ppm). Apabila kadar kalsium pada akuarium lebih kecil dari standar, maka dapat digunakan rumus:

$$N_{Ca,diff} = \frac{(Ca_{Ca,required} - Ca_{Ca,aquarium})}{n} * V_{aquarium} \quad (2.2)\ [3]$$

Maka total pemberian kalsium setiap harinya adalah:

$$N_{Ca,total} = N_{Ca,t} + N_{Ca,diff} \quad (2.3)\ [3]$$

Persamaan (2.1) sampai (2.3) di atas menghasilkan jumlah kalsium yang perlu ditambahkan setiap harinya dalam satuan mg $Ca^{2+}.d^{-1}$ [3]. Selain menjaga kadar kalsium tetap stabil, maka penting juga untuk menjaga kadar karbonat. Setiap pembentukan satu

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

mol kalsium, maka diperlukan dua mol *bicarbonate* untuk membentuk satu mol kalsium karbonat.

Beberapa metode dapat digunakan untuk menggantikan dan menjaga kadar kalsium dan karbonat dalam akuarium laut. Salah satu metode yang sering digunakan adalah penggunaan larutan jenuh *Calcium Hydroxide* ($Ca(OH)_2$) atau biasa disebut dengan *Kalkwasser*. Kelebihan menggunakan larutan ini adalah bahwa larutan ini tidak hanya menjaga kadar kalsium, namun sekaligus sebagai penyangga dan menambahkan karbonat ke dalam akuarium. Larutan ini juga dapat digunakan sebagai satu – satunya cairan *top-up* atau sebagai pengganti air penguapan [2].

$Ca(OH)_2$ padat sangat sulit larut di dalam air. Kelarutannya hanyalah sekitar 1,26 gram dari $Ca(OH)_2$ padat dalam 1 liter akuades pada 20^o C yang dapat menambahkan konsentrasi ion Ca^{2+} sebanyak 682 mg pada 1 liter air [3]. Kelarutan senyawa ini juga menurun seiring dengan kenaikan suhu. Dari persamaan (2.3) di atas dan sifat kelarutan $Ca(OH)_2$ ini, maka dapat diperoleh suatu persamaan baru, yaitu:

$$V_{kalkwasser} = {}^{N_{Ca,total}\,(g)}/_{0,68} \qquad (2.4)\ [3]$$

Apabila larutan *kalkwasser* ini digunakan sebagai pengganti penguapan, maka konsentrasi kalsium karbonat yang ditambahkan terbatasi oleh *volume* air yang menguap setiap harinya. Hal ini bisa diatasi dengan memperbesar kemungkinan penguapan yang terjadi, salah satunya adalah dengan menaruh kipas di atas akuarium. Penggunaan tutup akuarium yang rapat dan tertutup juga harus dihindari, karena akan semakin mengurangi terjadinya penguapan.

Larutan *kalkwasser* dapat dibuat dengan memberikan 3,26 gram $Ca(OH)_2$ padat (lebih banyak 2 gram dari kelarutan $Ca(OH)_2$ dalam air) ke dalam air akuades, mencampurnya dengan hati – hati, dan menunggu sampai kelebihan $Ca(OH)_2$ mengendap di dasar. Yang digunakan hanyalah larutan beningnya saja. Larutan *kalkwasser* rata – rata memiliki pH diatas 12, sehingga sebaiknya larutan ini dituang ke akuarium secara perlahan. Pemberian larutan kalkwasser sebaiknya dilakukan setelah lampu akuarium padam, karena pada saat itu, proses respirasi cenderung akan meningkat, sehingga nilai pH di akuarium akan turun.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

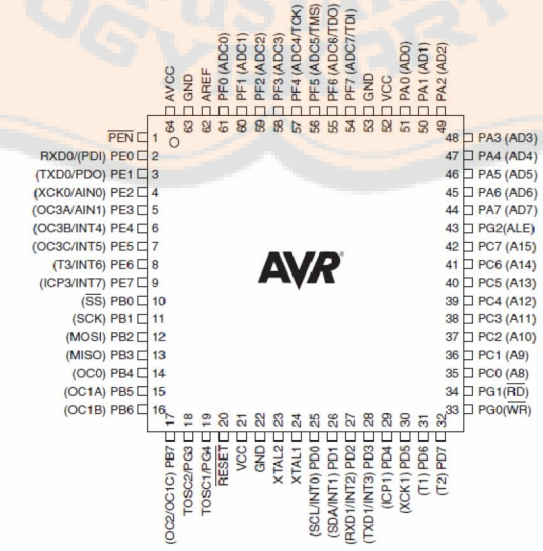
2.5. Mikrokontroler ATmega128

ATmega128 merupakan sebuah mikrokontroler yang memiliki arsitektur *RISC (Reduced Instruction Set Computing)* 8-bit. Satu perintah yang diberikan ke mikrokontroler akan dieksekusi selama 1 siklus *clock* [5].

Berikut merupakan fitur – fitur yang disediakan mikrokontroler ini:

1. Memori *flash* sebesar 128 *Kbytes* yang mampu dibaca dan ditulis secara bersamaan.
2. *EEPROM* sebesar 4 *Kbytes*.
3. *SRAM* sebesar 4 *Kbytes*.
4. 53 buah I/O pin yang terbagi dalam 7 *port* yaitu, *port* A, *port* B, *port* C, port D, *port* E, *port* F, dan *port* G.
5. 32x8 buah register pada CPU.
6. 2 buah *Timer / Counter 8-bit* yang dapat diatur aplikasinya termasuk mode pembanding.
7. 6 kanal *PWM* yang dapat diatur resolusinya dari *2 – 16-bit.*
8. Unit interupsi internal dan eksternal.
9. Port komunikasi serial USART.
10. *Two-wire serial interface.*
11. 8 kanal *Analog-to-Digital Converter* yang masing – masing memiliki resolusi sampai dengan 10-bit.

Gambar 2.1 merupakan gambar konfigurasi pin ATmega128 [5]:



Gambar 2. 1. Gambar Konfigurasi Pin ATmega128 [5]

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

2.5.1 Port Input / Output

ATmega128 mempunyai 53 pin I/O yang terbagi menjadi tujuh port dan mampu difungsikan sebagai masukan atau keluaran. Setiap pin I/O mempunyai tiga register yaitu DDRxn, PORTxn dan PINxn yang nilainya tergantung dari aplikasi pin I/O itu sendiri. Huruf "x" mewakili Port I/O tersebut, sedangkan huruf "n" mewakili nomor pin I/O yang dituju. Untuk mengatur sebuah pin I/O menjadi sebuah pin keluaran maka register DDR pada pin tersebut harus diberi logika tinggi, sedangkan agar berfungsi sebagai masukan, maka DDR pada pin tersebut diberi logika rendah.

Saat berfungsi sebagai sebuah pin masukan, maka register PINxn digunakan untuk membaca nilai pada pin tersebut. Sedangkan saat berfungsi sebagai sebuah pin keluaran, register PORTxn digunakan untuk mengatur nilai keluaran pin I/O tersebut.

2.5.2 Timer / Counter

ATmega128 memiliki empat buah modul *Timer / Counter* yang terdiri dari dua buah modul *Timer / Counter* 8-bit dan dua buah modul *Timer / Counter* 16-bit. Keempat modul *Timer / Counter* ini memiliki fungsi dan aplikasi yang berbeda – beda, dan masing – masing dapat diatur secara terpisah, tanpa mempengaruhi satu sama lain dan dapat digunakan sebagai sumber interupsi. Beberapa mode yang terdapat pada *Timer / Counter* antara lain *Normal Mode, Clear Timer on Compare Match (CTC) Mode, Fast PWM,* dan *Phase Correct PWM.*

Normal Mode adalah mode paling sederhana dari t*imer* ini. Pada mode ini cacahan yang dilakukan selalu naik *(incrementing). Timer* akan secara otomatis kembali ke nilai awal (0x00) setelah mencapai nilai maksimum.

Pada mode *Clear Timer on Compare,* register *OCRx* digunakan untuk memanipulasi resolusi pencacah. *Counter* akan kembali ke nilai awal saat nilai register *TCNTx* sama dengan nilai register *OCRx.* Secara umum, besarnya frekuensi sinyal yang dihasilkan oleh mode ini adalah sebesar:

$$f_{OCnA} = \frac{f_{clk_I/O}}{2.N.(1+OCRnA)} \quad (2.5)[5]$$

Mode *fast PWM* dapat digunakan sebagai penghasil sinyal PWM dengan frekuensi tinggi. Pencacah akan mencacah naik dari keadaan awal (0x00) sampai keadaan maksimum dan kemudian kembali ke keadaan awal. Pada mode *non-Inverting Compare*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Output, pin *OCx* akan bernilai nol *(cleared)* pada saat nilai register *OCRx* sama dengan nilai pencacah *TCNTx*, dan akan bernilai satu *(set)* pada saat pencacah kembali ke nilai awal (0x00). Sedangkan hal sebaliknya terjadi pada mode *Inverting Compare Output,* pin *OCx* akan bernilai satu saat *compare match,* dan bernilai nol saat pencacah kembali ke keadaan awal. Secara umum, besarnya frekuensi sinyal yang dihasilkan oleh mode ini adalah sebesar:

$$f_{OCnxPWM} = \frac{f_{clk_I/O}}{N.(1+TOP)} \quad (2.6)[5]$$

Mode *phase correct PWM* sedikit berbeda dari mode *fast PWM*. Pencacah akan mencacah naik, kemudian mencacah turun dan begitu seterusnya. Pada mode *non-Inverting Compare Output,* pin *OCx* akan bernilai nol *(cleared)* pada saat nilai register *OCRx* sama dengan nilai pencacah *TCNTx* saat mencacah naik, dan akan bernilai satu *(set)* pada saat nilai register *OCRx* sama dengan nilai pencacah *TCNTx* saat mencacah turun. Sedangkan hal sebaliknya terjadi pada mode *Inverting Compare Output,* pin *OCx* akan bernilai satu saat *compare match* cacah naik, dan bernilai nol saat *compare match* cacah turun. Secara umum, besarnya frekuensi sinyal yang dihasilkan oleh mode ini adalah sebesar:

$$f_{OCnxPCPWM} = \frac{f_{clk_I/O}}{2.N.TOP} \quad (2.7)[5]$$

2.5.2.1. *Timer / Counter 1 dan 3*

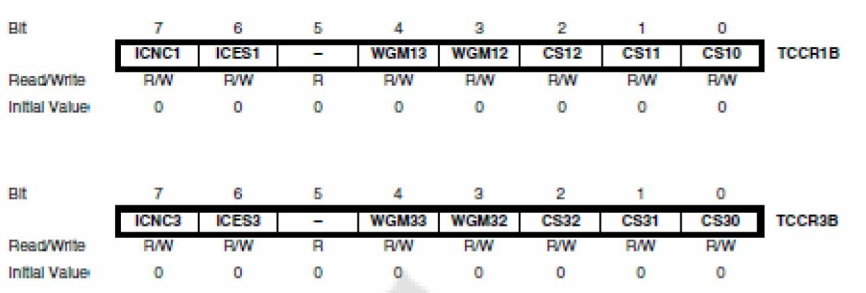
Timer / Counter 1 dan 3 merupakan modul *timer* 16-bit, sehingga sering digunakan sebagai pengatur jadwal perintah, sumber pembangkit sinyal dan pengukuran panjang sinyal [5]. Register yang digunakan untuk mengatur modul ini adalah register *TCCRnA, TCCRnB* dan *TCCRnC* dengan huruf n menyatakan nomor *Timer / Counter* yang digunakan.

Bit	7	6	5	4	3	2	1	0	
	COM1A1	COM1A0	COM1B1	COM1B0	COM1C1	COM1C0	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	COM3A1	COM3A0	COM3B1	COM3B0	COM3C1	COM3C0	WGM31	WGM30	TCCR3A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2. 2. Register *TCCRnA* [5]

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Bit	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	ICNC3	ICES3	–	WGM33	WGM32	CS32	CS31	CS30	TCCR3B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2. 3. Register *TCCRnB* [5]

Nilai *COMnA1, COMnA0, COMnB1, COMnB0, COMnC1, dan COMnC0* digunakan untuk mengatur keluaran pin *Output Compare (OCnA, OCnB, OCnC).* Pengaturan yang terjadi tergantung dari mode yang digunakan pada saat itu, seperti pada tabel 2.1 – tabel 2.3.

WGMn.3:0 berfungsi sebagai pemilih mode yang digunakan *Timer / Counter 1 / 3,* pengatur sinyal keluaran pencacah, dan sumber nilai maksimum *(TOP)* pencacah. Tabel 2.4 menunjukkan bagaimana hasil pengaturan Register *WGMn.3:0.*

CS1.2:0 digunakan untuk memilih sumber *clock* pencacah dan nilai *prescaler* yang digunakan.

Tabel 2. 1. *Compare Output Mode,* pada mode *non-PWM* [5]

COM1nA1 / COMnB1 / COMnC1	***COMnA0 / COMnB0 / COMnC0***	**Keterangan**
0	0	*OCnA / OCnB / OCnC* tidak aktif
0	1	Nilai *OCnA / OCnB / OCnC* berubah pada *Compare Match*
1	0	*OCnA / OCnB / OCnC* menjadi 0 *(low)* saat *Compare Match*
1	1	*OCnA / OCnB / OCnC* menjadi 1 *(high)* saat *Compare Match*

Tabel 2. 2. *Compare Output Mode,* pada mode *fast PWM* [5]

COM1nA1 / COMnB1 / COMnC1	***COMnA0 / COMnB0 / COMnC0***	**Keterangan**
0	0	*OCnA / OCnB / OCnC* tidak aktif
0	1	*WGM1.3:0* = 0b1111 (15): *OCnA* berubah nilainya pada *Compare Match*, *OCnB / OCnC* tidak aktif. Untuk seting *WGM1* selain itu, maka *OCnA / OCnB / OCnC* tidak aktif.
1	0	*OCnA / OCnB / OCnC* bernilai 0 saat *Compare Match* dan bernilai 1 saat *BOTTOM.*
1	1	*OCnA / OCnB / OCnC* bernilai 1 saat *Compare Match* dan bernilai 0 saat *BOTTOM.*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 2. 3. *Compare Output Mode,* pada mode *Phase Correct* dan *Phase and Frequency Correct PWM* [5]

COM1nA1 / COMnB1 / COMnC1	***COMnA0 / COMnB0 / COMnC0***	**Keterangan**
0	0	*OCnA / OCnB / OCnC* tidak aktif
0	1	*WGM1.3:0* = 0b1001 (9) atau 0b1011 (11): *OCnA* berubah nilainya pada *Compare Match*, *OCnB / OCnC* tidak aktif. Untuk seting *WGM1* selain itu, maka *OCnA / OCnB / OCnC* tidak aktif.
1	0	*OCnA / OCnB / OCnC* bernilai 0 saat *Compare Match* cacah naik dan 1 saat *Compare Match* cacah turun
1	1	*OCnA / OCnB / OCnC* bernilai 1 saat *Compare Match* cacah naik dan 0 saat *Compare Match* cacah turun

Tabel 2. 4. Mode Pembangkit Sinyal *Timer / Counter* [5]

Mode	***WGMn.3***	***WGMn.2 (CTC1)***	***WGMn.1 / PWM11***	***WGMn.0 / PWM10***	**Mode Operasi *Timer / Counter***	***TOP***
0	0	0	0	0	*Normal*	0xFFFF
1	0	0	0	1	*PWM, Phase Correct 8-bit*	0x00FF
2	0	0	1	0	*PWM, Phase Correct 9-bit*	0x01FF
3	0	0	1	1	*PWM, Phase Correct 10-bit*	0x03FF
4	0	1	0	0	*CTC*	*OCRnA*
5	0	1	0	1	*Fast PWM 8-bit*	0x00FF
6	0	1	1	0	*Fast PWM 9-bit*	0x01FF
7	0	1	1	1	*Fast PWM 10-bit*	0x03FF
8	1	0	0	0	*PWM, Phase and Frequency Correct*	*ICRn*
9	1	0	0	1	*PWM, Phase and Frequency Correct*	*OCRnA*
10	1	0	1	0	*PWM, Phase Correct*	*ICRn*
11	1	0	1	1	*PWM, Phase Correct*	*OCRnA*
12	1	1	0	0	*CTC*	*ICRn*
13	1	1	0	1	*Reserved*	-
14	1	1	1	0	*Fast PWM*	*ICRn*
15	1	1	1	1	*Fast PWM*	*OCRnA*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 2. 5. Nilai *clock* dan *prescaler* pada *Timer / Counter 1* [5]

CS0.2	*CS0.1*	*CS0.0*	**Keterangan**
0	0	0	*Timer / Counter* tidak aktif
0	0	1	*Clock* / 1, *Prescaler* = 1
0	1	0	*Clock* / 8, *Prescaler* = *8*
0	1	1	*Clock* / 64, *Prescaler* = 64
1	0	0	*Clock* / 256, *Prescaler* = 256
1	0	1	*Clock* / 1024, *Prescaler* = 1024
1	1	0	*External clock* pada pin *T1, falling edge*
1	1	1	*External clock* pada pin *T1, rising edge*

2.5.3 I^2C – Two-Wire Serial Communication

Two-wire serial communication (TWI) adalah salah satu fitur yang sering dipakai pada aplikasi mikrokontroler. *TWI* memungkinkan pengguna untuk terhubung dengan 128 perangkat yang berbeda dengan hanya menggunakan dua jalur data, *SCL* sebagai pengatur *clock* dan *SDA* sebagai jalur data utama [5].

Terdapat lima register yang digunakan untuk mengatur penggunaan *TWI,* yaitu *TWI Bit Rate Register (TWBR), TWI Control Register (TWCR), TWI Status Register (TWSR), TWI Data Register (TWDR), dan TWI (Slave) Address Register (TWAR).*

Register *TWBR* digunakan sebagai pengatur bit rate dari komunikasi serial. Nilai *TWBR* dapat dihitung dari rumus:

$$SCL\ frequency = \frac{CPU\ Clock\ frequency}{16+2(TWBR).4^{TWPS}} \quad (2.8)[5]$$

Register *TWCR* digunakan sebagai pengatur operasi *TWI.* Register ini terdiri dari 8-bit data seperti ditunjukkan gambar.

Bit	7	6	5	4	3	2	1	0	
	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	TWCR
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2. 4. Register *TWCR* [5]

Memberi logika *high* pada bit *TWSTA* menbuat perangkat menjadi *master device* pada jalur data. Perangkat akan mendeteksi keberadaan jalur data, apabila jalur data tersedia, maka perangkat akan menginisialisasi kondisi *START*. Apabila jalur data sedang digunakan oleh perangkat lain, maka perangkat akan menunggu sampai terdeteksi kondisi *STOP* kemudian menginisialisasi kondisi *START* dan mengambil alih jalur data.

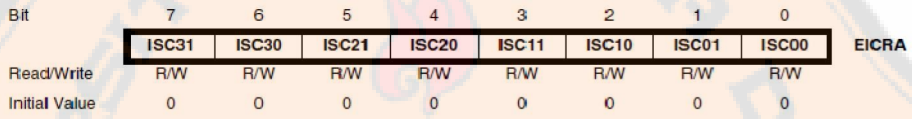
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Sedangkan bit TWSTO digunakan untuk menginisialisasi kondisi *STOP* pada jalur data. Apabila perangkat diatur sebagai *slave device*, maka bit ini dapat digunakan untuk memulihkan kondisi eror.

Bit *TWEN* digunakan untuk mengaktifkan antarmuka *TWI*. Saat *TWEN* bernilai 1, maka antarmuka *TWI* akan mengambil alih pin I/O *SDA* dan *SCL* dan menggunakannya sebagai jalur data.

2.5.4 External Interupt

Mikrokontroler *ATmega128* mempunyai delapan buah pin yang dapat digunakan sebagai *external interrupt.* Empat register utama digunakan sebagai pengatur kerja *external interupt.*

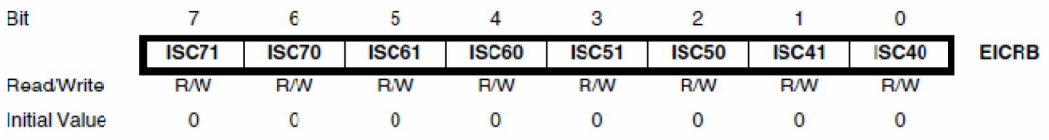


Gambar 2. 5. Register *EICRA* [5]

Register EICRA digunakan untuk mengatur kerja pin INT0 sampai INT3. Bit ISCn1 dan ISCn0 digunakan sebagai pengatur sinyal yang dapat digunakan sebagai perintah interupsi pada mikrokontroler, dengan huruf "n" menyatakan pin interupsi yang dituju.

Tabel 2. 6. Pengaturan Bit *ISCn1* dan *ISCn0* [5]

ISCn1	***ISCn0***	**Keterangan**
0	0	Logika rendah pada pin INTn menyatakan interupsi
0	1	*Reserved*
1	0	Perubahan logika tinggi ke rendah pada pin INTn menyatakan interupsi
1	1	Perubahan logika rendah ke tinggi pada pin INTn menyatakan interupsi



Gambar 2. 6. Register *EICRB* [5]

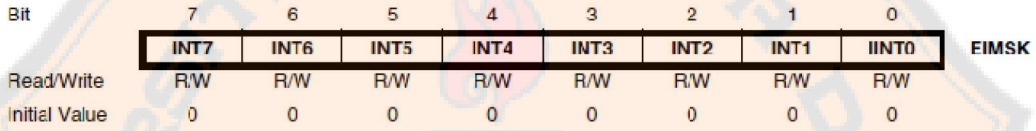
Register EICRB digunakan untuk mengatur kerja pin INT4 sampai INT7. Bit ISCn1 dan ISCn0 digunakan sebagai pengatur sinyal yang dapat digunakan sebagai perintah interupsi pada mikrokontroler, dengan huruf "n" menyatakan pin interupsi yang

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

dituju. Sinyal pada pin INTn akan diambil contohnya *(sampling)* sebelum dilakukan pengambilan keputusan interupsi. Sinyal yang panjangnya lebih besar dari sumber detak mikrokontroler akan digunakan sebagai sumber interupsi.

Tabel 2. 7. Pengaturan Bit *ISCn1* dan *ISCn0* [5]

ISCn1	***ISCn0***	**Keterangan**
0	0	Logika rendah pada pin INTn menyatakan interupsi
0	1	Setiap perubahan logika pada pin INTn menyatakan interupsi
1	0	Perubahan logika tinggi ke rendah yang kedua kalinya pada pin INTn menyatakan interupsi
1	1	Perubahan logika rendah ke tinggi yang kedua kalinya pada pada pin INTn menyatakan interupsi



Gambar 2. 7. Register *EIMSK* [5]

Register EIMSK digunakan untuk mengaktifkan fungsi interupsi pada pin INTn. Apabila bit INTn pada register ini bernilai satu, maka pin INT pada mikrokontroler akan berfungsi sebagai sumber interupsi program.

2.5.5 EEPROM (Electrical Erasable Programmable Read-Only Memory)

Mikrokontroler *ATmega128* mempunyai memori *EEPROM* sebesar 4 *kBytes*. Memori ini dapat dibaca dan ditulis melalui program dan data yang tersimpan tidak akan hilang walaupun mikrokontroler kehilangan catu daya. Alamat memori yang dituju oleh program sebelum membaca atau menulis data pada *EEPROM* ditunjukkan oleh register *EEARH* dan *EEARL*. Register ini adalah register 11-bit yang menyimpan alamat *EEPROM* dari alamat 0 sampai dengan alamat 4095 [5]. Register EEDR digunakan untuk membaca dan menulis data pada alamat yang ditunjukkan oleh register EEAR.

2.6. LCD Karakter 16x4

LCD karakter adalah perangkat yang mampu menampilkan karakter empat baris, dengan setiap baris 16 karakter [6]. Di dalam modul LCD karakter telah terpasang alat kontrol tersendiri, sehingga untuk menggunakannya hanya perlu mengikuti standar kontroler perangkat tersebut. Pada LCD karakter 16x4 ini terdapat 16 pin yang digunakan sebagai pengontrol kerja perangkat, seperti pada tabel 2.6.

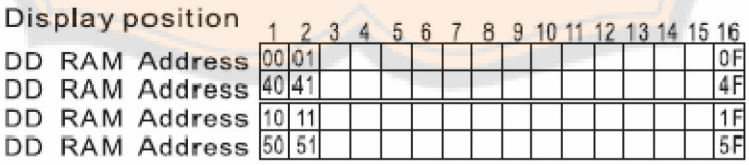
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Terdapat tiga *register* utama pada modul LCD karakter 16x4 yaitu r*egister DDRAM, register CGROM,* dan *register CGRAM. Register DDRAM* digunakan untuk menunjukkan ke alamat mana suatu data akan dikirimkan / ditampilkan. Dari gambar 2.6, maka untuk menampilkan karakter pada baris pertama kolom pertama, maka terlebih dahulu kita harus menunjuk ke alamat DDRAM 0x00.

Register CGROM digunakan untuk menampilkan karakter yang telah tersimpan di dalam modul LCD. Untuk menampilkan suatu karakter, maka hanya perlu menunjuk ke kode ASCII karakter yang dimaksud.

Tabel 2. 8. Tabel Konfigurasi pin LCD 16x4 [7]

Pin No.	Simbol	Keterangan
1	Vss	GND
2	Vdd	+3V atau +5V
3	Vo	Kontrol kecerahan
4	RS	Pemilih register
5	R/$\overline{W}$	Sinyal perintah baca / tulis
6	E	*Enable*
7	DB1	Jalur data
8	DB2	Jalur data
9	DB3	Jalur data
10	DB4	Jalur data
11	DB5	Jalur data
12	DB6	Jalur data
13	DB7	Jalur data
14	DB8	Jalur data
15	A/Vee	+4.2V untuk LED
16	K	*Power supply* (0v)



Gambar 2. 8. Isi alamat *DDRAM* [7]

Sedangkan *register CGRAM* digunakan untuk menampilkan karakter yang belum ada pada kode ASCII. Terdapat delapan *byte* data dari alamat 0x00 – 0x3f yang harus dikirimkan ke *register CGRAM* untuk menampilkan sebuah karakter yang dibuat sendiri.

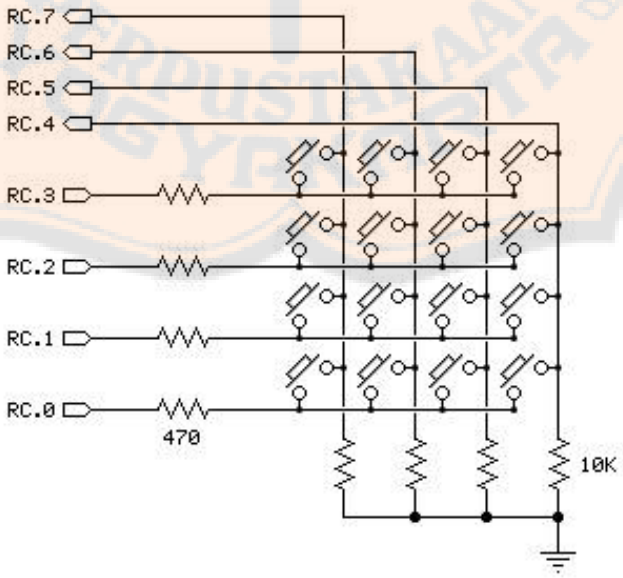
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Dari setiap *byte* data yang dikirimkan, hanya lima *bit* pertama yang akan digunakan. Hal ini karena setiap karakter mempunyai 5x8 *pixels*.

Gambar 2. 9. Kode ASCII dan karakter yang ditampilkan [7]

2.7. Keypad 4x4

Keypad 4x4 adalah perangkat yang tersusun dari *push button* yang membentuk matrik [6]. Perangkat ini bekerja dengan cara *scanning*. Empat kolom berfungsi sebagai keluaran, sedangkan empat barisnya sebagai masukan mikrokontroler. Berikut gambar rangkaian *keypad* 4x4.

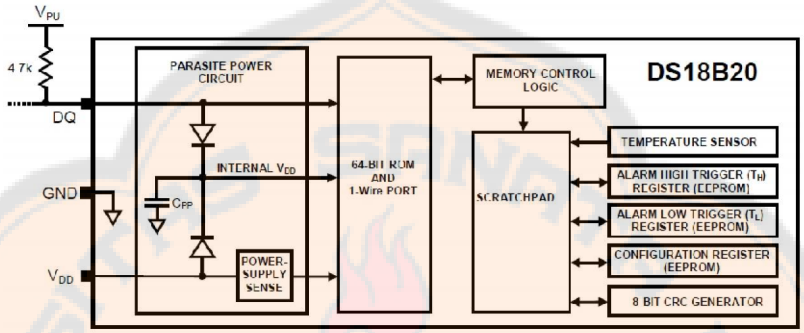


Gambar 2. 10. Rangkaian *keypad* matrik 4x4 [8]

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

2.8. Sensor Suhu DS18B20

Sensor suhu DS18B20 bekerja dengan sistem *one-wire communication* yang pertukaran data dari *master device* dan *slave device*-nya terjadi pada satu jalur data *(databus)*. Jalur data pada sensor DS18B20 memerlukan tahanan *pull-up* sekitar 5000 ohm untuk dapat bekerja, oleh karenanya pada keadaan *idle* jalur data bernilai 1 *(high)* [9].



Gambar 2. 11. Blok diagram sensor DS18B20 [9]

Keluaran sensor DS18B20 sudah berupa data digital, sehingga tidak diperlukan tambahan *ADC*. Pengguna dapat mengatur resolusi sensor dari 9, 10, 11, hingga 12-bit data. Pada resolusi 12-bit, dapat dihasilkan ketelitian hingga 0,0625 ^{o}C. Berikut merupakan tabel konversi data output sensor DS18B20 pada suhu yang diukur.

Tabel 2. 9. Tabel konversi data sensor DS18B20 [9]

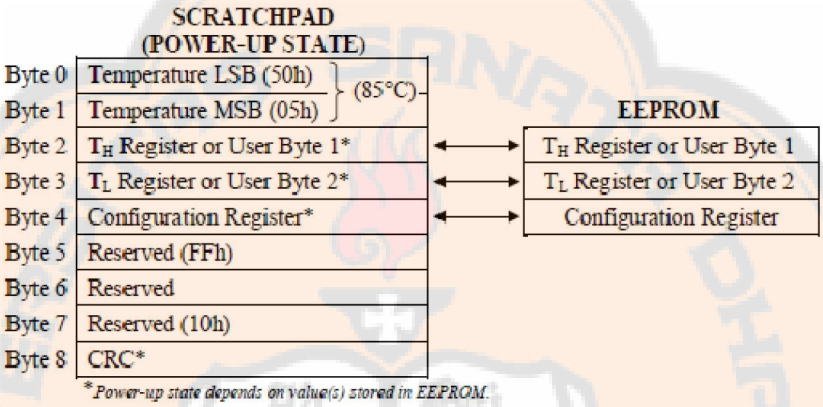
Suhu (^{o}C)	***Digital Output (binary)***	***Digital Output (hex)***
+125	0000 0111 1101 0000	07D0h
+85 *(default)*	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh

Arsitektur memori DS18B20 terlihat seperti pada gambar 2.12. *Byte* pertama dan kedua dari *scratchpad memory* berisi data hasil konversi suhu. Memori ini hanya bisa dibaca. Sedangkan register T_H dan T_L digunakan sebagai alarm. Memori di register ini bersifat tetap *(EEPROM)* oleh karenanya saat tidak ada catu ke sensor, data pada register ini tidak hilang. Saat hasil konversi suhu selesai dan suhu yang terbaca melebihi atau sama

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

dengan nilai T_H maka *alarm flag* akan bernilai satu (*set*). *Configuration register* berisi 2-bit data R1 dan R0 yang digunakan untuk mengatur resolusi sensor. Pada kondisi *default* sensor ini memiliki resolusi 12-bit.

Untuk mulai melakukan proses pertukaran data antara *master device* dengan sensor DS18B20, harus melewati 3 proses terlebih dahulu, yaitu inisialisasi, mengirim *ROM Command* (diikuti proses pertukaran data yang diharapkan), mengirim *Function Command* (diikuti proses pertukaran data yang diharapkan).



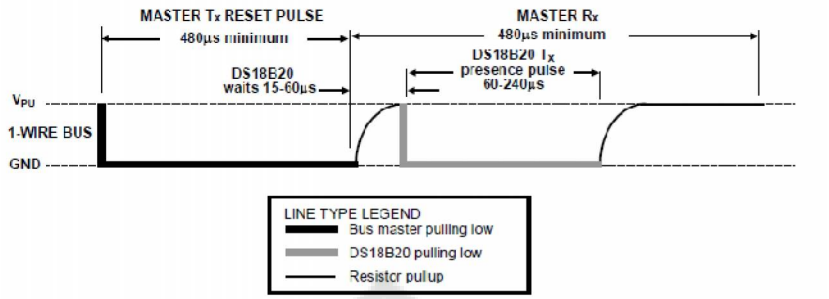
Gambar 2. 12. *Memory mapping* sensor suhu DS18B20 [9]

Tabel 2. 10. Pengaturan *Configuration Register* sensor DS18B20 [9]

R1	R0	Resolusi (bits)	Waktu konversi maksimum	
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})

Pada proses inisialisasi, *master device* akan mengirimkan sinyal *reset* ke semua *slave device* yang ada. *Slave device* akan merespon dengan mengirimkan sinyal kehadiran ke *master device.* Sinyal kehadiran berfungsi agar *master device* mengenali *slave device* siap untuk menerima data perintah selanjutnya [9]. Sinyal r*eset* adalah sinyal bernilai nol yang diberikan oleh *master device* ke perangkat DS18B20 setidaknya selama 480 µs. Sedangkan sinyal kehadiran adalah sinyal bernilai nol yang diberikan oleh perangkat DS18B20 selama 60-240 µs. Di antara sinyal *reset* dan sinyal kehadiran ini ada jeda sekitar 15-60 µs. Berikut merupakan *timing diagram* antara sinyal *reset* dan sinyal kehadiran antara *master device* dan perangkat DS18B20.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Gambar 2. 13. Timing diagram sinyal *reset* dan sinyal kehadiran [9]

Setelah sinyal kehadiran dari *slave device* diterima oleh *master device* maka *master device* dapat mengirimkan *ROM Command* [9]. *ROM Command* berfungsi untuk mengenali dan mencari *slave device* yang sesuai (pada kasus tertentu, satu *master device* mungkin terhubung ke beberapa *slave device*). Pada sistem yang hanya mempunyai satu *slave device* (dalam hal ini adalah DS18B20), maka *master device* dapat mengirimkan perintah *Skip ROM (0xCCh)*. Dengan cara ini *slave device* dapat langsung diberi perintah selanjutnya *(Function Command)*.

Function Command digunakan untuk memberikan perintah yang harus dilakukan *slave device* (DS18B20). *Master device* dapat memberikan perintah konversi suhu, tulis dan baca data di memori *scratchpad*, dan lain sebagainya. Berikut merupakan perintah – perintah data yang bisa dikirimkan ke perangkat DS18B20.

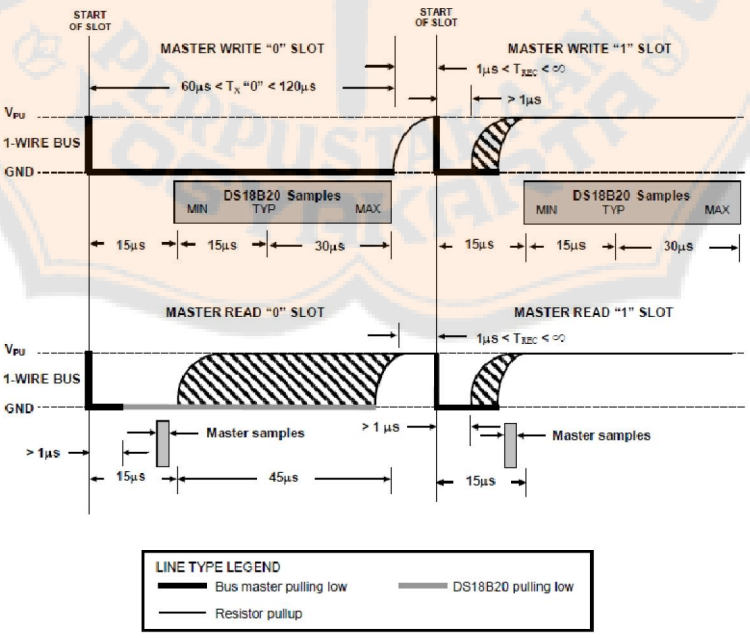
Antara pengiriman data dan pengambilan data terdapat jeda waktu tertentu. Pada saat pengiriman data, waktu ini disebut dengan *write time slot.* Sedangkan pada saat pengambilan data, waktu ini disebut dengan *read time slot.* Karena jalur yang digunakan sama, maka antara *write time slot,* dan *read time slot* harus dilakukan secara bergantian berdasarkan protokol tertentu.

Terdapat dua jenis *write time slot* pada IC DS18B20, yaitu *write "1" time slot* dan *write "0" time slot*. Setiap *write time slot* harus berdurasi setidaknya 60 µs dengan waktu pemulihan setidaknya 1µs dan harus diawali dengan *master device* memberikan logika nol selama kurang dari 15 µs. Pada *write "1" time slot* setelah *master device* memberikan logika nol, maka master memberikan logika satu selama setidaknya 60 µs. Sedangkan pada *write "0" time slot,* setelah memberikan logika nol, master hanya perlu menahan keadaan ini selama 60 µs. Gambar pewaktuan *write time slot* dan *read time slot* dapat dilihat pada gambar 2.14.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 2. 11. *Function Command* pada sensor suhu DS18B20 [9]

Perintah	Penjelasan	Data yang dikirimkan	Aktivitas jalur data
Perintah konversi suhu			
Convert T	Lakukan proses konversi suhu	0x44h	DS18B20 memberikan status konversi data suhu ke *master device*
Perintah pengaturan memori DS18B20			
Read Scratchpad	Baca semua isi memori *scratchpad* termasuk *CRC*	0x0BEh	DS18B20 mengirim 9 *byte* data ke *master device*
Write Scratchpad	Tulis data ke *scratchpad byte* 2, 3, 4 (register T_H, T_L, dan konfigurasi)	0x4Eh	*Master device* mengirimkan 3 *byte* data ke DS18B20
Copy Scratchpad	Pindahkan nilai register T_H, T_L, dan konfigurasi dari *scratchpad* ke *EEPROM*	0x48h	Tidak ada aktivitas di jalur data
Recall E^2	Kembalikan nilai register T_H, T_L, dan konfigurasi dari *EEPROM* ke *scratchpad*	0x0B8h	DS18B20 memberikan status *recall* ke *master device*
Read Power Supply	Memberikan sinyal mode *supply* DS18B20 ke *masterdevice*	0x0B4h	DS18B20 memberikan status *supply* ke *master device*

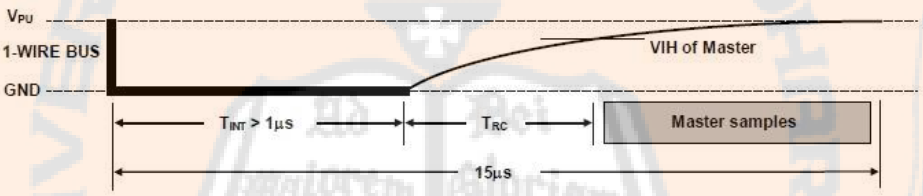


Gambar 2. 14. *Timing diagram write / read time slot* [9]

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Sensor DS18B20 hanya dapat memberikan data kepada *master device* apabila *master device* memberikan *read time slot.* Oleh karenanya *master device* harus memberikan *read time slot* setelah memberikan perintah R*ead scratchpad (0x0BEh)* atau perintah *Read power supply (0x0B4h)*. Sebagai tambahan, *master device* juga harus memberikan *read time slot* setelah memberikan perintah *Convert T (0x44h)* atau *Recal* E^2 *(0x0B8h)* untuk mengetahui status proses yang dilakukan [9].

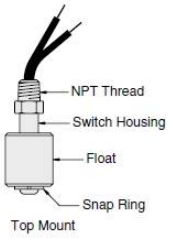
Read time slot mempunyai durasi selama 60 µs dengan waktu pemulihan minimal selama 1µs di antara setiap *read time slot*. *Read time slot* dimulai dengan *master* memberikan logika nol selama minimal 1µs kemudian berubah ke mode penerima. Sensor kemudian akan mulai mengirimkan data nol dan satu dimulai dari *least significant byte (LSB)*. Data yang diterima *master* valid apabila diambil pada rentang waktu 1-15 µs setelah master memberikan *read time slot* [9].



Gambar 2. 15. *Timing diagram read time slot* DS18B20 [9]

2.9. Float Switch – Water Level Sensor

Model paling sederhana dari sensor ketinggian air adalah *float switch* [10]. Sensor ini berupa sebuah tabung yang di dalamnya terdapat sebuah *reed switch* dan di luar tabung tersebut, tergantung sebuah bandul yang terdapat magnet di dalamnya. *Reed switch* di dalam tabung utama berubah kondisinya, tergantung dari posisi bandul tersebut. Apabila bandul berada pada posisi atas, maka keadaan *reed switch* di dalamnya adalah hubung buka. Sedangkan apabila bandul berubah posisinya, menuju ke bawah, maka *reed switch* di dalamnya berada dalam keadaan hubung singkat. Berikut gambar ilustrasi dari *float switch.*

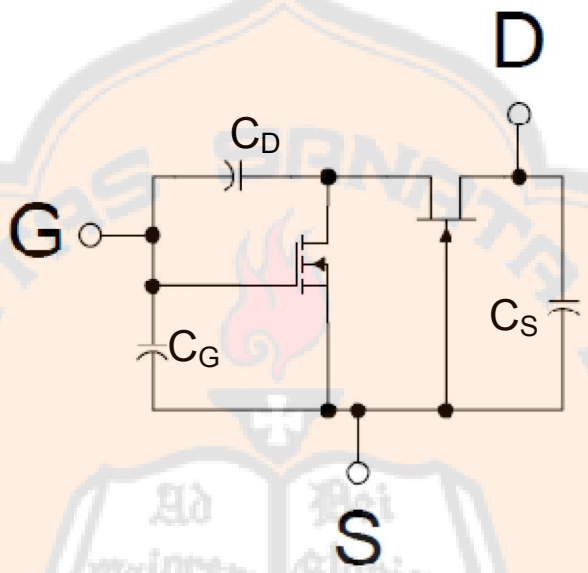


Gambar 2. 16. *Float switch* [10]

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

2.10. MOSFET

MOSFET adalah singkatan dari *Metal Oxide Semiconductor Field Effect Transistor*. Pada model rangkaian *MOSFET,* terdapat tiga buah kapasitor pada masing – masing kakinya. Kecepatan waktu *switching* pada *MOSFET* sangat dipengaruhi oleh seberapa lama waktu *charging* dan *discharging* arus pada ketiga kapasitor ini.



Gambar 2. 17 Model komponen *MOSFET* [11]

Seperti halnya sebuah transistor *BJT*, *MOSFET* juga memerlukan tegangan bias agar dapat bekerja. Berbeda dengan transistor *BJT* yang bekerja berdasarkan arus bias pada kaki *base, MOSFET* bekerja berdasarkan tegangan pada kaki *gate* terhadap kaki *source*-nya. Tegangan antara kaki *gate* dan *source* (V_{GS}) yang lebih besar dari tegangan ambang ($V_{GS(TH)}$) suatu *MOSFET* akan membuat arus dapat mengalir dari kaki *drain* ke kaki *source* (I_{DS}). Arus I_{DS} yang dapat mengalir akan semakin besar seiring dengan bertambahnya tegangan V_{GS} [12]. Berikut persamaan yang digunakan dalam mendesain suatu rangkaian *MOSFET:*

$$I_D = k(V_{GS} - V_{GS(TH)})^2 \qquad (2.9)\ [12]$$

Salah satu keunggulan penggunaan *MOSFET* adalah waktu *switching (turn-on time* dan *turn-off time)* yang relatif cepat. Waktu *switching* ini dipengaruhi oleh salah satu karakteristik *MOSFET* yaitu *gate charge* [13]. *Gate charge* adalah muatan yang diperlukan oleh *MOSFET* untuk dapat berubah keadaan dari kondisi *off* ke kondisi *on* dengan:

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

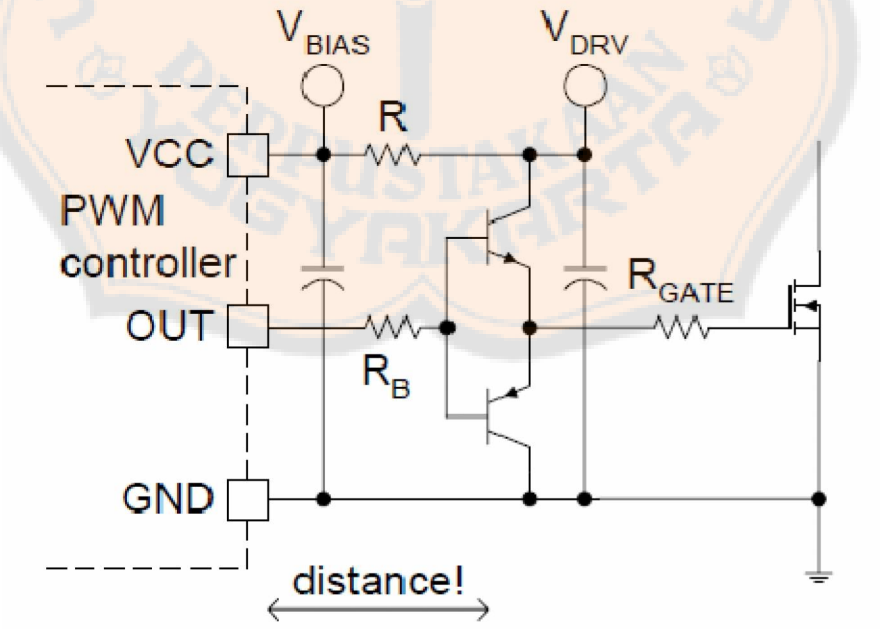
$$Q_{Gate,total} = I_G . t \qquad (2.10)\ [13]$$

t adalah waktu yang diperlukan *MOSFET* untuk berubah dari keadaan *off* ke keadaan *on*. Dari persamaan (2.10) maka seiring bertambahnya I_G waktu t akan semakin kecil, sehingga waktu *switching* juga akan semakin kecil.

2.10.1 Bipolar Totem-pole MOSFET's driver

Rangkaian ini digunakan untuk menyuplai kebutuhan arus pada MOSFET. Keunggulan dari rangkaian ini adalah bahwa rangkaian ini dapat mengurangi kehilangan daya dan lecutan arus yang besar pada *MOSFET* apabila dihubungkan secara langsung dengan rangkaian kontrol. Selain itu Rangkaian ini juga dapat mempercepat waktu *switching MOSFET (turn-on time* dan *turn-off time).*

Pada saat sinyal kontrol berada pada logika tinggi maka transistor NPN akan berada pada kondisi *On,* sehingga arus mengalir menuju kaki *Gate MOSFET* dan dengan demikian membuat *MOSFET* berada pada kondisi *On.* Sedangkan pada saat sinyal kontrol berada pada logika rendah, maka transistor NPN akan *Off* sedangkan transistor PNP akan *On,* sehingga arus *discharging* akan langsung menuju ke *ground* rangkaian. Dengan demikian arus *charging* dan *discharging MOSFET* dapat diatur dengan cara mengatur *dc biasing* transistor NPN dan PNP.



Gambar 2. 18. Gambar rangkaian *Bipolar totem-pole driver* [11]

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

2.11. BJT

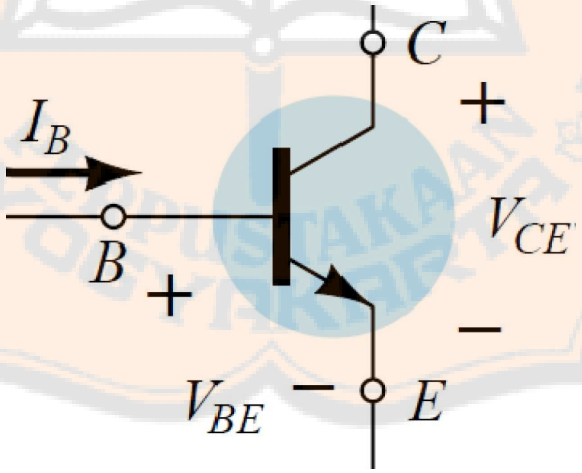
BJT adalah singkatan dari *Bi-polar Junction Transistor. Transistor* ini mempunyai banyak sekali fungsi, salah satunya adalah untuk kebutuhan *switching.* Terdapat dua jenis *BJT,* yaitu jenis *NPN* dan *PNP* namun keduanya mempunyai karakteristik yang sama (hanya berbeda dalam memberikan tegangan prasikap).

BJT adalah perangkat yang bekerja berdasarkan arus prasikap yang diberikan. Pada transistor *NPN,* saat tegangan kaki base terhadap kaki emiter lebih besar dari atau sama dengan 0,7 volt (untuk jenis silicon) maka transistor akan berada pada kondisi *on.* Pada kondisi *on*, arus dapat mengalir dari kaki *collector* ke kaki *emitter* yang nilainya sebesar:

$$I_C = h_{FE} * I_B \quad (2.11)[12]$$

I_B adalah arus yang mengalir pada kaki *base* menuju kaki *emitter*, dengan demikian arus yang mengalir di kaki *emitter* adalah gabungan antara arus *collector* dan arus *base*.

BJT akan mengalami saturasi apabila arus yang mengalir pada kaki collector *(I_C)* lebih besar dari atau sama dengan hasil perkalian arus I_B dengan *hfe (base-collector reverse biased, base-emitter forward biased)* [12]. Pada kondisi saturasi, tegangan V_{CE} transistor adalah sebesar nol volt.



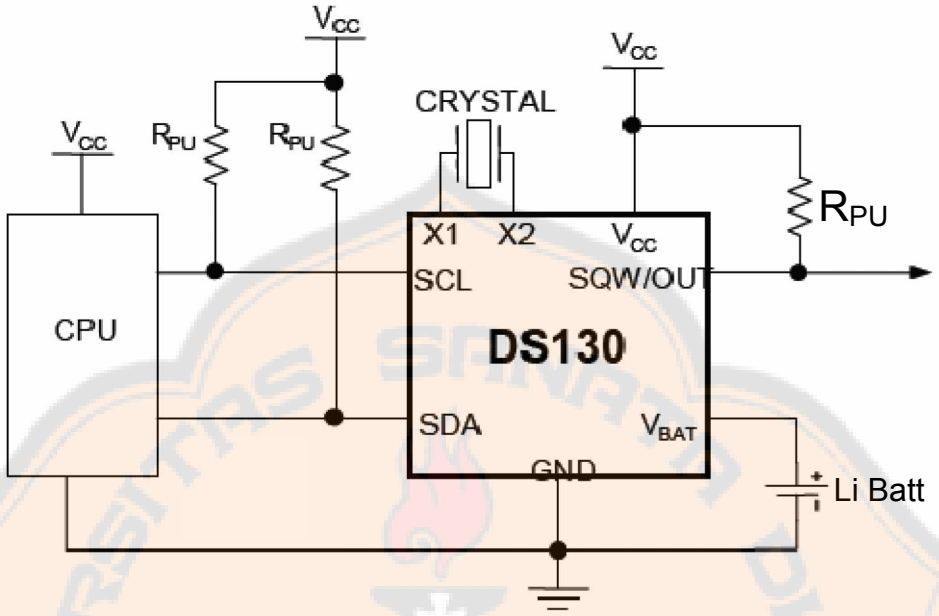
Gambar 2. 19. Gambar konfigurasi kaki *BJT* [12]

2.12. IC DS1307 - Real Time Clock (RTC)

IC DS1307 adalah sebuah *IC RTC* yang dapat digunakan untuk menyimpan waktu. Perangkat ini mampu menyimpan data waktu, dari mulai detik, menit, jam, hari, hingga tanggal, bulan, dan tahun. *IC DS1307* bekerja dengan menggunakan komunikasi

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

serial I^2C. Ketika catu utama tidak aktif, maka *IC* ini secara otomatis akan berpindah ke catuan dari baterai 3,2 V.



Gambar 2. 20. Rangkaian umum dari *IC DS1307* [14]

Semua data yang diterima dari *IC DS1307* sudah berupa data *Binary Coded Decimal (BCD)*. Pertukaran data menggunakan antarmuka I^2C, yang setiap memulai pertukaran data, *master device* harus menginisialisasi keadaan *START* dan diakhiri dengan keadaan *STOP*. Keadaan *START* terjadi apabila pin *SDA* berubah dari logika satu ke logika nol saat pin *SCL* berada pada logika satu. Sedangkan keadaan *STOP* terjadi saat pin *SDA* berubah dari logika nol ke logika satu saat pin *SCL* berada pada logika satu. Sedangkan pertukaran data terjadi pada saat pin *SCL* berada pada logika nol.

Memori *IC DS1307* terdiri dari dua register utama, yaitu *Timekeeper Register* dan *Control Register*. *Timekeeper Register* berisi data – data pewaktuan, mulai dari detik, menit, jam, tanggal, bulan, tahun, hingga hari. Sedangkan *Control Register* berisi bit untuk mengatur keluaran pin *SQW/OUT*. *S*aat *Square Wave Output* tidak aktif, bila bit *Out* bernilai satu, maka keluaran pin *SQW/OUT* akan bernilai satu, sedangkan apabila bit *Out* bernilai nol, maka keluaran pin *SQW/OUT* juga bernilai nol. Bit *SQWE* berfungsi untuk mengaktifkan *Square Wave Output*. Apabila bit ini bernilai satu, maka *Square Wave Output* akan aktif. Sedangkan nilai frekuensi yang dihasilkan tergantung dari kombinasi bit RS1 dan RS0.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 2. 12. Memori pada *IC DS1307* [14]

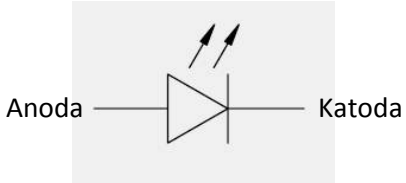
Alamat	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Fungsi	Rentang
00h	CH	10 detik			detik				Detik	00-59
01h	0	10 menit			menit				Menit	00-59
02h		12	10 jam	10 jam	jam				Jam	1-12 +AM/PM 00-23
		24	AM / PM							
03h	0	0	0	0	0	hari			Hari	01-07
04h	0	0	10 tanggal		tanggal				Tanggal	01-31
05h				10 bulan	bulan				Bulan	01-12
06h	10 tahun				tahun				Tahun	00-99
07h	Out	0	0	SQWE	0	0	RS1	RS0	Kontrol	-
08h-3Fh									RAM 56x8	00h-FFh

Tabel 2. 13. Pengaturan dan keluaran pin SQW/OUT [14]

RS1	RS0	Frekuensi SQW/OUT	SQWE	OUT
0	0	1 Hz	1	X
0	1	4096 Hz	1	X
1	0	8,192 kHz	1	X
1	1	32,768 kHz	1	X
X	X	0	0	0
X	X	1	0	1

2.13. High Power LED

High power LED adalah sebuah semikonduktor yang dapat memancarkan cahaya tertentu (monokromatik) ketika diberi tegangan bias maju. Dikatakan *high power* karena jenis *LED* ini mempunyai rating daya yang lebih besar dari *LED* biasanya.

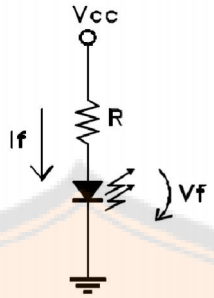


Gambar 2. 21. Simbol *LED*

Karena pada dasarnya *high power LED* adalah sebuah dioda, maka perangkat ini juga memiliki sifat seperti dioda. Arus yang melewati sebuah *LED* harus sesuai dengan

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

spesifikasi yang dimiliki perangkat tersebut. Untuk membatasi arus yang melewati *LED,* biasanya digunakan tambahan hambatan pada rangkaiannya.



Gambar 2. 22. Rangkaian sederhana *LED* [15]

Arus yang mengalir pada *LED* dibatasi oleh nilai hambatan R, sehingga diperoleh sebuah persamaan:

$$R = \frac{V_{cc} - V_f}{I_f} \quad (2.12)\ [15]$$

Pada aplikasinya, sebuah rangkaian seri dapat memiliki lebih dari satu buah *LED*. Dari persamaan (2.12) dapat diambil sebuah rumus baru, yaitu:

$$R = \frac{V_{cc} - n * V_f}{I_f} \quad (2.13)\ [15]$$

dengan *n* merupakan jumlah *LED* dalam satu rangkaian seri.

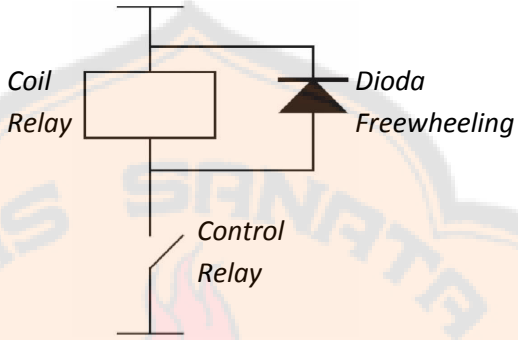
Tingkat intensitas cahaya yang dihasilkan *LED* dapat diatur dengan menggunakan metode pensaklaran. *Pulse width modulation* adalah teknik yang sering digunakan untuk metode pensaklaran *LED*. Dengan mengubah - ubah *duty cycle* dari sinyal yang digunakan, maka intensitas cahaya yang dihasilkan *LED* juga akan berubah. Semakin besar *duty cycle*-nya, maka intensitas cahaya yang dihasilkan akan semakin tinggi.

2.14. Relay

Relay adalah perangkat elektromekanik yang digunakan untuk menghubungkan atau memutus suatu rangkaian elektronik karena mendapat suatu sinyal dari perangkat pengontrolnya [16]. Terdapat dua jenis *relay, relay* elektromagnetik dan *solid state relay. Relay* elektromagnetik bekerja karena adanya gaya magnet yang disebabkan arus listrik melewati lilitannya. Arus yang melewati lilitan *relay* akan menyebabkan gaya magnetik pada bagian inti sehingga cukup kuat untuk menarik kontak relay tersebut. *Solid state relay* pada prinsipnya adalah sebuah *TRIAC.*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Lilitan pada *relay* pada prinsipnya adalah sebuah komponen induktor, sehingga diperlukan sebuah rangkaian sebagai pengaman kondisi *transient.* Sebuah dioda yang dihubungkan secara paralel dengan lilitan *relay* dapat digunakan sebagai pengaman rangkaian *relay.* Dioda pengaman ini disebut dioda *freewheeling.* Gambar 2.23 menunjukkan rangkaian pengaman kondisi *transient* pada *relay.*



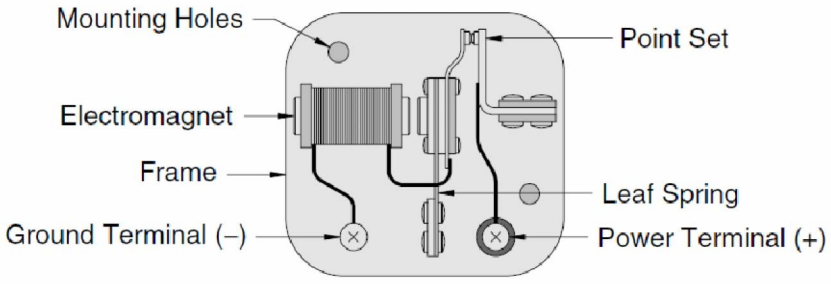
Gambar 2. 23. Rangkaian pengaman lilitan *relay* [13]

Arus yang melewati dioda *freewheeling* dibatasi oleh nilai hambatan dalam lilitan *relay,* yaitu sebesar:

$$I_{D,freewheeling} = \frac{V_{cc}}{R_{coil}} \tag{2.14}$$

2.15. Buzzer

Buzzer adalah perangkat elektronika yang mampu menghasilkan bunyi satu nada karena adanya gaya elektromagnetik. Saat diberi tegangan, arus akan mengalir ke bagian gulungan *buzzer,* sehingga menarik bagian penggetarnya. Disaat yang bersamaan, arus yang menuju ke gulungan terputus, sehingga gulungan kehilangan gaya elektromagnetisnya, dan penggetar kembali ke keadaan semula. Kondisi ini terjadi terus menerus, sehingga menghasilkan suatu bunyi. *Buzzer* biasa digunakan sebagai penanda terjadinya kondisi tertentu, misalnya kondisi sistem yang eror, sehingga operator dapat dengan segera menangani masalah yang terjadi pada sistem.



Gambar 2. 24. *Buzzer* [10]

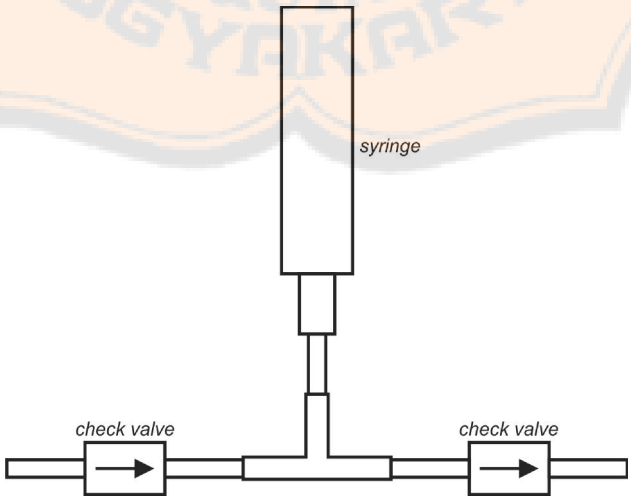
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

2.16. Syringe Pump

Syringe pump adalah pompa yang bekerja dengan prinsip vakum. Pompa ini biasa digunakan pada bidang kedokteran sebagai pompa obat – obatan dalam bentuk cairan yang diberikan secara terus menerus menurut dosis tertentu. Pompa ini digunakan karena mampu memompa cairan dengan dosis yang akurat. Pada selang masukan dan keluaran pompa ini terdapat sebuah *check valve* sehingga cairan di dalamnya hanya dapat mengalir ke satu arah saja. Berikut gambar dari *syring pump:*

Gambar 2. 25. *Syringe pump* [17]

Cairan yang dipompa mengalir melalui selang dari tempat penampungan cairan, menuju tempat tujuan cairan. Saat pompa menarik, cairan akan tertarik menuju alat suntik yang ada. Saat pompa mendorong, cairan akan terdorong menuju tempat tujuan cairan.



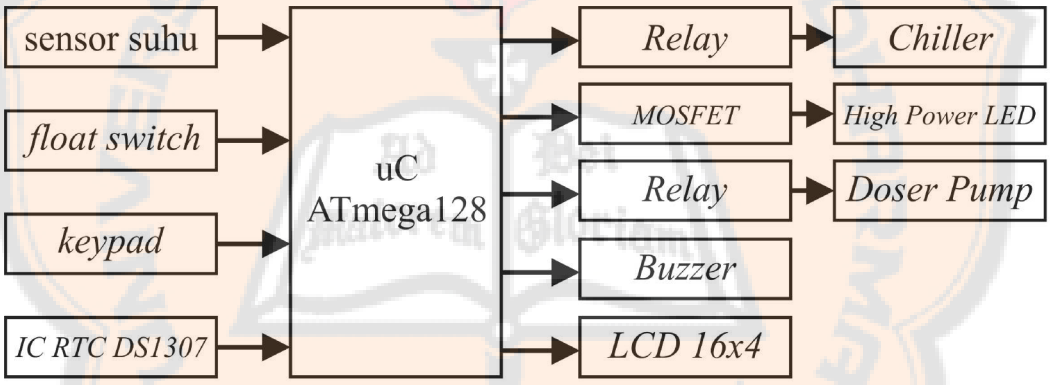
Gambar 2. 26. Skema *syringe pump*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

BAB III
RANCANGAN PENELITIAN

Perancangan sistem kontrol akuarium laut ini dibagi menjadi dua bagian utama, yaitu:

1. Perancangan *hardware* yang terdiri dari mikrokontroler dan perangkat pendukung, sensor suhu DS18B20, IC RTC DS1307, *keypad, float switch, high power LED,* dan modul LCD. Gambar 3.1 menunjukkan blok diagram sistem yang akan dibuat.
2. Perancangan *software* yang terdiri dari program utama, dan subrutin – subrutinnya seperti subrutin sensor suhu, IC RTC, dan modul LCD.



Gambar 3. 1 Blok diagram sistem yang akan dibuat

Berikut merupakan keterangan cara kerja sistem yang ditunjukkan gambar 3.1:

1. Sensor suhu DS18B20 akan mendeteksi suhu akuarium, dan mengirimkan data suhu kepada sistem. Data tersebut kemudian akan diolah oleh mikrokontroler untuk mengaktifkan atau mematikan relay *chiller* dan *buzzer*.
2. Sensor *float switch* digunakan untuk mendeteksi ketinggian air pada ruang filter terakhir akuarium.
3. *Keypad* matrik 4x4 digunakan untuk memberikan masukan berupa pengaturan dari pengguna. Pengaturan ini memungkinkan pengguna untuk mengubah konfigurasi dan mode – mode pengendalian yang dilakukan sistem.
4. *IC RTC DS1307* digunakan sebagai penjaga waktu mikrokontroler, sehingga semua kontrol pencahayaan dan penambahan bahan aditif dapat dilakukan secara *real-time.*

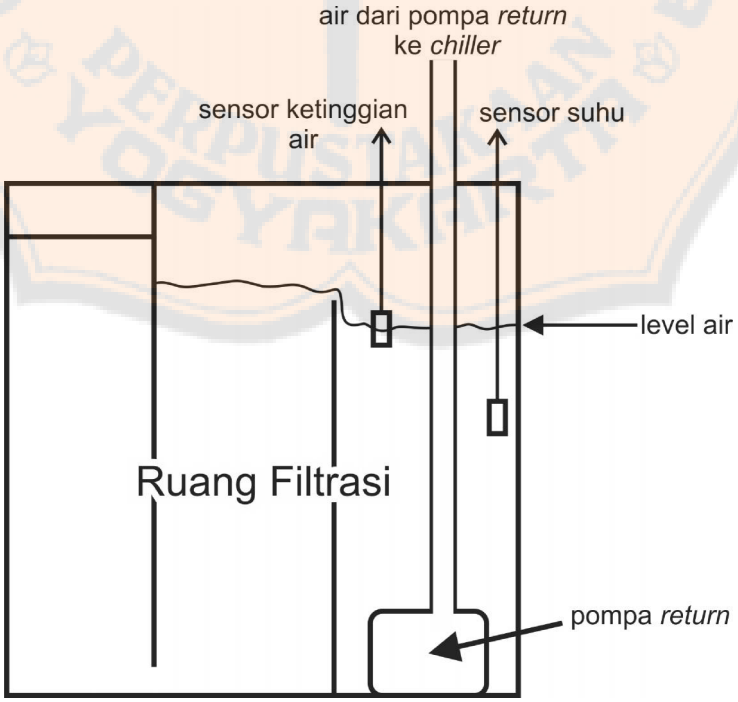
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

5. Khusus untuk kontrol penambahan bahan aditif, pengguna dapat dengan leluasa memilih mode yang digunakan. Terdapat dua mode utama pada subsistem ini, yaitu mode *float switch* dan mode *timer*. Mode *timer* bekerja berdasarkan waktu yang telah diatur oleh pengguna. Banyaknya dosis bahan aditif yang ditambahkan dapat diatur oleh pengguna dengan kelipatan 10 ml. Mode *float switch* bekerja berdasarkan kondisi ketinggian air pada ruang filter akuarium yang terakhir. Mode ini juga bisa digunakan sebagai pengaman pompa filter ketika dilakukan penggantian air pada akuarium. Tersedia empat kanal subsistem penambahan bahan aditif, sehingga pengguna dapat mengontrol empat *dosing pump* sekaligus tanpa saling mempengaruhi satu sama lain.
6. Subsistem pencahayaan mempunyai enam kanal yang dapat diatur secara terpisah dan tidak mempengaruhi satu sama lain.

3.1. Perancangan Perangkat Keras

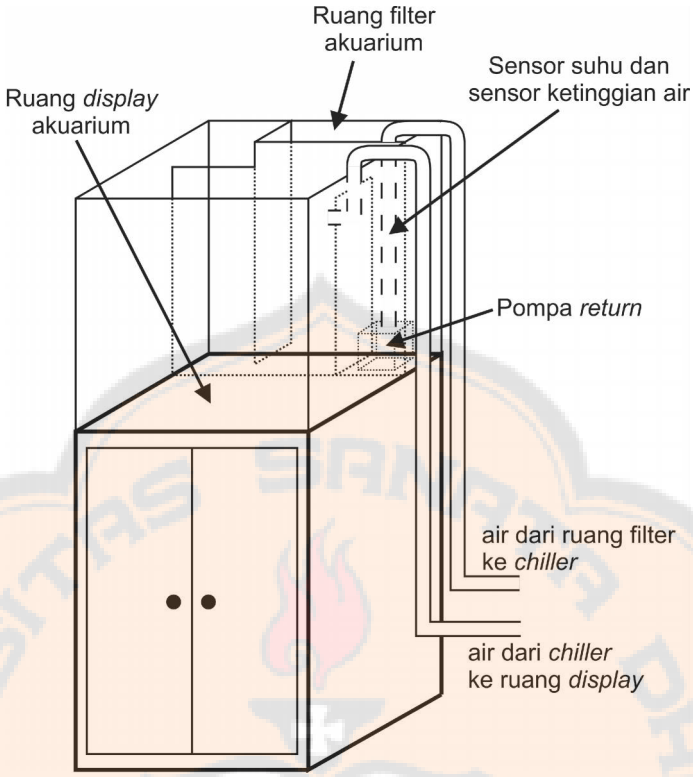
3.1.1 Perancangan Akuarium dan Sistem Kontrol

Akuarium yang digunakan mempunyai dimensi 40cm x 35cm x 40cm (panjang x lebar x tinggi). Ruang filtrasi berada di belakang ruang *display* utama akuarium. Sensor ketinggian air dan sensor suhu berada pada ruang filter yang terakhir. Berikut merupakan gambar sketsa akuarium dan tata letak sensor yang akan digunakan.



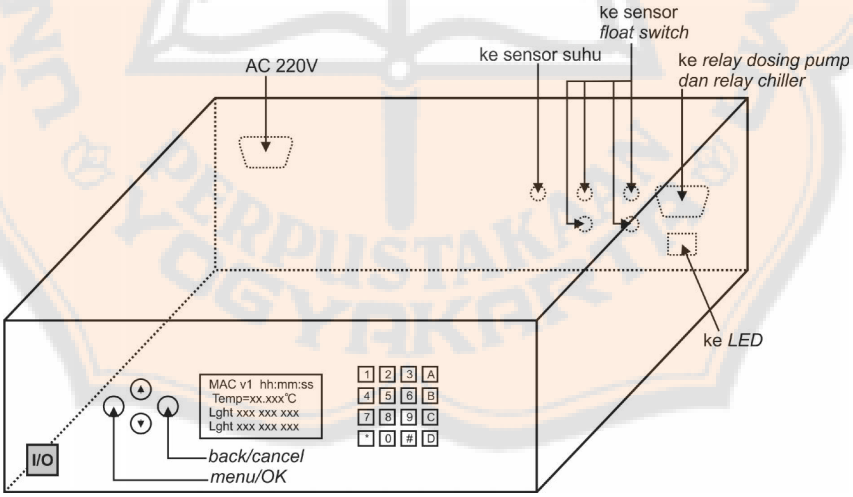
Gambar 3. 2. Desain dan tata letak sensor ruang filtrasi akuarium

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Gambar 3. 3. Desain dan tata letak sensor akuarium

Desain sistem kontrol yang akan dibuat ditunjukkan oleh gambar 3.4.



Gambar 3. 4. Desain sistem kontrol

3.1.2 Rangkaian LCD

LCD yang digunakan pada perancangan ini adalah LCD 16x4 yang di dalamnya telah terdapat sebuah kontroler. Tipe kontroler yang terdapat pada modul LCD ini adalah HD44780. LCD ini bekerja dengan komunikasi paralel empat dan delapan bit. Dengan menggunakan komunikasi paralel empat bit, maka dibutuhkan tujuh buah pin sebagai

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

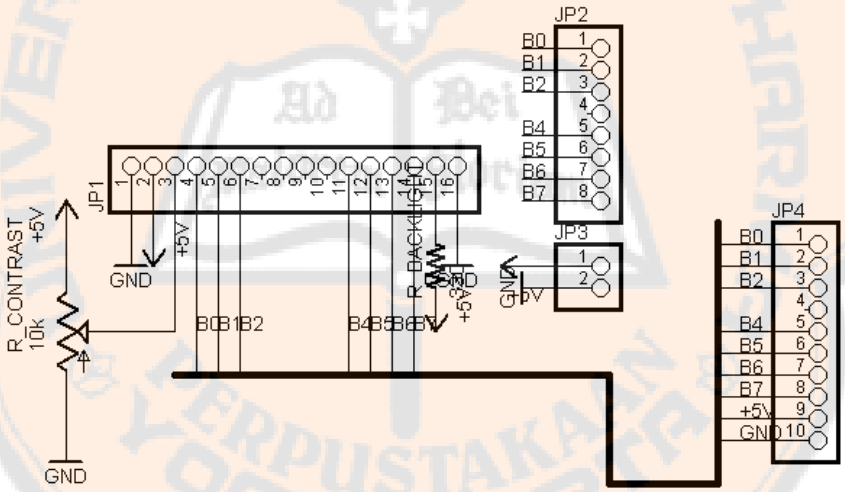
antarmuka antara mikrokontroler dengan perangkat LCD ini (empat jalur untuk data dan tiga jalur untuk kontrol) [7].

Berdasarkan datasheet tegangan kontras (*pin* V_O) maksimum LCD ini adalah 5 volt, sehingga digunakan sebuah varistor 10K ohm yang digunakan untuk membatasi tegangan pada pin ini. LCD ini juga memiliki sebuah lampu latar yang dapat dinyalakan dengan memberikan tegangan sebesar 4,2 volt pada *pin* A, dan menghubungkan *pin* K ke jalur *ground*. Arus maksimum yang mengalir pada pin adalah sebesar 80 mA, sehingga diperoleh nilai hambatan minimum untuk membatasi arus pada pin ini sebesar:

$$R_{minimum} = \frac{V_{CC} - V_F}{I_F} = \frac{5 - 4{,}2}{0{,}08} = 10\ ohm$$

Digunakan hambatan sebesar 100 ohm pada *pin* A , agar lampu latar belakang LCD tidak menyala terlalu terang.

Berikut rangkaian LCD dengan empat jalur data:



Gambar 3. 5. Rangkaian LCD

3.1.3 Rangkaian High Power LED

High Power LED yang digunakan pada perancangan ini memiliki tegangan maju sebesar 3,3 – 3,8 volt dengan arus maju sebesar 350 mA (*rating* daya 1 watt) [18]. *Power supply* yang digunakan pada perancangan ini memiliki tegangan sebesar 12 volt, sehingga dalam satu rangkaian terdapat tiga buah *LED*. Sebuah hambatan digunakan pada setiap rangkaian *LED* untuk membatasi arus yang melewatinya. Hambatan minimum yang digunakan pada setiap rangkaian *LED* adalah sebesar:

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

$$I_F = \frac{V_{CC} - n * V_F}{R_{min}}$$

$$R_{min} = \frac{V_{CC} - n * V_F}{I_F}$$

$$R_{min} = \frac{12 - 3 * 3{,}8}{0{,}35}$$

$$R_{min} = 1{,}71\ ohm$$

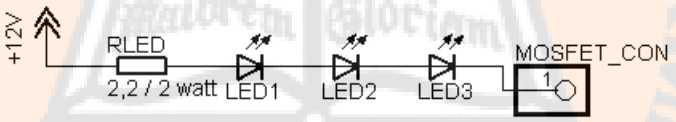
Digunakan hambatan sebesar 2,2 ohm, sehingga disipasi daya pada hambatan adalah sebesar:

$$P = V * \left(\frac{V_{cc} - n * V_F}{R}\right)$$

$$3{,}8 * \left(\frac{12 - 3 * 3{,}8}{2{,}2}\right) = 1{,}036\ watt$$

Digunakan hambatan yang memiliki rating daya sebesar 2 watt.

Pada kaki katoda *LED* terakhir akan dihubungkan ke rangkaian *MOSFET*. Berikut gambar rangkaian *LED*:



Gambar 3. 6. Rangkaian *High Power LED*

3.1.4 Rangkaian *MOSFET* dan D*river*

MOSFET digunakan untuk pensaklaran h*igh power LED*. *MOSFET* dapat digunakan sebagai perangkat *switching* dengan kecepatan tinggi, sehingga cocok digunakan sebagai *dimmer high power LED*. Di antara rangkaian kontrol dan *MOSFET* ditambahkan sebuah rangkaian *driver MOSFET*. Rangkaian *driver* juga berfungsi untuk meningkatkan waktu *switching MOSFET* [11]. Rangkaian *driver MOSFET* menggunakan sebuah *transistor NPN* dan sebuah *transistor PNP*.

Persamaan (2.10) digunakan untuk menghitung besarnya arus $I_{G,minimum}$ pada waktu *switching* tertentu. *N-chanel MOSFET* tipe *IRF540* digunakan pada perancangan ini. *MOSFET* tipe *IRF540* mempunyai *total gate charge* sebesar 70 nC, maka agar waktu *switching MOSFET (turn-on time* dan *turn-off time)* kurang dari 1 µs, diperlukan arus $I_{G,minimum}$ sebesar:

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

$$Q_{Gate,total} = I_{G,min} * t$$

$$I_{G,min} = \frac{Q_{Gate,total}}{t}$$

$$I_{G,min} = \frac{70.10^{-9}}{1.10^{-6}}$$

$$I_{G,min} = 70\ mA$$

Satu buah *transistor NPN 2N2222* dan satu buah *transistor PNP 2N2907* yang dirangkai seperti pada gambar 3.7 digunakan sebagai *driver MOSFET*. Saat mikrokontroler berkondisi logika satu, maka *transistor NPN* akan *on* dan *transistor PNP* akan *off* sehingga tegangan antara kolektor dan emitor *transistor NPN* mendekati atau sama dengan nol, sedangkan saat mikrokontroler berkondisi logika nol, maka *transistor NPN* akan *off* dan *transistor PNP* akan *on* sehingga tegangan antara kolektor dan emitor *transistor PNP* mendekati atau sama dengan nol. Agar arus yang mengalir pada resistor *gate* (R_G) lebih besar atau sama dengan arus *gate* minimum, maka resistor gate minimum yang digunakan adalah sebesar:

$$R_{G,min} = \frac{V_{CC} - V_{CE}}{I_{G,min}}$$

$$R_{G,min} = \frac{5 - 0}{0{,}07}$$

$$R_{G,min} = 71{,}42\ ohm$$

Digunakan resistor *gate* sebesar 56 ohm sehingga disipasi daya pada resistor ini adalah sebesar:

$$P = (V_{CC} - V_{CE}) * \left(\frac{(V_{CC} - V_{CE})}{R_G}\right)$$

$$P = (5 - 0) * \left(\frac{(5 - 0)}{56}\right)$$

$$P = 0{,}45\ watt$$

Rating daya resistor *gate* yang digunakan adalah 1 watt.

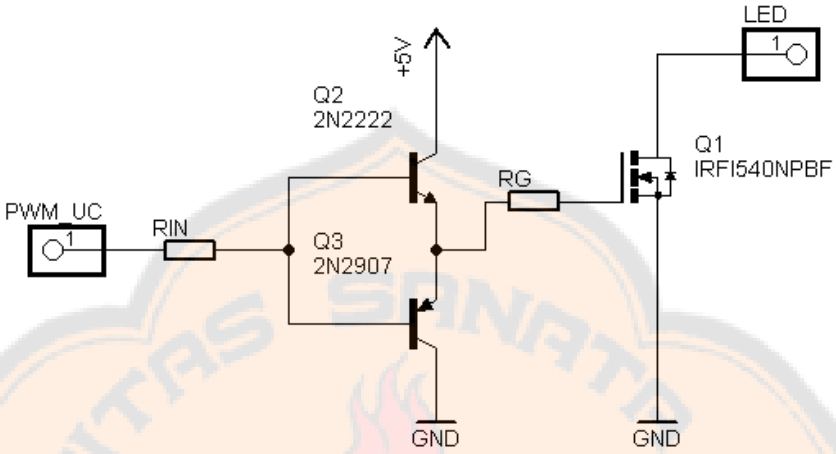
Untuk menghitung nilai R_{IN}, maka harus diketahui arus kolektor atau arus emitor transistor. Arus yang mengalir pada resistor *gate* sama dengan arus yang mengalir pada kaki emitor transistor yang sedang dalam kondisi *on* yaitu sebesar:

$$I_{RG} = I_E = \left(\frac{(V_{CC} - V_{CE})}{R_G}\right)$$

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

$$I_{RG} = I_E = \left(\frac{(5-0)}{56}\right)$$

$$I_{RG} = I_E = 89\,mA$$



Gambar 3. 7. Rangkaian *driver MOSFET*

Menggunakan persamaan (2.11), apabila nilai *hfe* transistor 2N2222 dan 2N2907 adalah sebesar 100, maka nilai R_{IN} maksimum yang dapat digunakan adalah sebesar:

$$I_E = (I_B + I_C)$$

$$I_E = (I_B + h_{FE} * I_B)$$

$$I_E = I_B(1 + h_{FE})$$

$$I_E = \left(\frac{V_{High} - V_{BE}}{R_{IN,maks}}\right) * (1 + h_{FE})$$

$$0{,}089 = \left(\frac{5 - 0{,}7}{R_{IN,maks}}\right) * (1 + 100)$$

$$R_{IN,maks} = \left(\frac{434{,}3}{0{,}089}\right)$$

$$R_{IN,maks} = 4{,}88\,kOhm$$

Digunakan resistor sebesar 4,7 kOhm sebagai R_{IN} yang mempunyai rating daya sebesar:

$$P = (V_{High} - V_{BE}) * \left(\frac{(V_{High} - V_{BE})}{R_{IN}}\right)$$

$$P = (5 - 0{,}7) * \left(\frac{(5 - 0{,}7)}{4700}\right)$$

$$P = 3{,}93\,mWatt \cong 0{,}25\,watt$$

Arus yang mampu dilewatkan oleh *MOSFET* dipengaruhi oleh tegangan *gate-source*-nya *(V_{GS})*. Semakin kecil *V_{GS}*, maka arus yang mampu dilewatkan *MOSFET* akan

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

semakin kecil [12]. Untuk menghitung arus maksimum yang mampu dilewatkan *MOSFET* dengan tegangan *gate* tertentu, maka perlu dihitung konstanta k seperti pada persamaan (2.9). Dari kurva karakteristik *MOSFET IRF540* dan persamaan (2.9) diperoleh nilai k sebesar:

$$I_D = k\left(V_{GS} - V_{GS(TH)}\right)^2$$

$$k = \frac{\left(V_{GS} - V_{GS(TH)}\right)^2}{I_D}$$

$$k = \frac{(10-3)^2}{33}$$

$$k = 1{,}485$$

Dari perhitungan di atas, maka dapat diperoleh arus I_D maksimal yang mampu dilewatkan *MOSFET* ini pada nilai V_{GS} 5 volt, yaitu sebesar:

$$I_{D,max} = k\left(V_{GS} - V_{GS(TH)}\right)^2$$

$$I_{D,max} = 1{,}485(5-3)^2$$

$$I_{D,max} = 5{,}94\ A$$

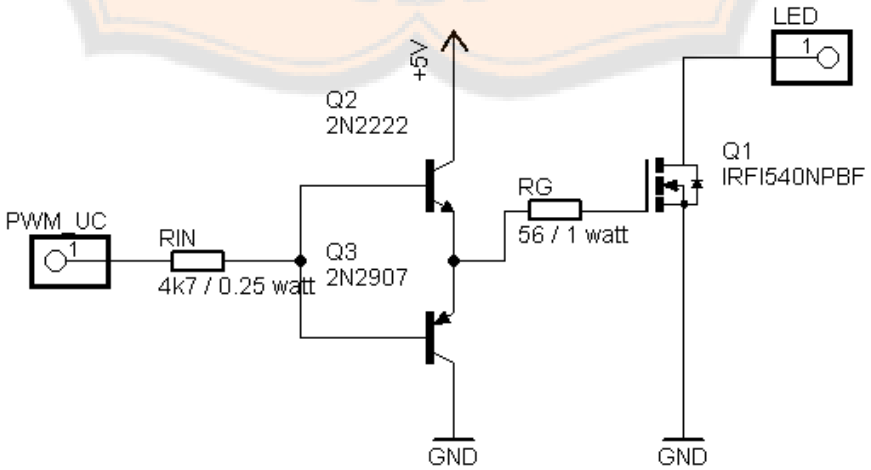
Karena setiap rangkaian *LED* membutuhkan arus sebesar 350 mA, maka setiap rangkaian *MOSFET* dapat digunakan untuk menyuplai rangkaian *LED* sebanyak:

$$n_{max} = \frac{I_D}{I_{F,LED}}$$

$$n_{max} = \frac{5{,}94}{0{,}35}$$

$$n_{max} = 16{,}97 \cong 16\ rangkaian\ LED$$

Berikut gambar rangkaian *driver MOSFET*:



Gambar 3. 8. Rangkaian *driver MOSFET*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

3.1.5 Rangkaian *Relay*

Relay digunakan sebagai pengontrol kerja *chiller* dan *dosing pump*. Digunakan *transistor NPN* untuk memisahkan rangkaian *relay* dengan rangkaian mikrokontroler. *Transistor* tipe 2N2222 digunakan pada perancangan ini.

3.1.5.1. Rangkaian Relay Chiller

Chiller yang digunakan pada perancangan ini adalah *chiller Resun CL-280*. *Chiller* ini mempunyai *rating* daya sebesar 1/10 hp atau sekitar 75 watt dengan *rating* arus antara 0,9 – 1,2 *ampere*. Dua r*elay* elektromagnetis yang dirangkai secara parallel digunakan pada perancangan ini. *Relay* ini masing - masing mempunyai *rating* arus sebesar 5 *ampere* dan hambatan gulungan sebesar 400 ohm. Arus yang mengalir pada gulungan *relay* adalah sebesar:

$$\frac{12}{(400 * 400) \Big/ (400 + 400)} = 0{,}06\ ampere$$

Gambar 3. 9. *Chiller Resun CL-280*

Transistor 2N2222 mempunyai h_{FE} sebesar 100, agar arus yang mengalir pada relay menjadi sebesar 60 mA, maka resistor *base* (R_B) maksimum yang dapat digunakan adalah sebesar:

$$I_C = h_{FE} * I_B$$

$$I_C = h_{FE} * \left(\frac{V_{High} - V_{BE}}{R_{B,max}}\right)$$

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

$$R_{B,max} = h_{FE} * \left(\frac{V_{High} - V_{BE}}{I_C}\right)$$

$$R_{B,max} = 100 * \left(\frac{5 - 0{,}7}{0{,}06}\right)$$

$$R_{B,max} = 7{,}167\ kOhm$$

Hambatan sebesar 4,7 kOhm digunakan, sehingga disipasi daya pada hambatan ini adalah sebesar:

$$P = (V_{High} - V_{BE}) * \left(\frac{(V_{High} - V_{BE})}{R_B}\right)$$

$$P = (5 - 0{,}7) * \left(\frac{(5 - 0{,}7)}{4700}\right)$$

$$P = 3{,}9\ mWatt$$

Digunakan hambatan yang mempunyai rating daya sebesar 0,25 watt.

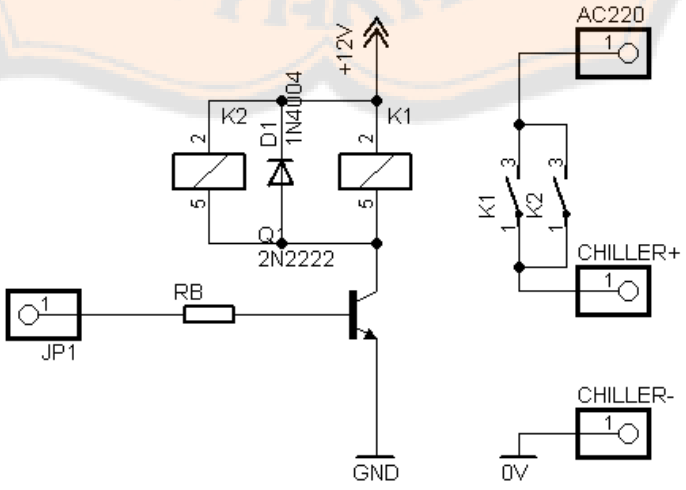
Ketika *relay* berubah kondisinya dari *on* ke *off,* tegangan pada gulungan relay menjadi sangat besar, sehingga diperlukan pengaman kondisi *transient (dv/dt)* untuk mengamankan rangkaian kontrol [13]. Digunakan dioda pengaman yang dipasang paralel dengan lilitan *relay. Rating* arus minimum yang dimiliki dioda adalah sebesar:

$$I_{D,freewheeling} = \frac{V_{cc}}{R_{coil}}$$

$$I_{D,freewheeling} = \frac{12}{400}$$

$$I_{D,freewheeling} = 0{,}03\ ampere$$

Digunakan dioda 1N4004 pada perancangan ini. Berikut gambar rangkaian *relay chiller:*



Gambar 3. 10. Rangkaian r*elay chiller*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

3.1.5.2. Rangkaian Relay Dosing Pump

Dosing pump yang digunakan pada perancangan ini mempunyai *rating* daya 4 watt. Digunakan *relay* elektromagnetik yang memiliki *rating* arus sebesar 5 *ampere*. *Relay* ini mempunyai hambatan dalam sebesar 400 ohm dan tegangan *coil* sebesar 12 volt. Arus yang mengalir pada gulungan *relay* adalah sebesar:

$$\frac{12}{400} = 0{,}03\ ampere$$

Agar transistor mampu mengalirkan arus sebesar 30 mili *ampere*, maka resistor *base* (R_B) maksimum yang dapat digunakan adalah sebesar:

$$I_C = h_{FE} * I_B$$

$$I_C = h_{FE} * \left(\frac{V_{High} - V_{BE}}{R_{B,max}}\right)$$

$$R_{B,max} = h_{FE} * \left(\frac{V_{High} - V_{BE}}{I_C}\right)$$

$$R_{B,max} = 100 * \left(\frac{5 - 0{,}7}{0{,}03}\right)$$

$$R_{B,max} = 14{,}33\ kOhm$$

Hambatan sebesar 4,7 kOhm digunakan, sehingga disipasi daya pada hambatan ini adalah sebesar:

$$P = (V_{High} - V_{BE}) * \left(\frac{(V_{High} - V_{BE})}{R_B}\right)$$

$$P = (5 - 0{,}7) * \left(\frac{(5 - 0{,}7)}{4700}\right)$$

$$P = 3{,}9\ mWatt$$

Digunakan hambatan yang mempunyai *rating* daya sebesar 0,25 watt.

Ketika *relay* berubah kondisinya dari *on* ke *off,* tegangan pada gulungan relay menjadi sangat besar, sehingga diperlukan pengaman kondisi *transient (dv/dt)* untuk mengamankan rangkaian kontrol [13]. Digunakan dioda pengaman yang dipasang paralel dengan lilitan *relay*. *Rating* arus minimum yang dimiliki dioda adalah sebesar:

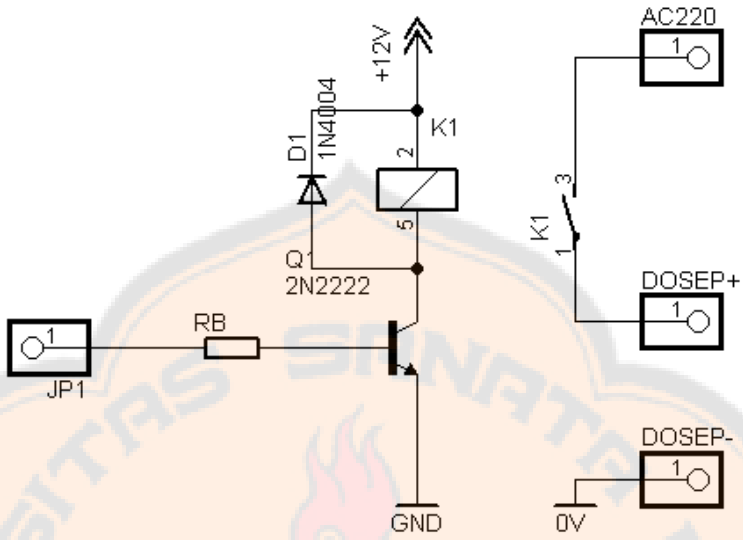
$$I_{D,freewheeling} = \frac{V_{cc}}{R_{coil}}$$

$$I_{D,freewheeling} = \frac{12}{400}$$

$$I_{D,freewheeling} = 0{,}03\ ampere$$

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

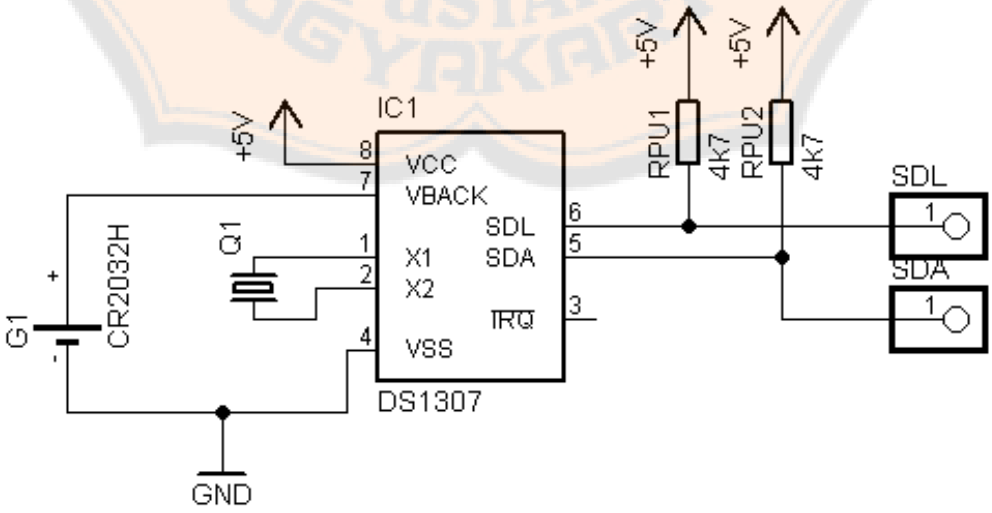
Digunakan dioda 1N4004 pada perancangan ini. Gambar rangkaian *relay dosing pump* adalah sebagai berikut:



Gambar 3. 11. Rangkaian *relay dosing pump*

3.1.6 Rangkaian RTC – IC DS1307

Pada datasheet *IC DS1307* telah terdapat rangkaian standar *IC* ini. Pada jalur data dan sumber *clock* komunikasi dua jalur diperlukan tambahan hambatan *pull-up* agar dapat bekerja. Hambatan 4,7 kOhm digunakan sebagai hambatan *pull-up.* Kristal *oscillator* yang digunakan adalah kristal yang memiliki frekuensi 32,768 kHz. Ditambahkan baterai tipe CR2032 sebagai sumber daya *IC* ini ketika sumber daya utama tidak tersedia. Berikut gambar rangkaian *RTC* dengan *IC DS1307:*

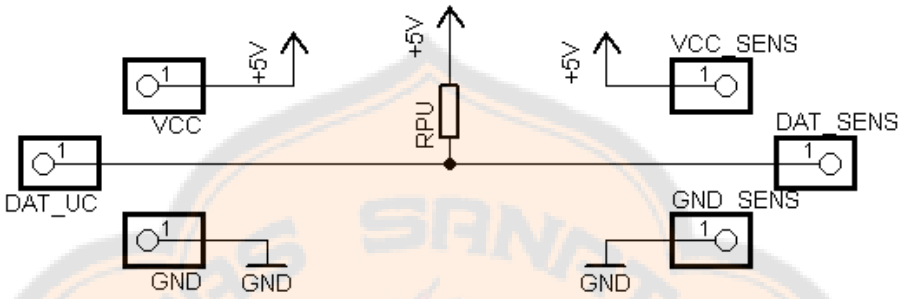


Gambar 3. 12. Rangkaian *RTC*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

3.1.7 Rangkaian Sensor Suhu – *DS18B20*

Sensor suhu *DS18B20* memerlukan sebuah hambatan *pull-up* pada jalur data untuk dapat bekerja [9]. Dari *datasheet IC DS18B20* disarankan menggunakan hambatan *pull-up* sebesar 4,7 kohm.



Gambar 3. 13. Rangkaian sensor *DS18B20*

3.1.8 Rangkaian Water Level Sensor – Float Switch

Float switch bekerja seperti sebuah saklar. Sensor ini dihubungkan ke mikrokontroler dan digunakan sebagai sumber interupsi program. Pada keadaan normal, sensor ini berada pada keadaan hubung buka. Ditambahkan sebuah hambatan untuk membatasi arus yang melewati sensor ini. Agar arus maksimum yang melewati sensor ini menjadi sebesar 10 mA, maka hambatan minimum yang digunakan untuk membatasi arus ini adalah sebesar:

$$R_{FS,min} = \frac{V_{cc}}{I_{max}}$$

$$R_{FS,min} = \frac{5}{0{,}01}$$

$$R_{FS,min} = 500\ ohm$$

Digunakan hambatan sebesar 560 ohm sehingga disipasi daya pada hambatan ini adalah sebesar:

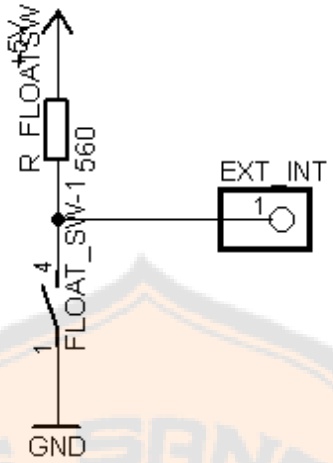
$$P = V * \left(\frac{V}{R_{FS,min}}\right)$$

$$P = 5 * \left(\frac{5}{560}\right)$$

$$P = 45\ mWatt$$

Digunakan hambatan sebesar 560 ohm dengan *rating* daya sebesar 0,25 watt. Berikut gambar rangkaian sensor float switch:

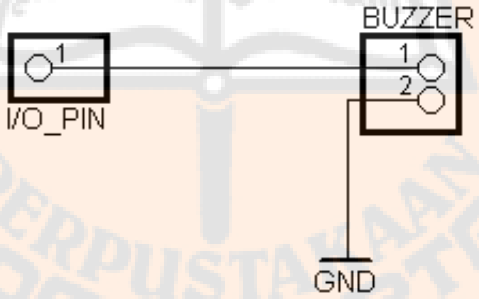
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Gambar 3. 14. Rangkaian *float switch*

3.1.9 Rangkaian *Buzzer*

Buzzer digunakan sebagai penanda terjadinya kegagalan pada sistem. *Buzzer* yang digunakan pada perancangan ini mempunyai spesifikasi tegangan sebesar 5 volt dan arus maksimum sebesar 30 mA. Pin *I/O* ATmega128 dapat menyuplai arus sampai dengan 40 mA, sehingga tidak diperlukan tambahan driver untuk menggunakan *buzzer* ini. Berikut merupakan gambar rangkaian *buzzer:*



Gambar 3. 15. Rangkaian *buzzer*

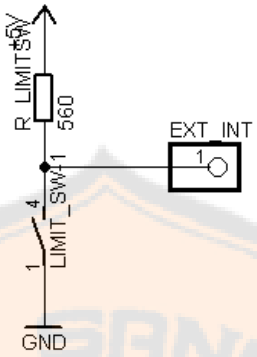
3.1.10 Perancangan Dosing / Syringe Pump

Sistem rotasi digunakan untuk menarik *plunger* alat suntik agar cairan masuk ke dalam tabung alat suntik. Jenis alat suntik yang digunakan pada perancangan ini adalah *Terumo 12 ml*. Berdasarkan hasil pengukuran, jarak antara dosis 0 ml dan 10 ml pada alat suntik ini adalah 52 mm. Agar dapat menarik *plunger* sejauh 52 mm, maka jarak penarik dengan titik tengah bagian pemutar adalah 21 mm.

Motor yang digunakan sebagai pemutar alat penarik adalah motor AC yang mempunyai spesifikasi daya 4 watt. Di bagian bawah pemutar terdapat sebuah *limit switch* yang digunakan untuk menghitung jumlah putaran motor. Apabila penanda melewati *limit*

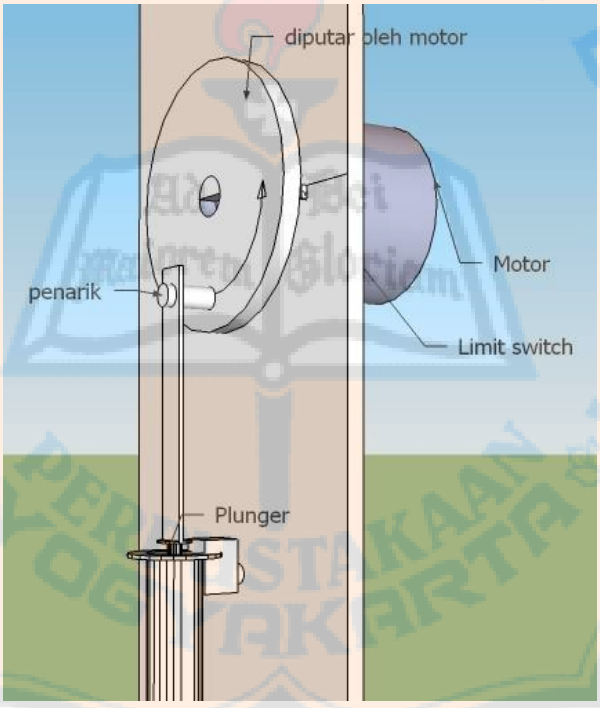
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

switch, maka program akan terinterupsi dan menambah nilai sebuah variabel pencacah. Rangkaian *limit switch* yang digunakan sama dengan rangkaian *float switch*.



Gambar 3. 16. Rangkaian *limit switch*

Berikut merupakan gambar desain perancangan pompa *syringe:*

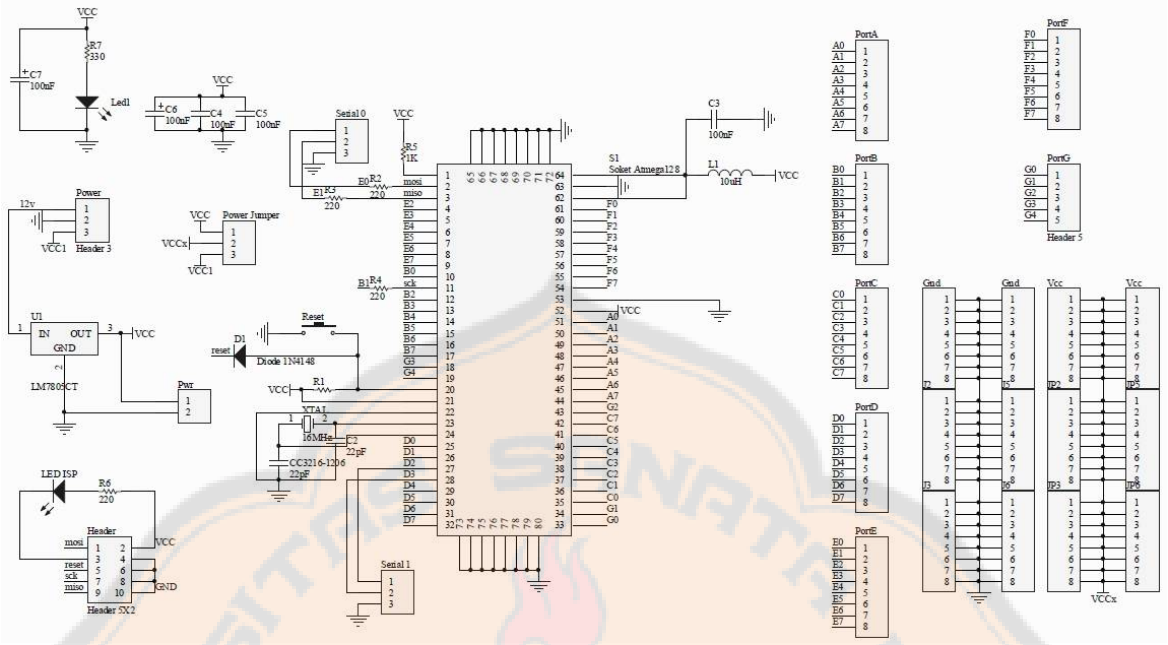


Gambar 3. 17. Desain *dosing / syringe pump*

3.1.11 Rangkaian Minimum System ATmega128

Minimum system yang digunakan pada perancangan ini adalah *EMA-128 ATMEGA128 CPU Module* buatan *Creative Vision. Minimum system* ini telah dilengkapi *port ISP* dan *port I/O.* Berikut gambar rangkaian *minimum system:*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

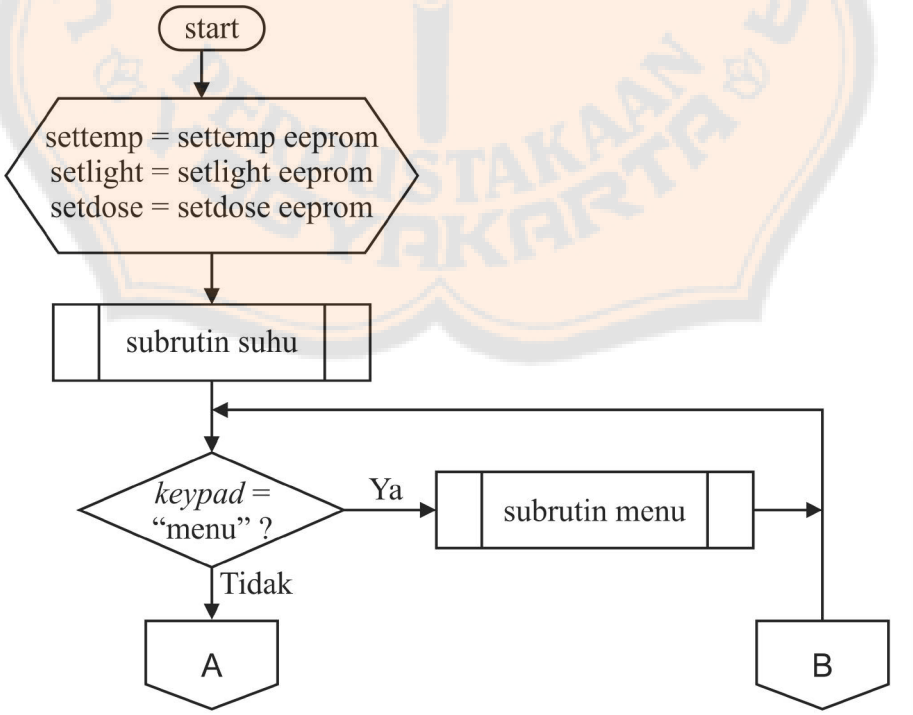


Gambar 3. 18. Gambar rangkaian *minimum system* ATmega128 [19]

3.2. Perancangan Perangkat Lunak

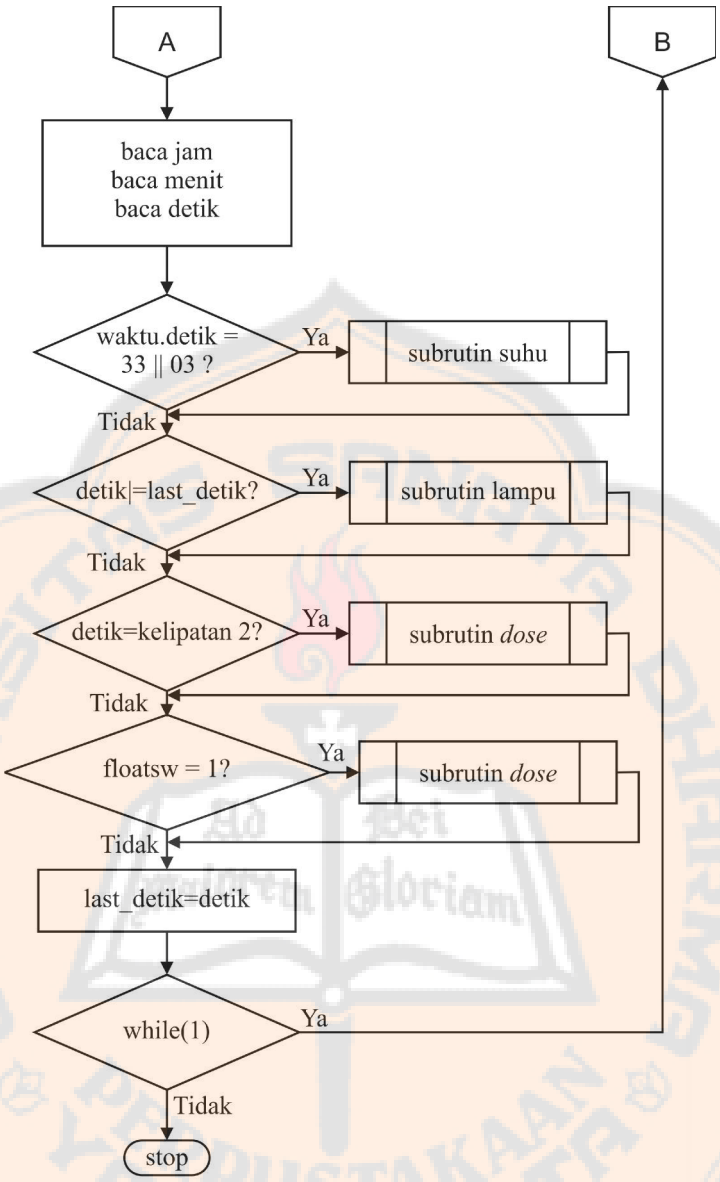
3.2.1 Diagram Alir Program Utama

Diagram alir program utama ditunjukkan gambar 3.19. Program dimulai dengan mengambil pengaturan dari memori *EEPROM,* dan proses inisialisasi *I/O* dan fitur – fitur yang digunakan.



Gambar 3. 19.Diagram alir program utama

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



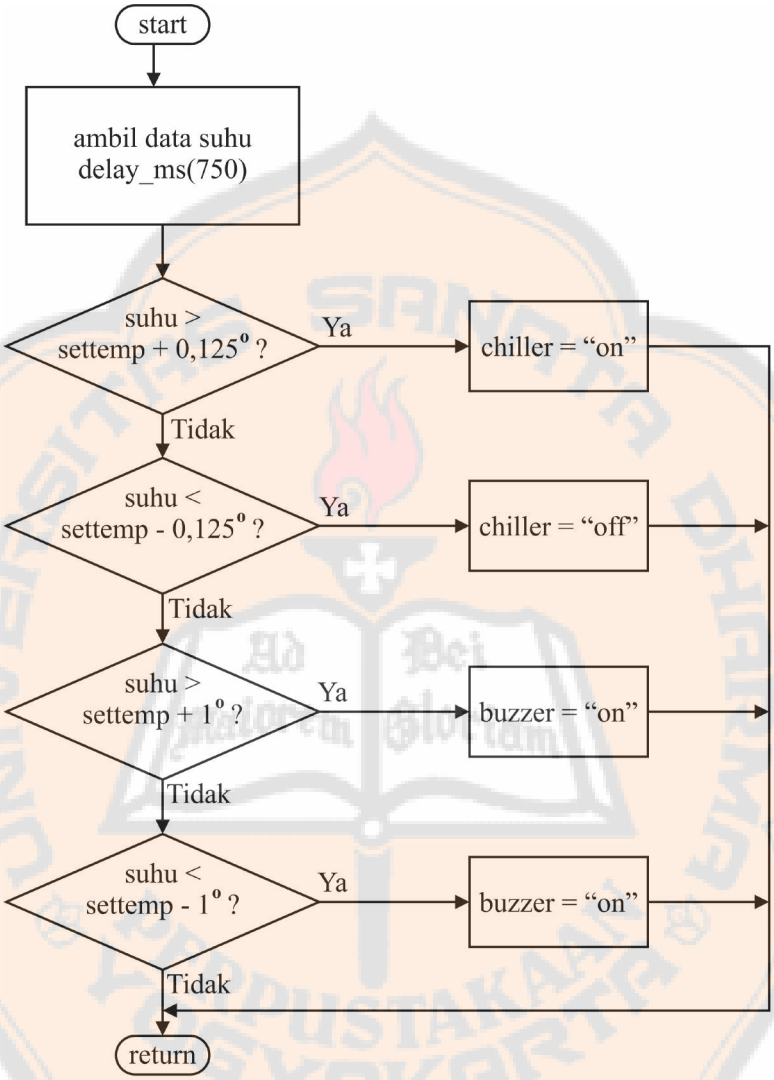
Gambar 3. 19.(Lanjutan) Diagram alir program utama

Setiap siklus program utama dilakukan proses pengambilan data waktu dari *RTC,* dan pengambilan data suhu akuarium setiap 30 detik sekali. Proses ini dilakukan berulang – ulang, kecuali tombol menu ditekan oleh pengguna untuk mengubah pengaturan yang ada, sehingga program akan terinterupsi dan menuju ke subrutin menu. Pengguna dapat mengubah pengaturan waktu, pencahayaan, suhu, dan penambahan bahan aditif di dalam subrutin menu. Sensor *float switch* juga digunakan sebagai masukan interupsi program. Apabila *float switch* mendeteksi perubahan level air, maka program akan terinterupsi menuju subrutin penambahan bahan aditif. Masukan ini kemudian dapat digunakan sebagai

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

tanda untuk mengaktifkan *dosing pump*, ataupun mematikan pompa yang ada di akuarium apabila diperlukan.

3.2.2 Diagram Alir Subrutin Suhu

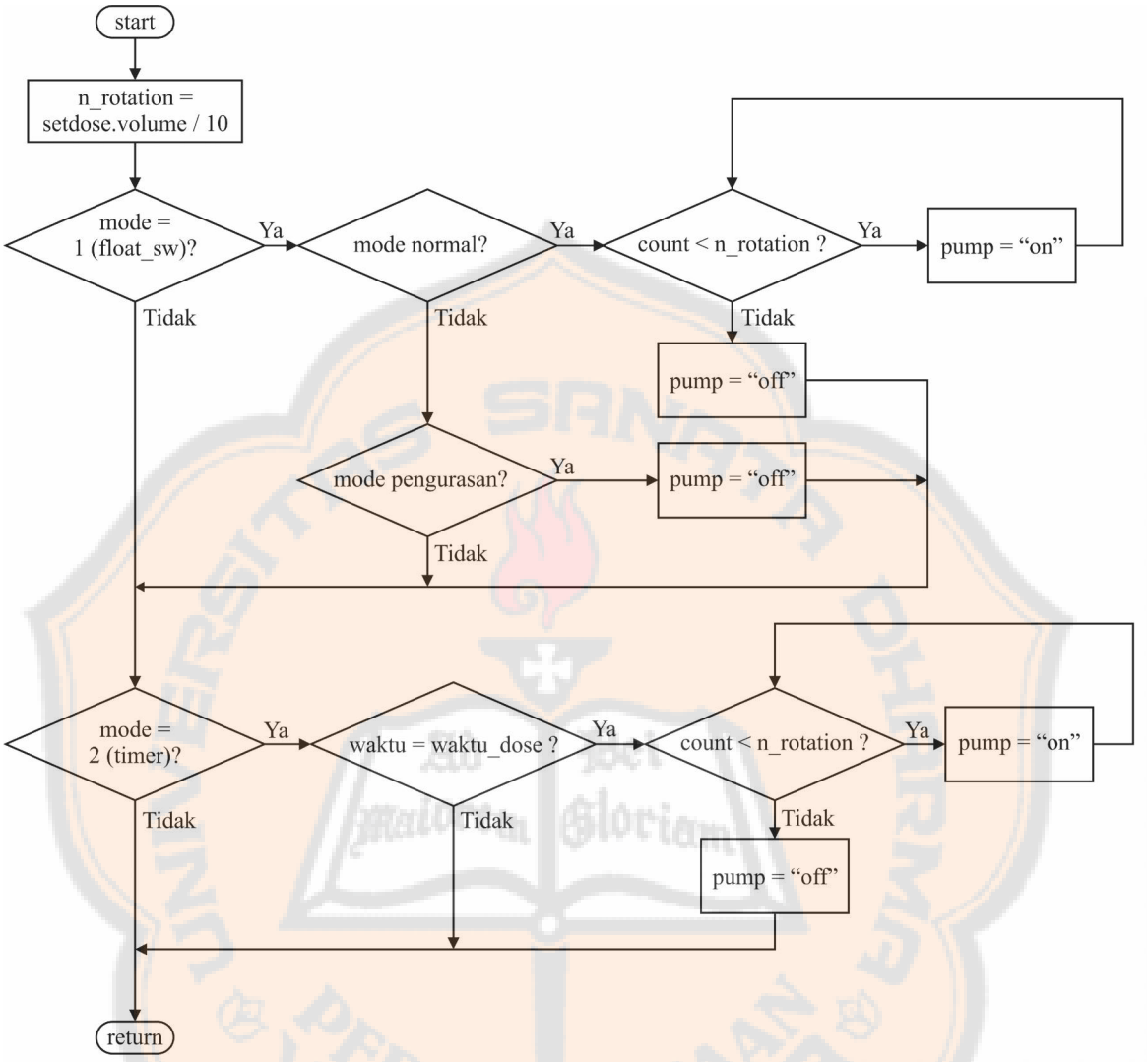


Gambar 3. 20. Diagram alir subrutin suhu

Diagram alir subrutin suhu ditunjukkan oleh gambar 3.20. Dengan mengirimkan data 0x44, maka sensor akan mulai melakukan konversi suhu, dan mengirimkan nilai suhu yang terbaca ke mikrokontroler setelah waktu konversi selesai. *Chiller* akan menyala apabila suhu akuarium lebih tinggi 0,125 derajat Celsius dari nilai suhu yang diatur pengguna. *Chiller* akan terus menyala sampai suhu akuarium menjadi lebih rendah 0,125 derajat Celsius dari nilai suhu yang diatur pengguna. Apabila nilai suhu akuarium yang terbaca lebih tinggi atau lebih rendah satu derajat Celsius dari nilai suhu akuarium yang diatur pengguna, maka *buzzer* akan menyala.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

3.2.3 Diagram Alir Subrutin Penambahan Bahan Aditif *(Dose)*



Gambar 3. 21. Diagram alir subrutin penambahan bahan aditif

Diagram alir subrutin penambahan bahan aditif ditunjukkan oleh gambar 3.21. Terdapat dua mode pada subrutin penambahan bahan aditif, yaitu mode *timer* dan mode *float switch.* Pada mode *timer*, pompa penambah bahan aditif akan aktif apabila waktu menunjukkan nilai yang sama dengan waktu *dosing* yang telah diatur oleh pengguna. Pompa akan terus menyala sampai banyaknya volum cairan yang diberikan sesuai dengan volum cairan yang diatur pengguna.

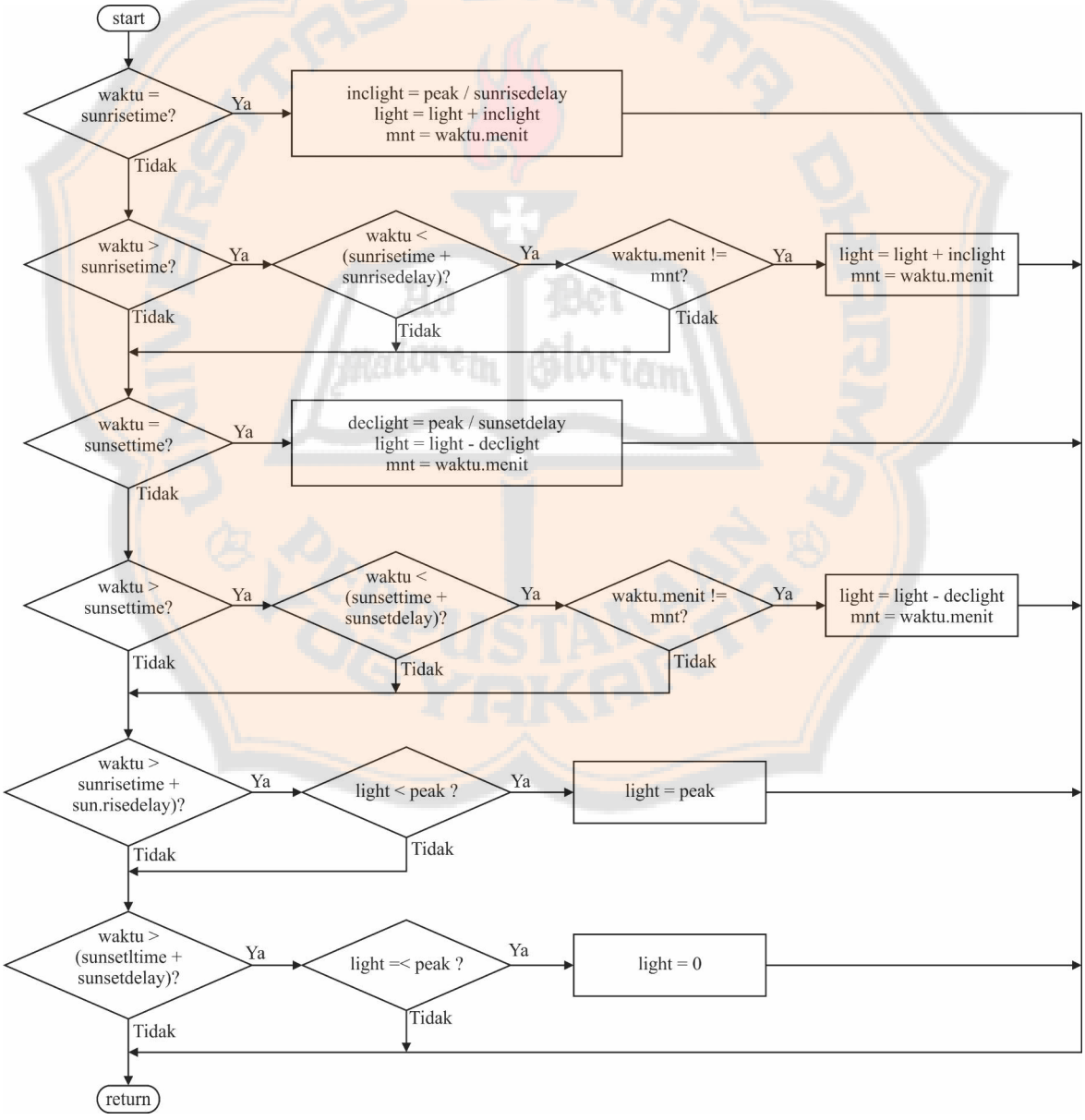
Khusus pada mode *float switch,* terdapat dua mode yang dapat diatur oleh pengguna, yaitu normal dan mode pengurasan. Pada metode normal, maka ketika *float switch* berubah kondisinya, dari hubung buka menjadi hubung singkat pompa *dosing* akan menyala sampai *volume* cairan yang diberikan sama dengan *volume* yang diatur oleh

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

pengguna. Mode pengurasan bukan digunakan sebagai kepentingan penambahan bahan aditif, namun sebagai pengaman pompa filter, misalnya saat dilakukan pergantian air. Ketika *level* air di ruang filter akuarium berkurang sangat banyak, maka pompa yang berada pada ruang ini tidak lagi mampu menarik air. Kondisi ini dapat merusak pompa apabila pompa tidak segera dimatikan.

Variabel *count* adalah variabel yang digunakan untuk menghitung jumlah putaran pompa penambah bahan aditif. Setiap satu kali putaran, *count* akan bertambah nilainya yang menandakan cairan sebanyak 10ml telah ditambahkan ke dalam akuarium.

3.2.4 Diagram Alir Subrutin Pencahayaan *(Light)*



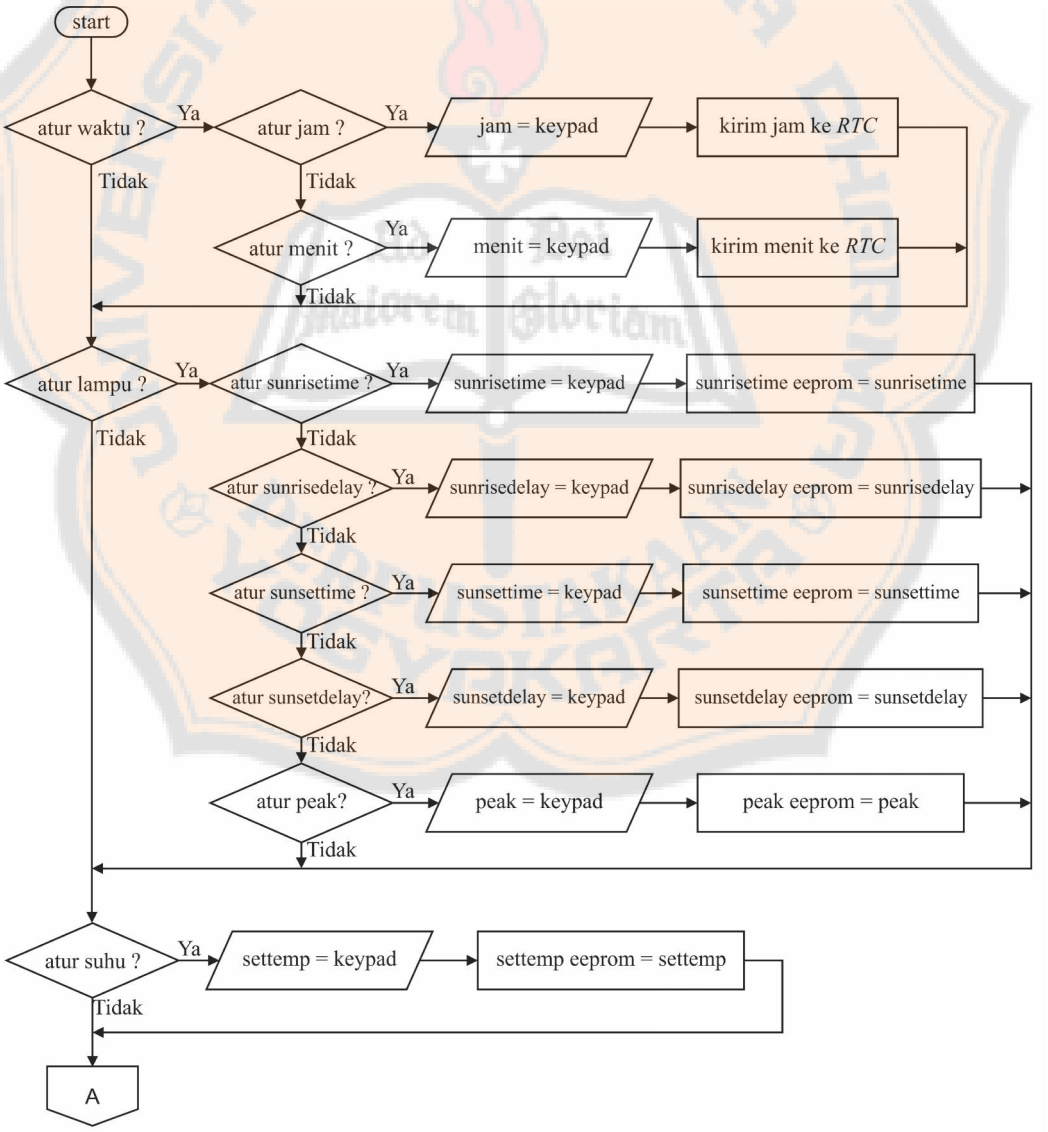
Gambar 3. 22. Diagram alir subrutin pencahayaan

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Diagram alir subrutin pencahayaan ditunjukkan oleh gambar 3.22. Variabel *light* adalah variabel yang menunjukkan intensitas cahaya yang dihasilkan oleh lampu. Intensitas cahaya yang dihasilkan lampu akan semakin tinggi apabila nilai variabel *light* juga semakin tinggi. Variabel *sunrisetime* dan *sunsettime* adalah variabel yang menunjukkan waktu lampu mulai menyala dan mulai mati. Intensitas cahaya yang dihasilkan lampu akan naik atau turun setiap menitnya.

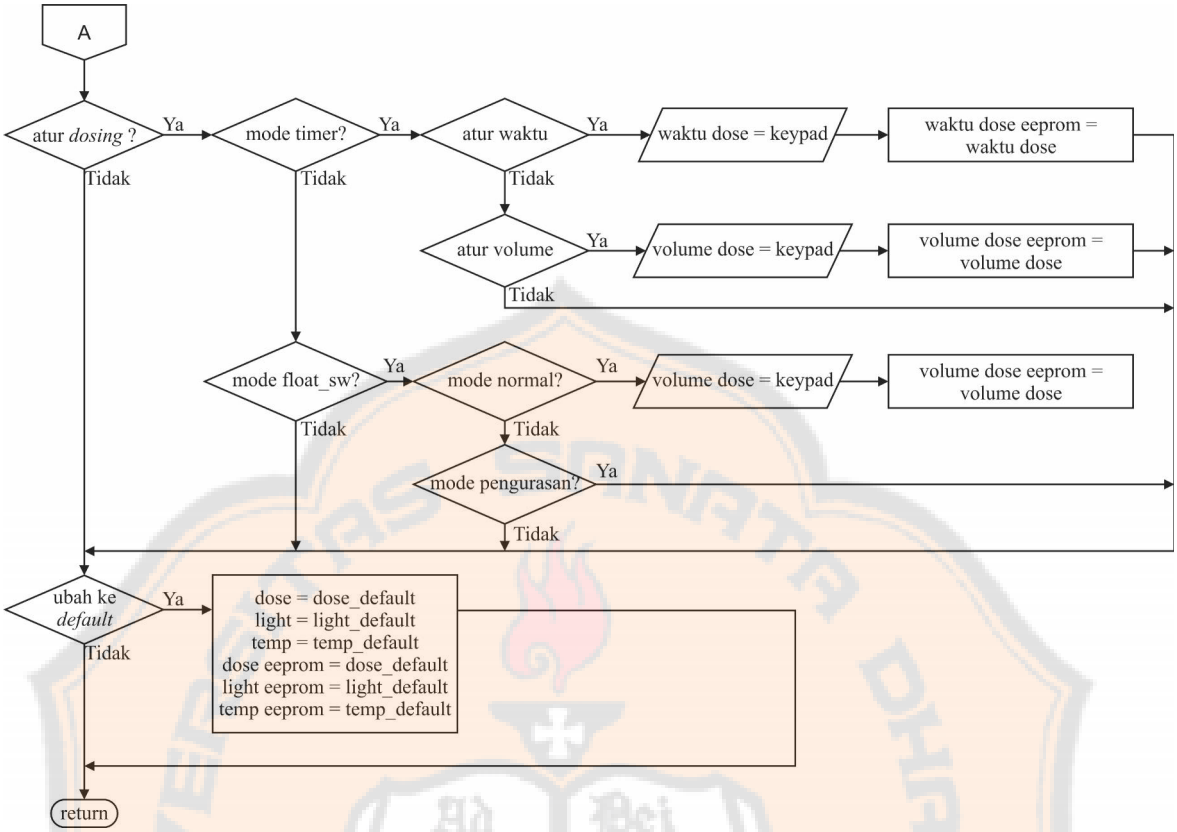
3.2.5 Diagram Alir Subrutin Menu

Diagram alir subrutin menu ditunjukkan oleh gambar 3.23. Di dalam subrutin menu, pengguna dapat mengatur waktu, pencahayaan, suhu, dan waktu pemberian bahan aditif.



Gambar 3. 23. Diagram alir subrutin menu

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Gambar 3. 23. (Lanjutan) Diagram alir subrutin menu

Pada subrutin menu, pengguna juga bisa mengubah semua pengaturan ke pengaturan *default.* Dengan memilih pengaturan ke mode *default,* semua pengaturan yang telah diatur pengguna akan kembali ke pengaturan awal. Pengaturan awal disimpan di *memori flash* mikroprosesor. Pengaturan ini akan memudahkan pengguna yang masih baru dalam hobi ini. Berikut merupakan tabel pengaturan *default* sistem kontrol:

Tabel 3. 1. Tabel Pengaturan *Default* Sistem Kontrol Akuarium

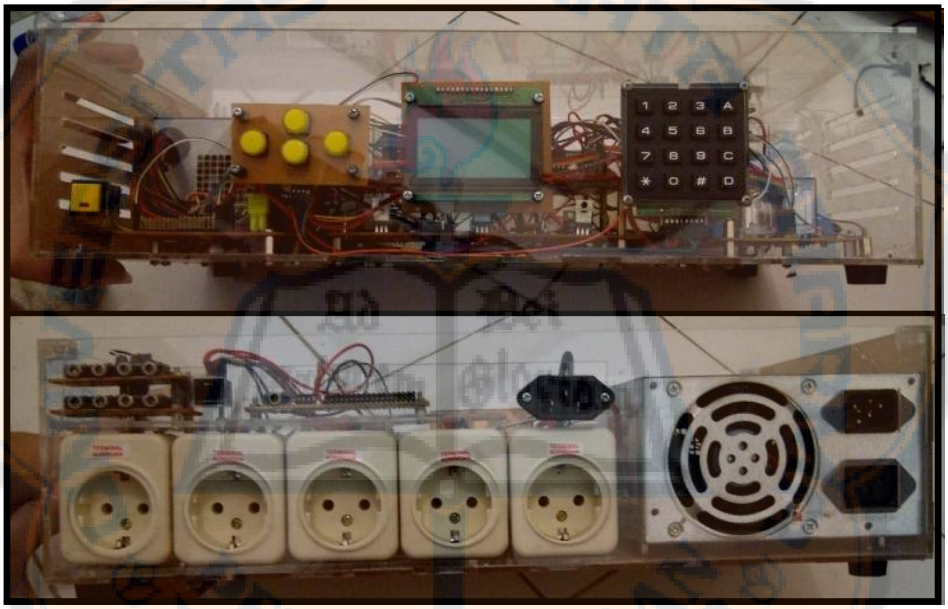
Kontrol Pencahayaan	Kanal 1 – 6	Waktu *sunrise* *Delay sunrise* Waktu *sunset* *Delay sunset* *Peak*	= 07.00 = 120 menit = 17.00 = 120 menit = 100%
Kontrol Penambahan Bahan Aditif	Kanal 1	Mode *Volume*	= *float switch normal* = 100 ml
	Kanal 2	Mode *Volume*	= *float switch* pengurasan = -
	Kanal 3 dan 4	*off*	
Kontrol Suhu		*Setpoint* suhu	= 23,5 ^{o}C

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

BAB IV
HASIL DAN PEMBAHASAN

4.1. Implementasi Alat

Alat yang dibuat terdiri dari beberapa subsistem diantaranya subsistem pengontrol waktu, subsistem pengontrol intensitas cahaya, subsistem pengontrol pemberian bahan aditif, dan subsistem pengontrol suhu. Gambar alat yang dibuat ditunjukkan oleh gambar di bawah ini:



Gambar 4. 1. Alat Kontrol Tampak Depan (Atas) dan Tampak Belakang (Bawah)

Setiap subsistem mempunyai komponen – komponen penunjang yang berbeda – beda. Dari hasil percobaan setiap subsistem yang ada, alat sudah dapat bekerja sesuai yang diharapkan.

4.1.1 Akuarium

Akuarium yang dibuat berukuran 40 cm x 35 cm x 40 cm (p x l x t). Di bagian belakang terdapat tiga sekat yang digunakan sebagai tempat alat filtrasi. Pada sekat filter terakhir, terdapat dua pompa yang digunakan untuk mengalirkan air kembali ke ruang *display* akuarium dan ke *chiller*. Sensor *float switch* juga berada pada sekat ini. Berikut gambar akuarium yang dibuat dari beberapa sisi:

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Gambar 4. 2. Akuarium TampakDepan

Gambar 4. 3. Akuarium Tampak Samping

4.1.2 Subsistem Pengontrol Waktu

Subsistem pengontrol waktu berfungsi untuk menjaga agar waktu alat kontrol sesuai dengan waktu yang sesungguhnya. Hal ini berhubungan dengan pengendalian yang bersifat *real-time*. Pada subsistem ini terdiri dari *IC DS1307* dan sebuah baterai yang digunakan sebagai sumber energi *IC* saat sumber listrik utama tidak tersedia. Rangkaian yang dibuat mengacu pada gambar 3.12. Berikut gambar rangkaian subsistem pengontrol waktu:

Gambar 4. 4. Rangkaian Subsistem Pengontrol Waktu

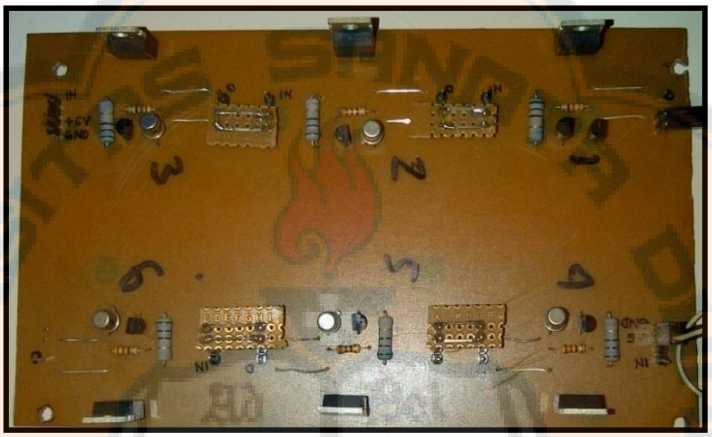
Pengujian dilakukan dengan mematikan semua sistem kontrol beberapa kali selama 10 detik. Setelah 10 detik, kontroler dinyalakan kembali, dan waktu yang ditunjukkan oleh kontroler telah bertambah 10 detik. Dari pengujian ini didapat kesimpulan bahwa sistem pengontrol waktu sudah dapat bekerja dan mempertahankan waktu kontroler.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

4.1.3 Subsistem Pengontrol Intensitas Cahaya

Subsistem ini berfungsi untuk mengontrol intensitas cahaya yang dihasilkan lampu *(high power LED)*. Subsistem ini terdiri dari sebuah rangkaian *driver high power LED* yang menggunakan *MOSFET.*

Rangkaian *driver MOSFET* yang dibuat mengacu pada gambar 3.8. Terdapat enam rangkaian *driver MOSFET* yang digunakan untuk mengontrol enam kanal *LED.* Berikut merupakan gambar rangkaian *driver MOSFET* yang dibuat:



Gambar 4. 5. Gambar Rangkaian *Driver LED*

Berikut merupakan hasil pengujian *driver high power LED:*

Tabel 4. 1. Data Hasil Pengujian *Driver High Power LED*

Tegangan pada pin masukan driver (volt)	**Tegangan pada pin keluaran *driver* CH1 (volt)**	**Tegangan pada pin keluaran *driver* CH2 (volt)**	**Tegangan pada pin keluaran *driver* CH3 (volt)**	**Tegangan pada pin keluaran *driver* CH4 (volt)**	**Tegangan pada pin keluaran *driver* CH5 (volt)**	**Tegangan pada pin keluaran *driver* CH6 (volt)**
5,03	0,41	0,41	0,38	0,53	0,53	0,54
0	6,46	4,99	7,04	5,21	5,42	5,42

Dari data tabel di atas, dapat dilihat bahwa saat kondisi logika tinggi pada pin masukan *driver*, maka tegangan keluaran *driver* berada di bawah 0,55 volt, yang menandakan kaki *drain* dan *source MOSFET* berada pada kondisi hubung singkat, sehingga *LED* dapat menyala. Sedangkan pada saat logika rendah pada pin masukan *driver*, maka tegangan keluaran *driver* berada di atas 4,5 volt, yang menandakan kaki *drain* dan *source MOSFET* berada pada kondisi hubung buka, sehingga *LED* tidak dapat menyala.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

4.1.4 Subsistem Pengontrol Penambahan Bahan Aditif

Subsistem ini berfungsi untuk menyalakan dan mematikan pompa penambah bahan aditif atau pompa filtrasi akuarium tergantung dari mode yang digunakan pengguna. Subsistem ini terdiri dari rangkaian *relay* yang dikontrol oleh mikrokontroler.

Rangkaian *relay* pompa *dosing* yang dibuat mengacu pada gambar 3.11. Terdapat empat rangkaian *relay dosing pump* yang dibuat untuk mengontrol empat kanal pengendalian penambahan bahan aditif. Keempat kanal ini dapat diatur secara individual mode pengendaliannya. Berikut merupakan gambar rangkaian *relay dosing pump* yang dibuat:

Gambar 4. 6. Gambar Rangkaian *Relay*

Berikut data pengujian rangkaian *relay:*

Tabel 4. 2. Data Hasil Pengujian Rangkaian *Relay*

Tegangan pada pin masukan rangkaian *relay* (volt)	**Kondisi Kontak *Relay***
5,03	Hubung singkat
0	Hubung buka

Dari tabel data di atas, dapat dilihat bahwa *relay* akan terhubung ketika diberi logika tinggi.

4.1.5 Subsistem Pengontrol Suhu

Subsistem ini berfungsi untuk mempertahankan suhu akuarium pada suhu yang diinginkan pengguna. Proses mendinginkan air dibantu oleh alat *chiller*. Kontroler kemudian akan mengontrol nyala dan mati *chiller* sehingga suhu akuarium sesuai dengan pengaturan pengguna. Subsistem ini terdiri dari rangkaian *relay* yang dikontrol oleh mikrokontroler. Rangkaian *relay chiller* yang dibuat mengacu pada gambar 3.10. Berikut merupakan gambar rangkaian *relay chiller:*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Gambar 4. 7. Gambar Rangkaian *Relay Chiller*

Berikut data pengujian rangkaian *relay:*

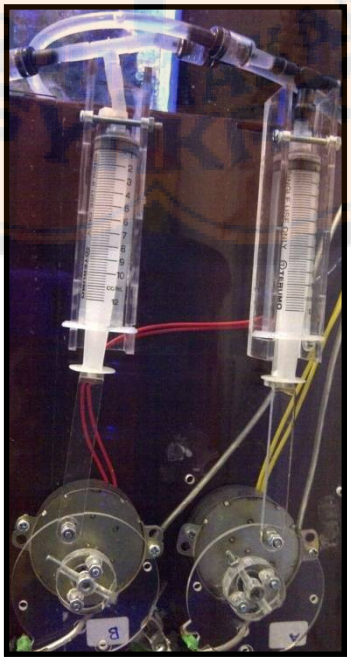
Tabel 4. 3. Data Hasil Pengujian Rangkaian *Relay Chiller*

Tegangan pada pin masukan rangkaian *relay* (volt)	**Kondisi Kontak *Relay***
5,03	Hubung singkat
0	Hubung buka

Dari tabel data di atas, dapat dilihat bahwa *relay* akan terhubung ketika diberi logika tinggi.

4.1.6 Pengujian Dosing Pump

Desain pompa *dosing* yang dibuat mengacu pada gambar 3.17. Terdapat perbedaan letak sensor limit switch dari gambar perancangan sebelumnya. Letak *limit switch* berada di atas penarik masing – masing pompa *dosing.* Berikut merupakan gambar pompa *dosing*:



Gambar 4. 8. Pompa *Dosing*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Pengujian *dosing pump* dilakukan dengan membandingkan volume cairan yang diberikan pompa, dengan pengaturan dari kontroler. Volume cairan yang diberikan pompa diukur dengan menggunakan gelas ukur yang mempunyai ketelitian satu milliliter dengan volume maksimal 50 ml. Pengukuran cairan dengan volume lebih dari 50 ml dilakukan secara berulang – ulang. Cairan pertama – tama ditampung dalam gelas, kemudian dari gelas, cairan tersebut dipindahkan ke dalam gelas ukur dan diukur volumenya sampai cairan dalam gelas habis. Volume yang terukur kemudian dijumlahkan, dan dicatat sebagai data volume cairan yang diberikan pompa. Berikut merupakan data perbandingan pengaturan kontroler dengan volume cairan yang diberikan pompa:

Tabel 4. 4. Data Pengujian Pompa *Dosing*

Perbobaan ke-	**Pengaturan Kontroler (ml)**	**Volume Larutan (ml)**	***Error* (%)**	**Perbobaan ke-**	**Pengaturan Kontroler (ml)**	**Volume Larutan (ml)**	***Error* (%)**
1	10	10	0	**19**	100	97	3
2	10	11	10	**20**	100	97	3
3	20	20	0	**21**	110	109	0,91
4	20	21	5	**22**	110	108	1,81
5	30	31	3,33	**23**	120	117	2,5
6	30	31	3,33	**24**	130	127	2,31
7	40	41	2,5	**25**	140	136	2,86
8	40	41	2,5	**26**	150	146	2,67
9	50	50	0	**27**	160	156	2,5
10	50	50	0	**28**	170	166	2,35
11	60	61	1,67	**29**	180	187	3,89
12	60	60	0	**30**	190	186	2,11
13	70	69	1,43	**31**	200	196	2
14	70	69	1,43	**32**	210	205	2,38
15	80	78	2,5	**33**	220	216	1,81
16	80	78	2,5	**34**	230	224	2,61
17	90	88	2,22	**35**	240	236	1,67
18	90	89	1,11	**36**	250	245	2
Rata – Rata *Error*							2,219

Dari percobaan yang dilakukan, masih terjadi *error* saat pemberian cairan. Volume cairan yang diberikan tidak sesuai dengan pengaturan kontroler. *Syringe* yang digunakan tidak tepat mengambil cairan sebanyak 10 ml setiap putarannya, karena terjadi *error* saat pengukuran diameter penarik *syringe.* Error yang terjadi juga disebabkan oleh cairan yang berpindah – pindah tempat sebelum pengukuran (untuk pengukuran volume diatas 50 ml). Dari pengujian yang dilakukan, didapatkan rata – rata *error* dalam pemberian cairan sebesar 2,219 %.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

4.2. Hasil Data Pengujian dan Pembahasan

Dari alat kontrol yang telah dibuat, dilakukan pengujian kinerja alat kontrol terhadap akuarium yang telah disiapkan. Hal ini guna mengetahui perbedaan parameter – parameter akuarium sebelum menggunakan alat kontrol, dengan kondisi setelah menggunakan alat kontrol.

4.2.1 Hasil Pengujian Pengaturan *Default*

Pengaturan kontroler yang digunakan pada mode *default* mengacu pada Tabel 3.1. Dari pengujian yang dilakukan, diperoleh data kinerja kontroler sebagai berikut:

Tabel 4. 5. Data Hasil Pengujian Kontroler pada Mode *Default*

Kontrol	Parameter	Nilai Pengaturan	Hasil Pengujian	*Error*
Pencahayaan Kanal 1 - 6	Waktu *sunrise*	07.00	Lampu mulai menyala pada 07:00:30 (intensitas 1%)	30 detik
	Delay sunrise	120 menit	Lampu mencapai intensitas maksimum (100%) pada 08:59:23	37 detik
	Waktu *sunset*	17.00	Lampu mulai berkurang intensitasnya pada 17:00:37 (intensitas 99%)	37 detik
	Delay sunset	120 menit	Lampu mati (intensitas 0%) pada 18:59:26	34 detik
Penambahan Bahan Aditif Kanal 1	Mode	*Float switch normal*	Pompa *dosing* akan aktif setiap kali air pada ruang filter terakhir turun sejauh 1 cm dari kondisi normal	Tidak ada *error*
	Volume	100 ml		
Penambahan Bahan Aditif Kanal 2	Mode	*Float switch* pengurasan	Pompa return akan mati setiap kali air pada ruang filter terakhir turun sejauh 11 cm dari kondisi normal	Tidak ada *error*
Suhu	*Setpoint* suhu	23,5 ^{o}Celcius	*Chiller* menyala pada suhu 23,625 ^{o}C dan mati pada suhu 23,375 ^{o}C	Tidak ada *error*

Pada kontrol cahaya masih terjadi error, namun error yang terjadi tidak berpengaruh terhadap pengendalian karena masih dalam orde detik. Dari data di atas, dapat disimpulkan bahwa pengaturan *default* alat kontrol sudah dapat bekerja sesuai dengan perancangan.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

4.2.2 Aplikasi Kontroler pada Akuarium

Kontroler yang telah dibuat, digunakan pada akuarium yang telah disiapkan. Hal ini guna menguji kinerja kontroler dan mengetahui dampak dari digunakannya kontroler ini pada sistem akuarium laut. Pada saat pengujian, kontroler menggunakan pengaturan sebagai berikut:

Tabel 4. 6. Pengaturan Kontroler pada Akuarium

Kontrol Pencahayaan	Kanal 1 – 3 (LED Putih)	Waktu *sunrise* = 11.30 *Delay sunrise* = 120 menit Waktu *sunset* = 20.00 *Delay sunset* = 180 menit *Peak* = 100%
	Kanal 4 – 6 (LED Biru)	Waktu *sunrise* = 11.30 *Delay sunrise* = 120 menit Waktu *sunset* = 21.00 *Delay sunset* = 150 menit *Peak* = 100%
Kontrol Penambahan Bahan Aditif	Kanal 1	Mode = *Disconnect* Volume = -
	Kanal 2	Mode = *float switch* pengurasan Volume = -
	Kanal 3	Mode = *float switch normal* Volume = 80 ml air akuades
	Kanal 4	Mode = *timer* (pada pukul 00:00; 04:00; 08:00; 12:00; 16:00; 20:00) Volume = 100 ml *kalkwasser* (sampai hari ketiga) Volume = 50ml *kalkwasser* (hari keempat – terakhir)
Kontrol Suhu		*Setpoint* suhu = 24,5^{o} Celcius

Dengan pengaturan seperti di atas, diperoleh data kinerja kontroler sebagai berikut:

Tabel 4. 7. Hasil Kinerja Kontroler

Kontrol	Parameter	Nilai Pengaturan	Hasil Pengujian	Error
Pencahayaan Kanal 1 - 3	Waktu *sunrise*	11.30	Lampu mulai menyala pada 11:30:36 (intensitas 1%)	36 detik
	Delay sunrise	120 menit	Lampu mencapai intensitas maksimum (100%) pada 15:29:24	36 detik
	Waktu *sunset*	20.00	Lampu mulai berkurang intensitasnya pada 20:00:56 (intensitas 99%)	56 detik
	Delay sunset	180 menit	Lampu mati (intensitas 0%) pada 18:59:08	52 detik

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 4. 7. (Lanjutan) Hasil Kinerja Kontroler

Kontrol	Parameter	Nilai Pengaturan	Hasil Pengujian	Error
Pencahayaan Kanal 4 - 6	Waktu *sunrise*	11.30	Lampu mulai menyala pada 11:30:36 (intensitas 1%)	36 detik
	Delay sunrise	120 menit	Lampu mencapai intensitas maksimum (100%) pada 15:29:24	36 detik
	Waktu *sunset*	21.00	Lampu mulai berkurang intensitasnya pada 21:00:46 (intensitas 99%)	46 detik
	Delay sunset	150 menit	Lampu mati (intensitas 0%) pada 23:29:16	44 detik
Penambahan Bahan Aditif Kanal 2	Mode	*Float switch* pengurasan	Pompa return akan mati setiap kali air pada ruang filter terakhir turun sejauh 11 cm dari kondisi normal	Tidak ada *error*
Penambahan Bahan Aditif Kanal 3	Mode	*Float switch normal*	Pompa *dosing* akan aktif setiap kali air pada ruang filter terakhir turun sejauh 1 cm dari kondisi normal	Tidak ada *error*
	Volume	80 ml		
Penambahan Bahan Aditif Kanal 4	Mode	*Timer* (pada pukul 00:00; 04:00; 08:00; 12:00; 16:00; 20:00)	Pompa dosing aktif pada pukul: 00:00:00; 04:00:00; 08:00:00; 12:00:00; 16:00:00; 20:00:00;	Tidak ada *error*
	Volume	100 ml (hari pertama sampai ketiga) 50 ml (hari keempat sampai terakhir)		
Suhu	*Setpoint* suhu	24,5 ^{o}Celcius	*Chiller* menyala pada suhu 24,625 ^{o}C dan mati pada suhu 24,375 ^{o}C	Tidak ada *error*

Error yang terjadi pada pengendali cahaya tidak berpengaruh terhadap kinerja kontroler, karena *error* yang terjadi masih dalam orde detik. Dari data di atas, dapat diketahui bahwa alat kontrol yang dibuat sudah dapat bekerja sesuai pengaturan yang diinginkan pengguna.

Untuk mengetahui efek yang terjadi pada parameter air akuarium, dilakukan pengujian parameter – parameter akuarium. Parameter yang diuji antara lain kadar kalsium, kadar karbonat, suhu, dan kadar garam. Berikut merupakan data hasil pengujian parameter akuarium dengan menggunakan kontroler:

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 4. 8. Data Parameter Akuarium Dengan Kontroler

Hari ke-	Kadar Kalsium (ppm)	Kadar Karbonat (dKH)	Suhu (°C)	Kadar Garam	Keterangan
1	390	11,2	24,625	1,026	*Dosing kalkwasser* 100ml x 6 perhari
3	400	11,2	24,5	1,026	
4	400	15,4	24,625	1,026	*Dosing kalkwasser* 50ml x 6 perhari
5	400	14	24,625	1,027	
6	390	11,2	24,5	1,027	
7	405	14	24,625	1,027	
8	405	11,2	24,5	1,027	
9	400	12,6	24,5	1,026	
10	400	14	24,5	1,027	
11	410	14	24,625	1,027	

Pada hari pertama, dilakukan pengujian terhadap parameter air akuarium, dan didapatkan nilai kadar kalsium berada pada *level* 390 ppm. Sehingga kalsium yang harus ditambahkan setiap harinya adalah sebesar:

$$N_{Ca,diff} = \frac{(Ca_{Ca,required} - Ca_{Ca,aquarium})}{n} * V_{aquarium}$$

$$N_{Ca,diff} = \frac{(400 - 390)}{1} * 40$$

$$N_{Ca,diff} = 400\,mg$$

Dari perhitungan di atas, maka diperoleh volume larutan *kalkwasser* yang harus di buat setiap harinya yaitu sebanyak:

$$V_{kalkwasser}(l) = {}^{N_{Ca,total}}/_{0,68}$$

$$V_{kalkwasser}(l) = {}^{0,400}/_{0,68}$$

$$V_{kalkwasser} = 0{,}588\,liter = 588\,ml = 98\,ml\ x\ 6$$

Dari perhitungan di atas, maka pada kontroler diatur agar kontroler memberikan larutan *kalkwasser* sebanyak 100 ml setiap empat jam sekali setiap harinya. Hal ini dilakukan setiap hari sampai dengan pengujian hari keempat. Karena kadar karbonat dalam air sudah mencapai 15,4 dKH, maka dosis *kalkwasser* yang diberikan dikurangi menjadi 50 ml setiap empat jam sekali setiap harinya. Dari percobaan yang dilakukan ini, didapatkan rata – rata kadar kalsium dan karbonat sebesar:

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

$$\bar{x}_{kalsium} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{4000}{10} = 400\ ppm$$

$$\bar{x}_{karbonat} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{128{,}8}{10} = 12{,}88\ dKH$$

Dari rata – rata data di atas, dapat dicari simpangan dan persentase simpangan data yang terjadi terhadap nilai rata – rata data, yaitu sebesar:

$$S = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

$$S_{kalsium} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} = \sqrt{\frac{350}{9}} = 6{,}34\ ppm$$

$$S_{karbonat} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} = \sqrt{\frac{22{,}736}{9}} = 1{,}59\ dKH$$

$$\%S = \frac{S}{\bar{x}} * 100\%$$

$$\%S_{kalsium} = \frac{S}{\bar{x}} * 100\% = \frac{6{,}34}{400} * 100\% = 1{,}59\ \%$$

$$\%S_{karbonat} = \frac{S}{\bar{x}} * 100\% = \frac{1{,}59}{12{,}88} * 100\% = 12{,}34\ \%$$

Untuk menjaga nilai kadar garam, digunakan kontroler penambahan bahan aditif mode *float switch.* Setiap kali float switch mendeteksi adanya penurunan *level* air di ruang filter, maka pompa *dosing* akan memberikan air sebanyak 80 ml. Sedangkan untuk menjaga suhu tetap stabil, kontroler diatur pada suhu 24,5 ^{o}C dan pada *thermostat chiller* diatur pada suhu 24 ^{o}C. Dengan cara yang sama seperti perhitungan persentase simpangan kadar kalsium dan kadar karbonat, diperoleh data simpangan yang terjadi dari masing – masing parameter akuarium sebagai berikut:

Tabel 4. 9. Persentase Simpangan yang Terjadi pada Parameter Air Akuarium

	Kadar Kalsium	**Kadar Karbonat**	**Suhu**	**Kadar Garam**
Rata – rata	400 ppm	12,88 dKH	24,57 ^{o}C	1,0266
Simpangan	6,34 ppm	1,59 dKH	0,082 ^{o}C	0,52x10^{-3}
Persentase simpangan	1,59 %	12,34 %	0,33 %	0,05 %

Dari data di atas, dapat diketahui tingkat kestabilan masing – masing parameter air akuarium setelah menggunakan alat kontrol. Hanya simpangan kadar karbonat yang masih berada di atas 5 %, sedangkan parameter air akuarium yang lain mempunyai simpangan di bawah 5 %.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

4.2.3 Kontrol Cahaya

Untuk mengetahui kinerja kontroler cahaya, dilakukan pengujian dengan mengukur tegangan LED. Pengukuran dilakukan hanya pada *driver LED* kanal ketiga. Berikut merupakan data hasil pengujian *driver LED* kanal ketiga:

Tabel 4. 10. Data Tegangan LED

Intensitas Kontroler (%)	**Tegangan LED (volt)**	**Tegangan LED (%)**	***Error* (%)**
5	3,2	-	-
10	3,8	10,42	0,42
15	4,07	15,1	0,1
20	4,4	20,83	0,83
25	4,7	26,04	1,04
30	5,02	31,59	1,59
35	5,29	36,29	1,29
40	5,63	42,19	2,19
45	5,91	47,05	2,05
50	6,22	52,43	2,43
55	6,65	59,89	4,89
60	6,94	64,93	4,93
65	7,2	69,44	4,44
70	7,45	73,79	3,79
75	7,72	78,47	3,47
80	7,96	82,64	2,64
85	8,21	86,98	1,98
90	8,49	91,84	1,84
95	8,7	95,49	0,49
100	8,96	100	0
Rata – rata *Error*			2,12

Pengambilan data dimulai dengan membandingkan persentase intensitas cahaya yang ditunjukkan oleh kontroler dan tegangan LED yang dihasilkan. Dari data tegangan pada intensitas 5 % sampai dengan 100 %, dapat dicari persentase tegangan LED dengan menggunakan rumus:

$$\%_{Tegangan\ LED} = \left(\frac{V_{x\%} - V_{5\%}}{V_{100\%} - V_{5\%}}\right) 100\%$$

Sebagai contoh, perhitungan persentase tegangan LED pada intensitas 45% sebagai berikut:

$$\%_{Tegangan\ LED} = \left(\frac{V_{x\%} - V_{5\%}}{V_{100\%} - V_{5\%}}\right) 100\%$$

$$\%_{Tegangan\ LED} = \left(\frac{5{,}91 - 3{,}2}{8{,}96 - 3{,}2}\right) 100\%$$

$$\%_{Tegangan\ LED} = \left(\frac{2{,}71}{5{,}76}\right) 100\%$$

$$\%_{Tegangan\ LED} = 47{,}05\ \%$$

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Dari data persentase tegangan LED, dapat diketahui perbandingan antara persentase tegangan LED dengan persentase intensitas yang dikehendaki kontroler. Dari perhitungan data – data di atas, didapat rata – rata *error* sebesar 2,12 %.

4.2.4 Kadar Ca dan KH

Untuk mengetahui nilai kadar kalsium dan karbonat akuarium, dilakukan pengujian dengan menggunakan *test kit* kalsium dan karbonat. *Test kit* kalsium *Salifert*® dan *test kit* karbonat *Seachem*® digunakan dalam pengujian ini. *Test kit* kalsium *Salifert*® mempunyai tingkat ketelitian lima ppm dengan rentang data antara nol ppm sampai dengan 500 ppm. Sedangkan *test kit* karbonat *Seachem*® mempunyai tingkat ketelitian 1,4 dKH dengan rentang data antara . Berikut gambar test kit yang digunakan:

Gambar 4. 9. *Test Kit* Kalsium

Gambar 4. 10. *Test Kit* Karbonat

Dari percobaan yang dilakukan, diperoleh data kadar kalsium dan karbonat sebagai berikut:

Tabel 4. 11. Tabel Kadar Kalsium (Ca) dan Karbonat (KH)

Hari ke-	Kadar Ca (ppm)	Kadar KH (dKH)	Keterangan
Hari ke-1	400	8,4	*uncontrolled*
Hari ke-5	385	8,4	Mulai *dosing* 20ml*6 perhari *(controlled)*
Hari ke-10	380	8,4	Mulai *dosing* 60ml*6 perhari *(controlled)*
Hari ke-15	410	9,8	Mulai *dosing* 100ml*6 perhari *(controlled)*
Hari ke-20	410	12,6	*Stop dosing (controlled)*
Rata - rata	397	9,52	

Pada hari pertama sampai kelima, tidak dilakukan penambahan bahan aditif sama sekali. Hal ini agar rasio penurunan kadar kalsium pada akuarium dapat diketahui. Dari hasil tes kadar kalsium dan karbonat pada hari kelima, didapat rasio penurunan kadar kalsium akuarium adalah sebesar:

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

$$N_{Ca,t} = \frac{(Ca_{Ca,hari=0} - Ca_{Ca,hari=n})}{n}$$

$$N_{Ca,t} = \frac{(400 - 385)}{5}$$

$$N_{Ca,t} = 3\ mg$$

Rasio penurunan ini hanya menunjukkan penurunan kadar kalsium setiap harinya. Untuk mengembalikan kadar kalsium ke nilai awal (400 ppm), perlu ditambahkan kalsium sebanyak:

$$N_{Ca,diff} = \frac{(Ca_{Ca,required} - Ca_{Ca,aquarium})}{n} * V_{aquarium}$$

$$N_{Ca,diff} = \frac{(400 - 385)}{1} * 40$$

$$N_{Ca,diff} = 600\ mg$$

Total kalsium yang harus diberikan adalah sebanyak:

$$N_{Ca,total} = N_{Ca,t} + N_{Ca,diff}$$

$$N_{Ca,total} = 3 + 600$$

$$N_{Ca,total} = 603\ mg$$

Dari perhitungan di atas, maka diperoleh kadar volume larutan *kalkwasser* yang harus di buat setiap harinya yaitu sebanyak:

$$V_{kalkwasser}(l) = {}^{N_{Ca,total}}/_{0,68}$$

$$V_{kalkwasser}(l) = {}^{0,603}/_{0,68}$$

$$V_{kalkwasser} = 0{,}879\ liter = 879\ ml = 146{,}5\ ml\ x\ 6$$

Karena larutan *kalkwasser* harus diberikan secara perlahan, maka penambahan larutan ini dilakukan sedikit demi sedikit dimulai dengan dosis yang kecil yaitu 20 ml setiap empat jam sekali (enam kali sehari). Dosis awal ini dikontrol melalui kontroler dan dilakukan terus menerus selama lima hari. Pada hari ke-10 dilakukan pengetesan kadar kalsium, dan diperoleh kadar kalsium yang menurun dari hari kelima sedangkan kadar karbonat masih stabil. Pada hari ini juga dilakukan penambahan dosis pemberian larutan *kalkwasser* yaitu menjadi sebanyak 60 ml setiap empat jam sekali (enam kali sehari). Pada hari ke-15 dilakukan pengetesan kadar kalsium, diperoleh kadar kalsium dan kadar karbonat yang naik menjadi 410 ppm dan 9,8 dKH. Karena dosis yang sesuai perhitungan belum tercapai, maka penambahan dosis larutan *kalkwasser* tetap dilakukan, menjadi 100

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

ml setiap empat jam sekali (enam kali sehari). Pada hari ke-20 dilakukan pengetesan kadar kalsium, dan diperoleh kadar kalsium yang stabil namun kadar karbonat naik menjadi 12,6 dKH. Karena kadar karbonat telah naik terlalu tinggi, maka penambahan bahan aditif dihentikan agar tidak berdampak buruk terhadap biota yang dipelihara.

Dari tabel 4.11 dapat dicari nilai simpangan kadar kalsium dan kadar karbonat, yaitu sebesar:

$$S = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

$$S_{Ca} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} = \sqrt{\frac{780}{4}} = 13{,}96$$

$$S_{KH} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} = \sqrt{\frac{13{,}328}{4}} = 1{,}83$$

Dari perhitungan di atas, dapat dicari perbandingan simpangan yang terjadi terhadap nilai rata – rata data, yaitu sebesar:

$$\%S = \frac{S}{\bar{x}} * 100\%$$

$$\%S_{Ca} = \frac{S}{\bar{x}} * 100\% = \frac{13{,}96}{397} * 100\% = 3{,}52\%$$

$$\%S_{KH} = \frac{S}{\bar{x}} * 100\% = \frac{1{,}83}{9{,}52} * 100\% = 19{,}2\%$$

Dari perhitungan di atas, dapat diketahui bahwa tingkat kestabilan kadar kalsium dan kadar karbonat setelah menggunakan alat kontrol adalah sebesar 3,53% dan 19,2%. Simpangan kadar karbonat masih cukup tinggi. Hal ini disebabkan alat tes yang digunakan kurang akurat dalam mengukur kadar karbonat yang ada. Alat tes yang digunakan untuk mengukur kadar karbonat hanya mempunyai ketelitian sebesar 1,4 dKH (12,5 %), sedangkan rentang data yang mungkin terukur adalah antara 5,6 – 16,8 dKH (kadar karbonat normal pada air laut [4]).

4.2.5 Kadar Garam

Pengambilan data kadar garam dilakukan secara terpisah dengan pengambilan data kadar kalsium dan karbonat. Hal ini dilakukan agar pengambilan data kadar garam tidak terganggu dengan adanya penambahan cairan *kalkwasser*. Cairan *kalkwasser* yang ditambahkan dapat mempengaruhi perubahan kadar garam di akuarium [2].

Untuk mengetahui nilai kadar garam yang ada pada akuarium, digunakan alat *hydrometer*. *Hydrometer* yang digunakan berbentuk tabung yang di dalamnya terdapat suatu pemberat, dan di atasnya terdapat skala pembacaan nilai kadar garam. Alat ukur yang

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

digunakan mempunyai ketelitian sampai dengan 0,001. Berikut merupakan gambar *hydrometer:*

Gambar 4. 11. *Hydrometer*

Berikut merupakan hasil pengambilan data yang dilakukan:

Tabel 4. 12. Tabel Kadar Garam Akuarium

Hari ke-	Jam	Kadar Garam	Keterangan
1	12:55	1,023	Tanpa kontroler
	12:55	1,023	
	13:55	1,023	
	14:55	1,024	
	15:55	1,024	
	16:55	1,024	
	17:55	1,024	
	18:55	1,024	
	21:45	1,024	
2	12:10	1,025	
	14:20	1,025	
	16:18	1,025	
	18:18	1,025	
	20:11	1,025	
3	11:06	1,025	Mulai mengaktifkan alat control
	12:07	1,023	
	14:09	1,023	
	17:22	1,024	
	20:07	1,024	
4	10:12	1,024	
	14:04	1,024	
	16:33	1,024	
	18:09	1,024	
	20:40	1,024	

Pengambilan data dimulai dengan pengambilan data kadar garam akuarium tanpa menggunakan kontroler. Air akuarium sengaja dibiarkan menguap, sehingga kadar garam yang ada menjadi meningkat. Kadar garam yang semula terbaca 1,023, berangsur – angsur

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

naik hingga pada hari kedua pukul 20:11 nilai kadar garam yang terbaca bernilai 1,025. Pada hari ketiga, kontroler mulai diaktifkan sehingga kadar garam yang semula bernilai 1,025 kembali turun ke 1,023. Kadar garam akuarium kemudian stabil di *level* 1,024.

Dari data di atas, dapat dicari rata – rata kadar garam selama menggunakan kontroler, dan sebelum menggunakan kontroler, yaitu sebesar:

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

$$\bar{x}_{non-control} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{15{,}362}{15} = 1{,}024$$

$$\bar{x}_{control} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{9{,}214}{9} = 1{,}02378$$

Dari rata – rata data di atas, dapat dicari simpangan dan persentase simpangan data yang terjadi terhadap nilai rata – rata data, yaitu sebesar:

$$S = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

$$S_{non-control} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} = \sqrt{\frac{11{,}73.10^{-6}}{14}} = 0{,}92.10^{-3}$$

$$S_{control} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} = \sqrt{\frac{1{,}56.10^{-6}}{8}} = 0{,}44.10^{-3}$$

$$\%S = \frac{S}{\bar{x}} * 100\%$$

$$\%S_{non-control} = \frac{S}{\bar{x}} * 100\% = \frac{0{,}92.10^{-3}}{1{,}024} * 100\% = 0{,}089\ \%$$

$$\%S_{control} = \frac{S}{\bar{x}} * 100\% = \frac{0{,}44.10^{-3}}{1{,}02378} * 100\% = 0{,}043\ \%$$

Dari perhitungan di atas, dapat diketahui bahwa setelah menggunakan alat kontrol, kestabilan kadar garam meningkat hingga 0,046 %.

4.2.6 Suhu

Pengambilan data suhu menggunakan bantuan dari sensor suhu *DS18B20* yang telah terpasang pada alat kontrol akuarium. Pengambilan data suhu dimulai dengan data tanpa kontroler, namun *chiller* dalam keadaan menyala. Pada *chiller* sudah terdapat *thermostat* yang berfungsi untuk mengatur suhu yang diinginkan, namun *thermostat* yang ada hanya dapat mengatur suhu dengan kelipatan satu derajat *Celcius*.

Pengambilan data suhu dilakukan sesaat setelah *chiller* berubah dari kondisi *on* ke *off* ataupun sebaliknya. Hal ini dilakukan agar diketahui lamanya waktu *on* dan waktu *off*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

chiller baik dengan kontroler ataupun tanpa kontroler. Sebelum dilakukan pengambilan data, kondisi *chiller* telah berada pada kondisi yang stabil (*steady state*).

Berikut merupakan data hasil pengujian kontroler suhu:

Tabel 4. 13. Tabel Data Suhu Akuarium

Tanpa Kontroler				**Dengan Kontroler**		
Menit ke-	**Suhu (^oC)**	***Chiller***		**Menit ke-**	**Suhu (^oC)**	***Chiller***
0	24,875	**Off – 7 mnt**		0	24,375	**Off – 8 mnt**
7	25,125	**On – 5 mnt**		8	24,625	**On – 6 mnt**
12	25	**Off – 6 mnt**		14	24,375	**Off – 9 mnt**
18	25,125	**On – 7 mnt**		*23*	24,625	**On – 6 mnt**
25	24,875	**Off – 7 mnt**		29	24,375	**Off – 8 mnt**
32	25,125	**On – 7 mnt**		37	24,625	**On – 5 mnt**
39	24,875	**Off – 6 mnt**		42	24,375	**Off – 10 mnt**
45	25,125	**On – 6 mnt**		*52*	24,625	**On – 6 mnt**
51	25	**Off – 7 mnt**		58	24,375	**Off – 9 mnt**
58	25,125	**On – 6 mnt**		67	24,625	**On – 6 mnt**
64	24,875	**Off – 7 mnt**		73	24,375	**Off – 9 mnt**
71	25,125	**On – 5 mnt**		82	24,625	**On – 5 mnt**
76	25	**Off – 7 mnt**		87	24,375	**Off – 10 mnt**
83	25,125	**On – 6 mnt**		97	24,625	**On – 5 mnt**
89	25	**Off – 7 mnt**		102	24,375	**Off – 10 mnt**
96	25,125	**On – 5 mnt**		112	24,625	**On – 5 mnt**
101	24,875	**Off**		117	24,375	**Off**

Pada pengujian tanpa kontroler, *thermostat chiller* diatur pada suhu 25 ^oC. Sedangkan pada pengujian dengan kontroler, *thermostat chiller* diatur pada suhu 24 ^oC dengan kontroler diatur pada suhu 24,5 ^oC. Pengatur suhu pada kontroler dapat diatur dengan kelipatan 0,125 ^oC.

Dari data di atas, dapat diketahui total waktu *on* dan waktu *off chiller* ketika menggunakan kontroler dan tanpa kontroler, yaitu sebagai berikut:

Tabel 4. 14. Perbandingan Waktu *On* dan Waktu *Off Chiller* dengan Kontroler dan Tanpa Kontroler

	Tanpa Kontroler	**Dengan Kontroler**
Total waktu *On Chiller*	47 menit	44 menit
Total waktu *Off Chiller*	54 menit	73 menit
Rata – rata waktu *On Chiller*	5,875 menit	5,5 menit
Rata – rata waktu *Off Chiller*	6,75 menit	9,125 menit
Rata – rata suhu	25,02 ^oC	24,49 ^oC
Setpoint	25 ^oC	24,5 ^oC

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Dari data di atas dapat diketahui, bahwa dengan kontroler, waktu kerja *chiller* menjadi semakin kecil. Pada jumlah data yang sama, waktu kerja *chiller* lebih sedikit apabila menggunakan kontroler. Ditambah lagi dengan kontroler, suhu yang diatur bisa lebih bervariasi, karena dapat diatur dengan kelipatan 0,125 ^{o}C.

4.3. Pembahasan Perangkat Lunak

Program yang dibuat pada perancangan ini terdiri dari beberapa subrutin dan mengacu pada gambar *flow chart* di Bab sebelumnya. Program ini sudah berfungsi seperti yang diharapkan. Berikut merupakan penjelasan lengkap dari program yang dibuat:

4.3.1 Inisialisasi

Inisialisasi dimulai dengan memasukkan *library* fitur – fitur yang digunakan dalam program, seperti *library* ATmega128, LCD, I2C, dan seterusnya. Pada inisialisasi LCD, I2C dan *One-wire Communication* menggunakan fitur *Code Wizard AVR* yang ada pada *software Code Vision AVR*, sehingga kode program akan terinisialisasi secara otomatis.

Tabel 4. 15. Parameter pada Memori *EEPROM*

Subsistem	Parameter	Fungsi
Suhu	settemp	Berisi pengaturan set point suhu
Pencahayaan	risetm	Berisi pengaturan waktu dimulainya simulasi *sunrise* (jam:menit)
	falltm	Berisi pengaturan waktu dimulainya simulai *sunset* (jam:menit)
	drise	Berisi pengaturan lama simulasi *sunrise* berlangsung (jam:menit)
	dfall	Berisi pengaturan lama simulasi *sunset* berlangsung (jam:menit)
	hiti	Berisi pengaturan lama simulasi *sunrise* dalam menit
	hitd	Berisi pengaturan lama simulasi *sunset* dalam menit
Penambahan Bahan Aditif	mode	Berisi pengaturan mode yang digunakan (*timer* atau *float switch*)
	method	Berisi pengaturan mode *floatswitch* yang digunakan (normal atau pengurasan)
	voldose	Berisi pengaturan volume cairan yang harus diberikan
	wdose	Berisi pengaturan waktu dimulainya pemberian bahan aditif (jam:menit)

Pada bagian ini, program juga akan mengambil variabel – variabel dari memori *EEPROM* yang dibutuhkan dalam pengendalian masing – masing subsistem. Variabel – variabel yang digunakan untuk menyimpan pengaturan pengguna disimpan dalam struktur

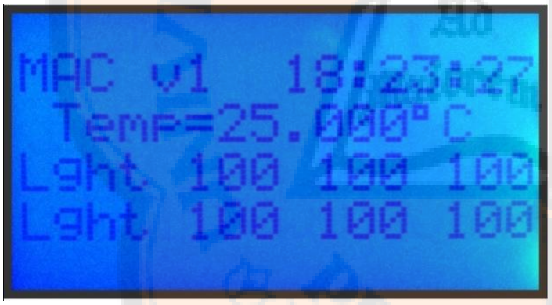
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

data sehingga lebih mudah untuk digunakan dalam program. Tabel 4.15 menunjukkan variabel – variabel yang tersimpan di memori *EEPROM.*

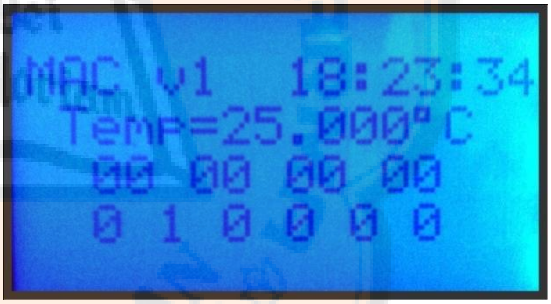
4.3.2 Program Utama

Program utama berisi alur perintah utama, yang berikutnya akan memanggil subrutin – subrutin lainnya. Perintah yang ada pada program utama ini akan dilakukan secara terus menerus. Seperti pada gambar 3.19, program utama pertama – tama akan menginisialisasi fitur – fitur yang digunakan. Setelah inisialisasi, program akan memanggil satu per satu subrutin yang ada secara terus menerus.

Pada program utama, kontroler juga akan menampilkan status pengendalian yang dilakukan. Beberapa parameter yang ditampilkan kontroler melalui LCD antara lain waktu, suhu, intensitas cahaya, jumlah putaran motor pompa *dosing*, status *relay* dan status *buzzer*. Parameter intensitas cahaya, jumlah putaran motor pompa *dosing,* status *relay* dan status *buzzer* ditampilkan secara bergantian setiap detiknya. Berikut merupakan tampilan status alat kontrol:



Gambar 4. 12. Tampilan Program Utama Saat Detik Menunjukkan Angka Ganjil



Gambar 4. 13. Tampilan Program Utama Saat Detik Menunjukkan Angka Genap

4.3.3 Subrutin *Default*

Subrutin *default* dipanggil apabila pengguna menginginkan semua pengaturan kontroler kembali ke pengaturan awal. Isi pengaturan *default* dapat dilihat di Tabel 3.1.

4.3.4 Subrutin *Light*

Subrutin *light* dipanggil setiap detik dan akan mengontrol *PWM* yang kemudian digunakan untuk mengatur intensitas cahaya yang dihasilkan *high power LED*. Simulasi *sunrise* dan *sunset* juga dilakukan dalam subrutin ini.

Subrutin *light* dimulai dengan proses inisialisasi subrutin. Proses inisialisasi subrutin berfungsi untuk menentukan bilangan gradien kenaikan intensitas cahaya dan gradien penurunan intesitas cahaya. Gradien kenaikan intensitas cahaya akan disimpan di

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

konstanta `inclight`. Sedangkan gradient penurunan intensitas cahaya akan disimpan di konstanta `declight`.

Dari gambar 3.22 yang menunjukkan diagram alir subrutin *light,* intensitas cahaya akan berubah setiap menit. Hal ini menyebabkan perubahan intensitas cahaya sangat terlihat dan masa transisinya tidak halus. Agar masa transisi terlihat menjadi lebih halus, perubahan intensitas cahaya dilakukan setiap detik.

4.3.5 Subrutin *Dose*

Subrutin *dose* dipanggil setiap dua detik sekali. Kontrol penambahan bahan aditif dilakukan dalam subrutin ini. Kontrol yang dilakukan setiap kanalnya sesuai dengan mode yang dipilih oleh pengguna, dan setiap kanal mungkin mempunyai mode yang berbeda.

Dari gambar 3.21 yang menunjukkan diagram alir subrutin *dose,* putaran motor yang mengendalikan pompa *dosing* dihitung dalam subrutin ini. Pada implementasi alat, hal ini menyebabkan proses kontrol terhenti saat pompa dosing aktif, dan kontroler tidak dapat melakukan proses lainnya. Agar hal ini tidak terjadi, ditambahkan sebuah subrutin baru yang khusus digunakan sebagai penghitung putaran motor yang mengendalikan pompa *dosing*. Subrutin ini disebut subrutin `cek_motor`.

4.3.6 Subrutin Cek Motor

Subrutin ini hanya dipanggil ketika terdapat kanal pengendali bahan aditif yang sedang aktif. Ketika terjadi perubahan logika rendah ke logika tinggi pada pin masukan mikrokontroler, maka variabel yang menyimpan jumlah putaran motor akan bertambah.

4.3.7 Subrutin Temperature

Subrutin *temperature* digunakan untuk mengontrol suhu pada akuarium. Untuk membantu mendinginkan air pada akuarium, digunakan alat tambahan berupa *chiller*. *Chiller* ini yang kemudian akan dikontrol oleh kontroler suhu. Subrutin ini dipanggil setiap tiga puluh detik sekali.

Apabila kondisi suhu yang diinginkan pengguna tidak bisa tercapai, maka *buzzer* akan menyala. *Buzzer* diharapkan akan memberitahukan kepada pengguna bahwa terjadi masalah dengan *chiller*. *Buzzer* akan mati setelah suhu yang ada di akuarium sesuai dengan pengaturan pengguna.

4.3.8 Subrutin Menu

Subrutin menu dipanggil apabila pengguna menekan tombol menu. Di dalam subrutin ini, program akan menampilkan menu yang dapat dipilih oleh pengguna dan

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

digunakan untuk mengatur kerja kontroler. Berikut merupakan gambar tampilan menu utama yang dapat dipilih pengguna:

Gambar 4. 14. Menu utama yang ditampilkan kontroler

Menu *time* digunakan untuk mengatur waktu pada kontroler. Pengguna dapat mengatur jam, menit dan detik sesuai dengan waktu yang ada. Untuk mengatur waktu yang ada saat ini, pengguna hanya tinggal memilih bagian yang ingin diatur. Dengan menekan dua digit angka, maka waktu yang ada dengan otomatis tersimpan. Berikut tampilan menu *time:*

Gambar 4. 15. Tampilan Menu *Time*

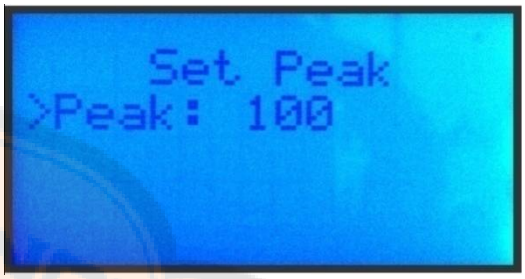
Menu *temp* digunakan untuk mengatur suhu yang diinginkan oleh pengguna. Pengguna dapat mengatur suhu yang diinginkan dengan kelipatan 0,125 ºC. Apabila pengguna memberikan masukan angka yang bukan merupakan kelipatan bilangan 0,125, maka program dengan otomatis akan menyesuaikan nilai suhu dengan bilangan kelipatan 0,125 di bawahnya. Sebagai contoh apabila pengguna mengatur suhu pada 25,3 ºC, maka program dengan otomatis menyesuaikan *setpoint* suhu pada 25,25 ºC. Berikut merupakan tampilan menu *temp:*

Gambar 4. 16. Menu *Temp*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

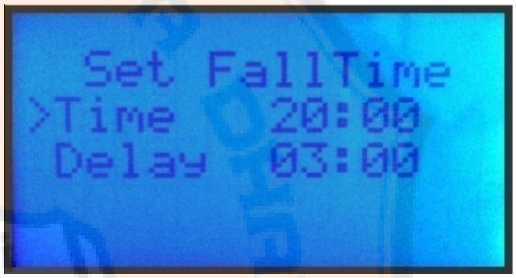
Menu *light* digunakan untuk pengaturan kontrol pencahayaan. Di dalam menu *light t*erdapat empat *frame* yang digunakan untuk mengatur pengendalian pencahayaan. Masing – masing frame berisi parameter – parameter yang dibutuhkan untuk pengendalian pencahayaan. Keterangan masing – masing parameter mengacu pada Tabel 4.15.

Gambar 4. 17. *Frame* Pertama Menu *Light*



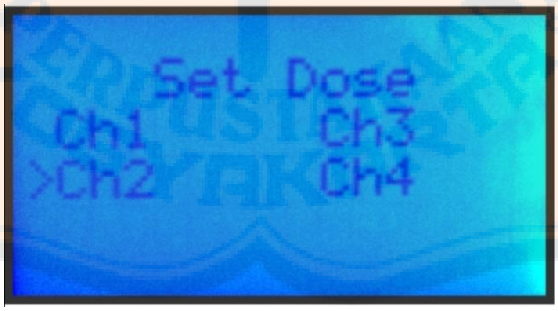
Gambar 4. 18. *Frame* Kedua Menu *Light*

Gambar 4. 19. *Frame* Ketiga Menu *Light*

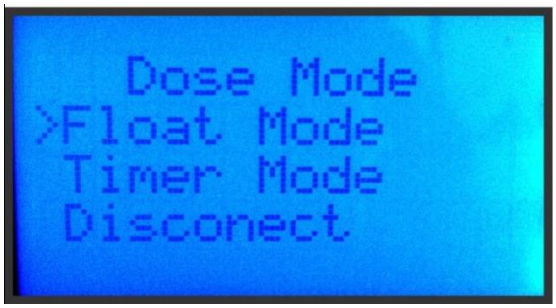


Gambar 4. 20. *Frame* Keempat Menu *Light*

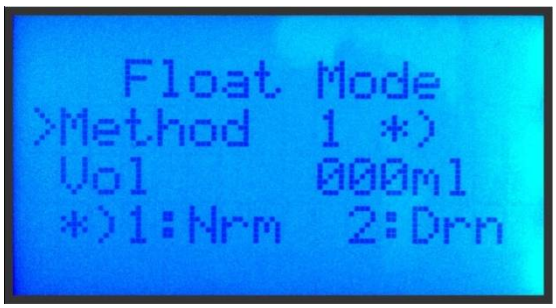
Menu *dose* digunakan untuk pengaturan kontrol penambahan bahan aditif. Di dalam menu ini, pengguna dapat mengubah mode pengendalian, besarnya volume cairan yang ingin ditambahkan, dan waktu pemberian cairan.



Gambar 4. 21. Frame Pertama Menu *Dosing*

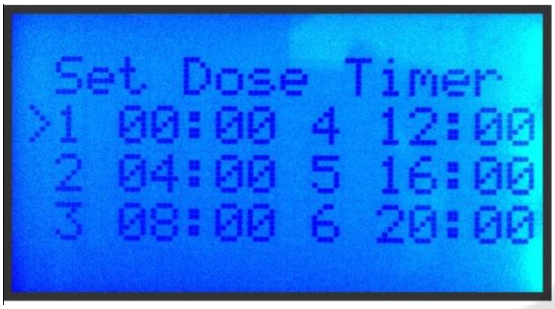


Gambar 4. 22. *Frame* Pilihan Mode Menu *Dosing*

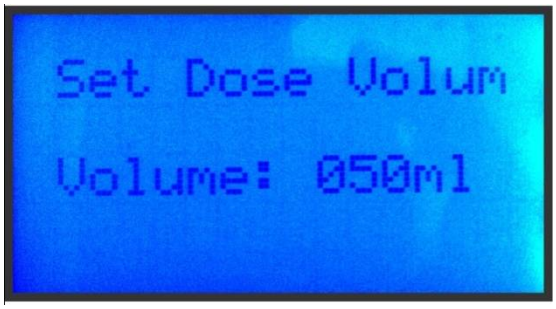


Gambar 4. 23. *Frame* Mode *Float Switch* Menu *Dosing*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Gambar 4. 24. *Frame* Pertama Mode *Timer* Menu *Dosing*



Gambar 4. 25. *Frame* Kedua Mode *Timer* Menu *Dosing*

Menu *default* digunakan untuk mengembalikan semua pengaturan, ke pengaturan awal. Sedangkan menu *sim* adalah menu tambahan, yang digunakan untuk simulasi kontrol pencahayaan.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

BAB V
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari hasil pengujian dan pengambilan data pada alat kontrol akuarium berbasis ATmega128 sebagai penjaga stabilitas akuarium laut, didapatkan kesimpulan sebagai berikut:

1. Alat kontrol yang dibuat sudah dapat bekerja sesuai dengan perancangan.
2. Alat kontrol yang dibuat sudah dapat mensimulasikan terjadinya siang dan malam di akuarium.
3. Alat kontrol yang dibuat mampu menjaga kestabilan kadar kalsium, kadar garam dan suhu dengan tingkat kestabilan 95%.
4. Alat kontrol yang dibuat mampu menjaga kestabilan kadar karbonat sampai dengan tingkat kestabilan 80%.

5.2. Saran

Saran – saran dari pengembangan alat kontrol ini selanjutnya adalah:

1. Alat tes yang digunakan sebaiknya memiliki akurasi yang baik, sehingga dalam proses pengujian alat, tidak menimbulkan *error* yang terlalu besar.
2. Penambahan sensor seperti sensor pH dan sensor kadar garam, sehingga kontroler dapat memantau secara langsung parameter pH dan kadar garam dalam air akuarium.
3. Penambahan fitur komunikasi ke komputer dan data logger, agar dapat diketahui trend parameter air akuarium.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

DAFTAR PUSTAKA

[1] Heru, Susanto, 2000, *Ikan Hias Air Laut,* Niaga Swadaya, Jakarta.

[2] Delbeek, J. C., Sprung, J., 1994, *The Reef Aquarium,* Ricordea Publishing, Florida.

[3] Leewis, R. J., Janse, M., 2008, *Advances in Coral Husbandry in Public Aquariums. Public Aquarium Husbandry Series,* Burger's Zoo, vol. 2, hal 133-142, 173-183

[4] --------, 2005, *Seachem Multitest pH and Alkalinity,* Seachem.

[5] --------, 2011, *Data sheet Microcontroller ATmega128,* Atmel.

[6] Winoto, Ardi, 2010, *Mikrokontroler AVR ATmega8/32/16/8535 dan Pemrogramannya dengan Bahasa C pada WinAVR,* Informatika, Bandung.

[7] --------, 2008, *Data sheet LCD Karakter,* Infineon.

[8] --------, ----, *SX/B On-Line Examples and Applications,* http://www.parallax.com/tabid/405/Default.aspx diakses tanggal 31 Januari 2013

[9] --------, 2008, *Data sheet DS18B20,* Maxim.

[10] Elliott, B. S., 2007, *Electromechanical Devices and Components Illustrated Sourcebook,* Mc Graw Hill, New York.

[11] Balogh, L., ----, *Design And Application Guide For High Speed MOSFET Gate Drive Circuits,* Texas Instruments.

[12] Boylestad, R., Nashelsky, L., 1999, *Electronic Devices and Circuit Theory,* Prentice Hall, New Jersey.

[13] Wiliams, T., 2005, *The Circuit Designer's Companion,* Newnes, Massachusetts.

[14] --------, 2008, *Data sheet DS1307,* Maxim.

[15] --------, 1998, *LED Theory and Application Notes,* Quantum Devices.

[16] Whitaker, Jerry C., 2005, *The Electronics Hand Book,* 2ed, Taylor & Francis, Boca Raton.

[17] --------, 2010, *DIY "Drews Doser" Peristaltic Dosing pump,* http://www.reefcentral.com/forums/showthread.php?p=16551630, diakses tanggal 1 April 2013

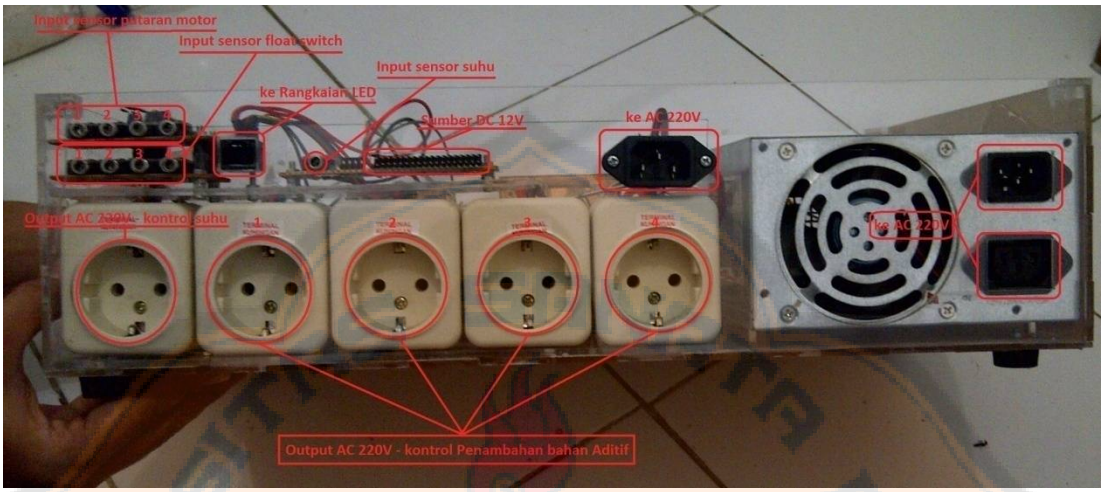
[18] --------, ----, *LEDs Aquarium Specials,* http://www.pusatled.com/index.php?route=product/category&path=78, diakses tanggal 6 April 2013.

[19] --------, 2009, *EMA-128 Minimum sistem Atmega128*, Creative Vision.

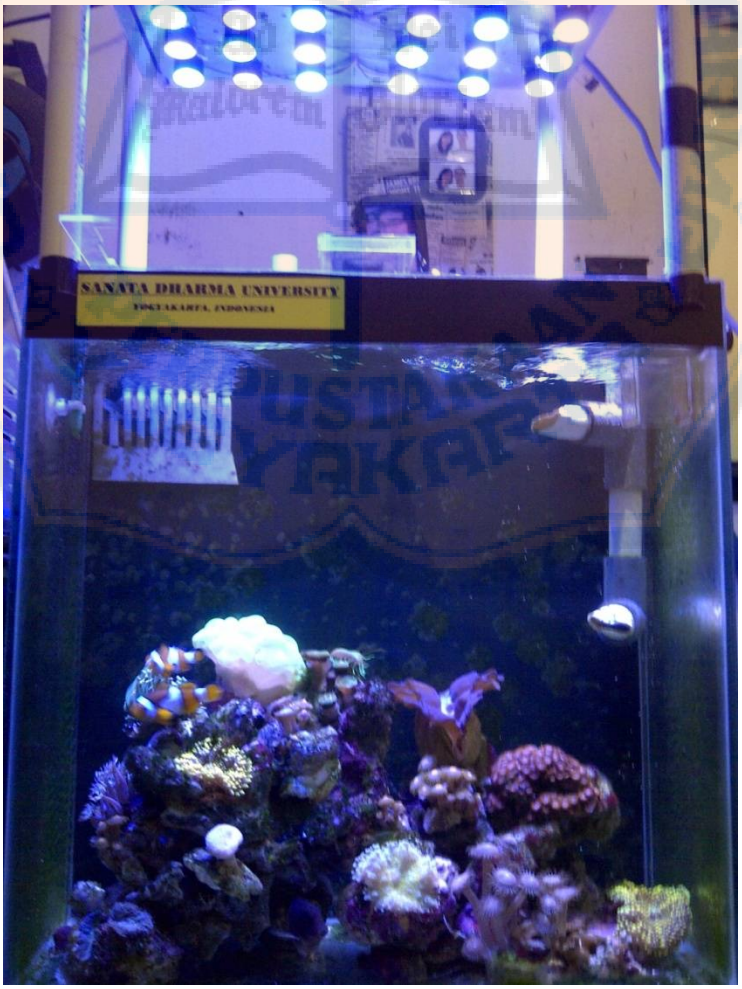
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

LAMPIRAN

***Back Panel* Kontroler**



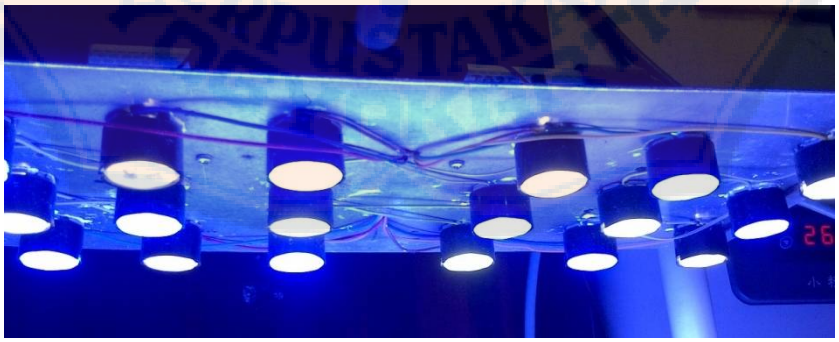
Akuarium Tampak Depan



PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

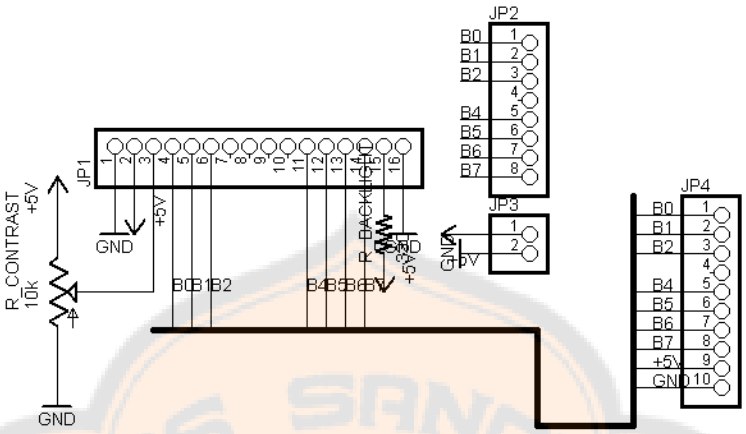
Akuarium Tampak Samping

***Fixture* LED**

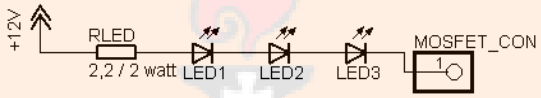


PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

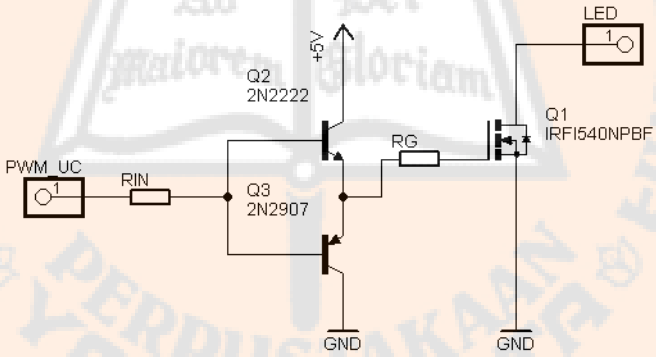
Rangkaian Skematik LCD



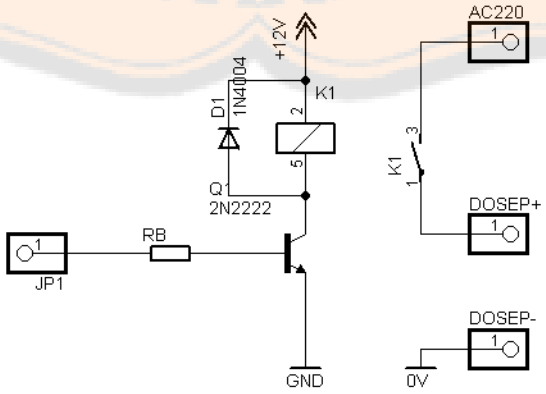
Rangkaian Skematik *High Power LED*



Rangkaian Skematik *Driver MOSFET*

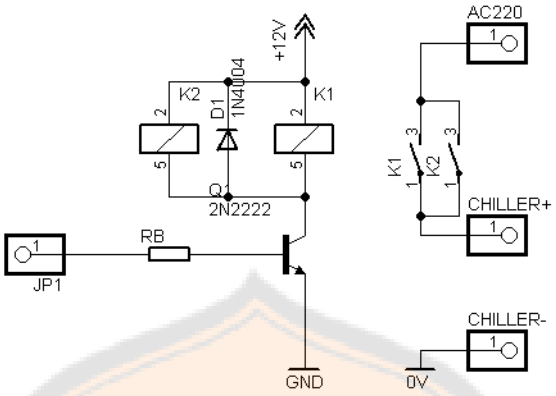


Rangkaian Skematik *Relay Dosing*

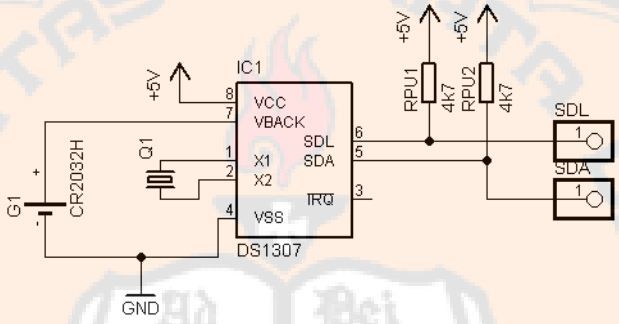


Rangkaian Skematik *Relay Chiller*

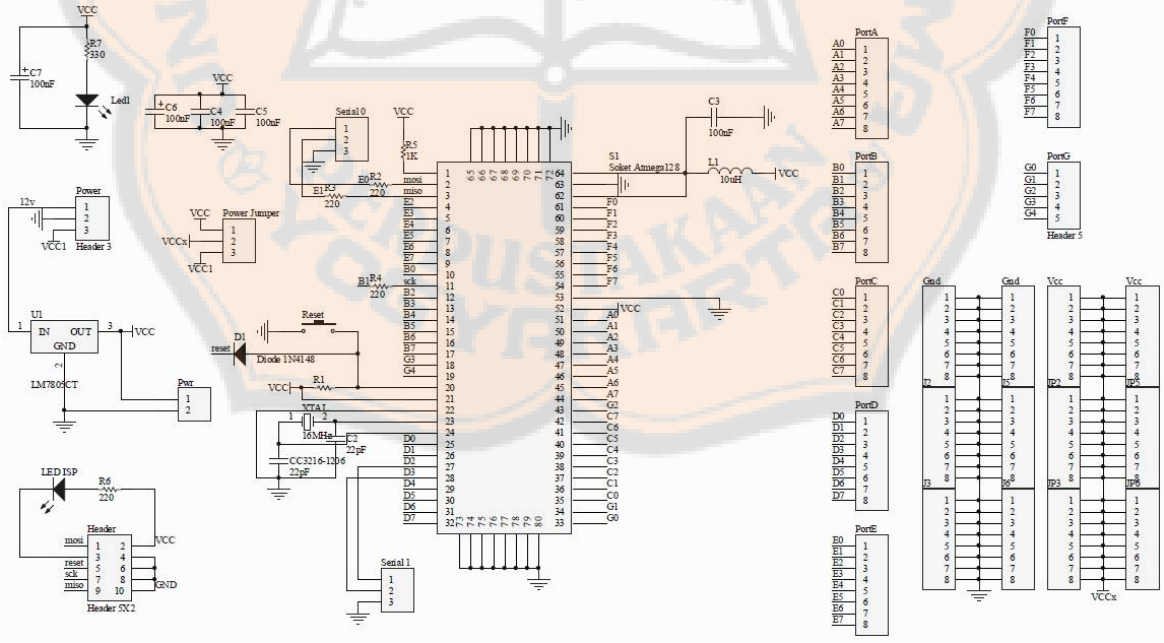
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Rangkaian Skematik RTC

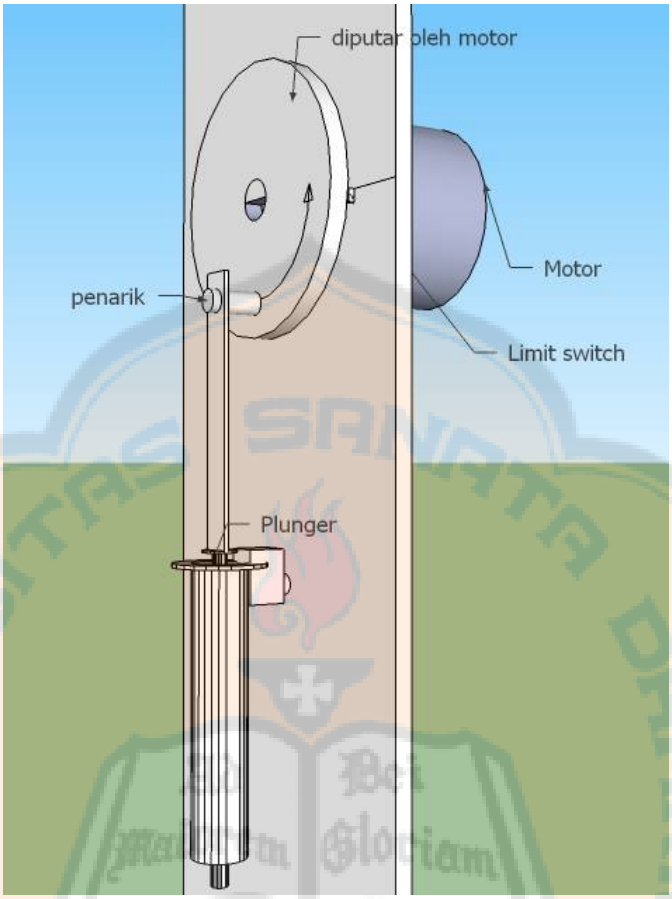


Rangkaian Minimum Sistem ATmega128

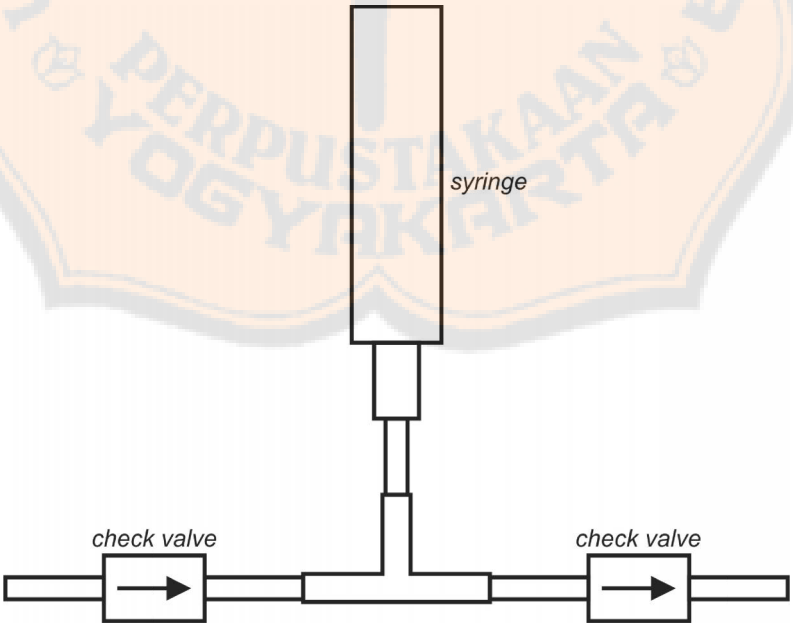


PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Perancangan *Dosing Pump* 3D



Perancangan *Dosing Pump* 2D



PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
/*****************************************************
This program was produced by the
CodeWizardAVR V2.03.9 Standard
Automatic Program Generator
© Copyright 1998-2008 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date    : 2/19/2013
Author  : Adhipa Tri Setyawan Alim
Company : USD
Comments:

Chip type               : ATmega128A
Program type            : Application
AVR Core Clock frequency: 16.000000 MHz
Memory model            : Medium
External RAM size       : 0
Data Stack size         : 1024
*****************************************************/

#include <mega128.h>
#include <delay.h>
#include <stdio.h>
#include <math.h>
#include <bcd.h>
// I2C Bus functions
#asm
   .equ __i2c_port=0x18 ;PORTB
   .equ __sda_bit=0
   .equ __scl_bit=1
#endasm
#include <i2c.h>

// DS1307 Real Time Clock functions
#include <ds1307.h>

// 1 Wire Bus functions
#asm
   .equ __w1_port=0x18 ;PORTB
   .equ __w1_bit=2
#endasm
#include <1wire.h>

// DS1820 Temperature Sensor functions
#include <ds18b20.h>

// Alphanumeric LCD Module functions
#asm
   .equ __lcd_port=0x15 ;PORTC
#endasm
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
#include <lcd.h>

unsigned char lcd[64],keypressed,navi,rom_codes[8][9],d,m,j;
unsigned char st[8]={0,0,0,0,0,0,0,0},buff[4]={0,0,0,0},lj,lm,ld;
unsigned char select,cursor[10],numb,tes[4]={0,0,0,0};
unsigned char jam,menit,detik,last_detik,n_rot[4]={0,0,0,0};
unsigned int time,sunrise[6],sunset[6];
float suhu,setsuhu,lightch[6]={0,0,0,0,0,0};
float inclight[6]={0,0,0,0,0,0},declight[6]={0,0,0,0,0,0};
//struktur data waktu
struct waktu
      {
      unsigned char hour,min;
      };
//struktur data pencahayaan
struct light
      {
      struct waktu risetm,falltm,drise,dfall;
      unsigned int hiti,hitd,peak;
      } ;
//struktur data dosing
struct dose
      {
      unsigned char mode, method;
      unsigned char voldose;
      struct waktu wdose[6];
      } ;
//parameter kontrol
struct light li[6];
struct dose dos[4];
eeprom struct light eepsetlight[6];
eeprom struct dose eepsetdose[4];
eeprom float eepsetsuhu;

// External Interrupt 7 service routine
interrupt [EXT_INT7] void ext_int7_isr(void)  //scan keypad
{
// Place your code here
      PORTA=0xff;
      PORTA.3=0;
      delay_us(10);
      if(PINA.7==0){
            navi=12;} //D
      else if(PINA.6==0){
            navi=11;} //C
      else if(PINA.5==0){
            navi=16;} //B
      else if(PINA.4==0){
            navi=15;} //A
            //
      PORTA.3=1;
      PORTA.2=0;
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
        delay_us(10);
        if(PINA.7==0){
              navi=14;} //#
        else if(PINA.6==0){
              keypressed=9;}
        else if(PINA.5==0){
              keypressed=6;}
        else if(PINA.4==0){
              keypressed=3;}
              //
        PORTA.2=1;
        PORTA.1=0;
        delay_us(10);
        if(PINA.7==0){
              keypressed=0;}  //0
        else if(PINA.6==0){
              keypressed=8;}
        else if(PINA.5==0){
              keypressed=5;}
        else if(PINA.4==0){
              keypressed=2;}
              //
        PORTA.1=1;
        PORTA.0=0;
        delay_us(10);
        if(PINA.7==0){
              navi=13;}  //*
        else if(PINA.6==0){
              keypressed=7;}
        else if(PINA.5==0){
              keypressed=4;}
        else if(PINA.4==0){
              keypressed=1;}
        delay_ms(10);
        if(navi==15){select++;}   //ok
        else if((navi==16)&&(select!=0)){select--;} //back or cancel
        else if((keypressed<=9)&&(select!=0)){numb++;};}

//ambil waktu
interrupt [TIM2_OVF] void timer2_ovf_isr(void)
{
// Place your code here
d++;
if(d==16){
    detik++;
    if(detik>=60){
        detik=0;
        menit++;}
    d=0;}
if(menit>=60){
    menit=0;
    jam++;}
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
157   if(jam>=24){jam=0;};}
158
159   void setdefault(){
160         unsigned char i;
161
162         //seting default lighting
163         for(i=0;i<6;i++){
164               li[i].risetm.hour=7;li[i].risetm.min=0;
165               li[i].drise.hour=2;li[i].drise.min=0;
166               li[i].falltm.hour=17;li[i].falltm.min=0;
167               li[i].dfall.hour=2;li[i].dfall.min=0;
168               li[i].peak=100;
169               li[i].hiti=((int)li[i].drise.hour*60)+li[i].drise.min;
170               li[i].hitd=((int)li[i].dfall.hour*60)+li[i].dfall.min;}
171
172         //seting default dose
173         dos[0].mode=1;dos[0].method=1;dos[0].voldose=10;
174         dos[1].mode=1;dos[1].method=2;dos[1].voldose=0;
175         dos[2].mode=3;dos[3].mode=3;
176
177         //seting default suhu
178         setsuhu=23.5;
179         //simpan seting default ke eeprom
180         eepsetsuhu=setsuhu;
181         for(i=0;i<6;i++){eepsetlight[i]=li[i];}
182         for(i=0;i<4;i++){eepsetdose[i]=dos[i];};}
183
184   void light_sim(){
185         unsigned char i;
186         for(i=0;i<3;i++){ //seting default lighting
187               li[i].risetm.hour=jam;li[i].risetm.min=menit+(i+1);
188               li[i].drise.hour=0;li[i].drise.min=3;
189               li[i].falltm.hour=jam+((menit+i+5)/60);
190               li[i].falltm.min=menit+(i+5);
191               li[i].dfall.hour=0;li[i].dfall.min=3;
192               li[i].peak=100;
193               li[i].hiti=((int)li[i].drise.hour*60)+li[i].drise.min;
194               li[i].hitd=((int)li[i].dfall.hour*60)+li[i].dfall.min;}
195         for(i=3;i<6;i++){ //seting default lighting
196               li[i].risetm.hour=jam;li[i].risetm.min=menit+(i-2);
197               li[i].drise.hour=0;li[i].drise.min=3;
198               li[i].falltm.hour=jam+((menit+i+2)/60);
199               li[i].falltm.min=menit+(i+2);
200               li[i].dfall.hour=0;li[i].dfall.min=3;
201               li[i].peak=100;
202               li[i].hiti=((int)li[i].drise.hour*60)+li[i].drise.min;
203               li[i].hitd=((int)li[i].dfall.hour*60)+li[i].dfall.min;}}
204
205   void menu(){
206         bit eint;
207         unsigned char a[4]={0,0,0,0};
208         float b[3]={0,0,0};
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
209         float c;
210         start:
211         lcd_clear();
212         if((select==1)){    //menu utama
213               if(navi==12){cursor[0]++;}  //next
214               else if((navi==11)&&(cursor[0]!=0)){cursor[0]--;} //prev
215               lcd_gotoxy(0,0);
216               lcd_putsf("      MENU\n Time    Dose\n Temp    Default\n
217                     Light   Sim");
218               switch(cursor[0]){
219                     case 0:
220                           lcd_gotoxy(0,1);
221                           lcd_putsf(">");
222                           break;
223                     case 1:
224                           lcd_gotoxy(0,2);
225                           lcd_putsf(">");
226                           break;
227                     case 2:
228                           lcd_gotoxy(0,3);
229                           lcd_putsf(">");
230                           break;
231                     case 3:
232                           lcd_gotoxy(8,1);
233                           lcd_putsf(">");
234                           break ;
235                     case 4:
236                           lcd_gotoxy(8,2);
237                           lcd_putsf(">");
238                           break ;
239                     case 5:
240                           lcd_gotoxy(8,3);
241                           lcd_putsf(">");
242                           break ;
243                     default:
244                           cursor[0]=0;
245                           break;}
246         }
247         else if((select==2)&&(cursor[0]==0)){ //seting waktu
248               if(navi==12){cursor[1]++;numb=0;}  //next
249               else if((navi==11)&&(cursor[1]!=0)){cursor[1]--;numb=0;}
250               lcd_gotoxy(0,0);
251               sprintf(lcd,"    Set Time\n Hour %02u Sec %02u\n Min
252                     %02u \n",jam,detik,menit);
253               lcd_puts(lcd);
254               switch(cursor[1]){     //cursor
255                     case 0:
256                           lcd_gotoxy(0,1);
257                           lcd_putsf(">");
258                           break;
259                     case 1:
260                           lcd_gotoxy(0,2);
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
                        lcd_putsf(">");
                        break;
                  case 2:
                        lcd_gotoxy(8,1);
                        lcd_putsf(">");
                        break;
                  default:
                        cursor[1]=0;
                        break;}
            if((cursor[1]==0)&&(numb!=0)){ //jam
                        a[numb-1]=keypressed;
                        lcd_gotoxy(6,1);
                        sprintf(lcd,"%u%u",a[0],a[1]);
                        lcd_puts(lcd);
                        if(numb==2){
                              jam=a[0]*10+a[1];
                              if(jam>=24){jam=00;}
                              rtc_set_time(jam,menit,detik);
                              numb=a[0]=a[1]=0;}}
            else if((cursor[1]==1)&&(numb!=0)){    //menit
                        a[numb-1]=keypressed;
                        lcd_gotoxy(6,2);
                        sprintf(lcd,"%u%u",a[0],a[1]);
                        lcd_puts(lcd);
                        if(numb==2){
                              menit=a[0]*10+a[1];
                              if(menit>=60){menit=00;}
                              rtc_set_time(jam,menit,detik);
                              numb=a[0]=a[1]=0;}}
            else if((cursor[1]==2)&&(numb!=0)){    //detik
                        a[numb-1]=keypressed;
                        lcd_gotoxy(13,1);
                        sprintf(lcd,"%u%u",a[0],a[1]);
                        lcd_puts(lcd);
                        if(numb==2){
                              detik=a[0]*10+a[1];
                              if(detik>=60){detik=00;}
                              rtc_set_time(jam,menit,detik);
                              numb=a[0]=a[1]=0;}}
            }
      else if((select==2)&&(cursor[0]==1)){ //seting temp
            if(navi==15){numb=0;}
            lcd_gotoxy(0,0);
            sprintf(lcd,"    Set Temp\n\n Temp=%.3f\xdfC",setsuhu);
            lcd_puts(lcd);
            if(numb!=0){
                  if(numb<=2){a[numb-1]=keypressed;}
                  else if((numb>2)&&(numb<=5)){
                        b[numb-3]=keypressed;}
                  lcd_gotoxy(6,2);
                  sprintf(lcd,"%u%u.%.0f%.0f%.0f",
                        a[0],a[1],b[0],b[1],b[2]);
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
313                     lcd_puts(lcd);
314                     if(numb==5){
315                           c=(float)(((int)((b[0]*100)+
316                                 (b[1]*10)+(b[2]))/125)*0.125);
317                           setsuhu=((a[0]*10)+a[1]+c);
318                           eepsetsuhu=setsuhu;
319                           numb=a[0]=a[1]=b[0]=b[1]=0;}}}
320         else if((select==2)&&(cursor[0]==2)){ //seting light
321               if(navi==12){cursor[1]++;numb=0;}  //next
322               else if((navi==11)&&(cursor[1]!=0)){cursor[1]--;numb=0;}
323               lcd_gotoxy(0,0);
324               lcd_putsf("    Set Light\n Ch1     Ch4\n Ch2     Ch5\n
325                     Ch3     Ch6");
326               switch(cursor[1]){     //cursor
327                     case 0:
328                           lcd_gotoxy(0,1);
329                           lcd_putsf(">");
330                           break;
331                     case 1:
332                           lcd_gotoxy(0,2);
333                           lcd_putsf(">");
334                           break;
335                     case 2:
336                           lcd_gotoxy(0,3);
337                           lcd_putsf(">");
338                           break;
339                     case 3:
340                           lcd_gotoxy(8,1);
341                           lcd_putsf(">");
342                           break;
343                     case 4:
344                           lcd_gotoxy(8,2);
345                           lcd_putsf(">");
346                           break;
347                     case 5:
348                           lcd_gotoxy(8,3);
349                           lcd_putsf(">");
350                           break;
351                     case 6:
352                           lcd_gotoxy(0,1);
353                           lcd_putsf(">");
354                           break;
355                     default:
356                           cursor[1]=0;
357                           break;}
358               }
359         else if((select==2)&&(cursor[0]==3)){ //seting dose
360               if(navi==12){cursor[1]++;numb=0;}  //next
361               else if((navi==11)&&(cursor[1]!=0)){cursor[1]--;numb=0;}
362               lcd_gotoxy(0,0);
363               lcd_putsf("    Set Dose\n Ch1     Ch3\n Ch2     Ch4\n");
364               switch(cursor[1]){     //cursor
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
                    case 0:
                          lcd_gotoxy(0,1);
                          lcd_putsf(">");
                          break;
                    case 1:
                          lcd_gotoxy(0,2);
                          lcd_putsf(">");
                          break;
                    case 2:
                          lcd_gotoxy(8,1);
                          lcd_putsf(">");
                          break;
                    case 3:
                          lcd_gotoxy(8,2);
                          lcd_putsf(">");
                          break;
                    default:
                          cursor[1]=0;
                          break;}
               }
          else if((select==2)&&(cursor[0]==4)){ //seting default
               setdefault();
               select=0;}
          else if((select==2)&&(cursor[0]==5)){ //simulasi lighting
               light_sim();
               select=0;}
          else if((select==3)&&(cursor[0]==2)){ //setlight b
               if(navi==15){numb=0;}
               lcd_gotoxy(0,0);
               sprintf(lcd,"    Set Peak\n>Peak: %03u\x25\n\n",
                    li[cursor[1]].peak);
               lcd_puts(lcd);
               if(numb!=0){
                    a[numb-1]=keypressed;
                    lcd_gotoxy(7,1);
                    sprintf(lcd,"%u%u%u",a[0],a[1],a[2]);
                    lcd_puts(lcd);
                    if(numb==3){
                         li[cursor[1]].peak=(a[0]*100)+
                              (a[1]*10)+a[2];
                         if(li[cursor[1]].peak>100){
                              li[cursor[1]].peak=100;}
                         eepsetlight[cursor[1]].peak=
                              li[cursor[1]].peak;
                         numb=a[0]=a[1]=a[2]=0;};};}
          else if((select==3)&&(cursor[0]==3)){ //setdose b
               if(navi==12){cursor[2]++;numb=0;}  //next
               else if((navi==11)&&(cursor[2]!=0)){cursor[2]--;numb=0;}
               lcd_gotoxy(0,0);
               lcd_putsf("   Dose Mode\n Float Mode \n Timer Mode\n
                    Disconect");
               switch(cursor[2]){
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
                    case 0:
                          lcd_gotoxy(0,1);
                          lcd_putsf(">");
                          break;
                    case 1:
                          lcd_gotoxy(0,2);
                          lcd_putsf(">");
                          break;
                    case 2:
                          lcd_gotoxy(0,3);
                          lcd_putsf(">");
                          break;
                    default:
                          cursor[2]=0;
                          break;}
              }
        else if((select==4)&&(cursor[0]==2)){ //setlight risetime
              if(navi==12){cursor[2]++;numb=0;}  //next
              else if((navi==11)&&(cursor[2]!=0)){cursor[2]--;numb=0;}
              lcd_gotoxy(0,0);
              sprintf(lcd,"  Set RiseTime\n Time   %02u:%02u\n Delay
                    %02u:%02u\n",li[cursor[1]].risetm.hour,
                    li[cursor[1]].risetm.min,li[cursor[1]].drise.hour,
                    li[cursor[1]].drise.min);
              lcd_puts(lcd);
              switch(cursor[2]){
                    case 0:
                          lcd_gotoxy(0,1);
                          lcd_putsf(">");
                          break;
                    case 1:
                          lcd_gotoxy(0,2);
                          lcd_putsf(">");
                          break;
                    default:
                          cursor[2]=0;
                          break;}
              if((numb!=0)&&(cursor[2]==0)){
                    a[numb-1]=keypressed;
                    lcd_gotoxy(8,1);
                    sprintf(lcd,"%u%u:%u%u",a[0],a[1],a[2],a[3]);
                    lcd_puts(lcd);
                    if(numb==4){
                          li[cursor[1]].risetm.hour=(a[0]*10)+a[1];
                          if(li[cursor[1]].risetm.hour>=24){
                                li[cursor[1]].risetm.hour=0;}
                          li[cursor[1]].risetm.min=(a[2]*10)+a[3];
                          if(li[cursor[1]].risetm.min>=60){
                                li[cursor[1]].risetm.min=0;}
                          eepsetlight[cursor[1]].risetm.hour=
                                li[cursor[1]].risetm.hour;
                          eepsetlight[cursor[1]].risetm.min=
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
                              li[cursor[1]].risetm.min;
                        numb=a[0]=a[1]=a[2]=a[3]=0;}}
            else if((numb!=0)&&(cursor[2]==1)){
                  a[numb-1]=keypressed;
                  lcd_gotoxy(8,2);
                  sprintf(lcd,"%u%u:%u%u",a[0],a[1],a[2],a[3]);
                  lcd_puts(lcd);
                  if(numb==4){
                        li[cursor[1]].drise.hour=(a[0]*10)+a[1];
                        if(li[cursor[1]].drise.hour>=24){
                              li[cursor[1]].drise.hour=0;}
                        li[cursor[1]].drise.min=(a[2]*10)+a[3];
                        if(li[cursor[1]].drise.min>=60){
                              li[cursor[1]].drise.min=0;}
                        eepsetlight[cursor[1]].drise.hour=
                              li[cursor[1]].drise.hour;
                        eepsetlight[cursor[1]].drise.min=
                              li[cursor[1]].drise.min;
                        li[cursor[1]].hiti=
                              ((int)li[cursor[1]].drise.hour*60)+
                              li[cursor[1]].drise.min;
                        eepsetlight[cursor[1]].hiti=
                              li[cursor[1]].hiti;
                        numb=a[0]=a[1]=a[2]=a[3]=0;};};}
      //float mode
      else if((select==4)&&(cursor[0]==3)&&(cursor[2]==0)){
            if(navi==12){cursor[3]++;numb=0;}  //next
            else if((navi==11)&&(cursor[3]!=0)){cursor[3]--;numb=0;}
            dos[cursor[1]].mode=1;eepsetdose[cursor[1]].mode=1;
            lcd_gotoxy(0,0);
            sprintf(lcd,"   Float Mode\n Method  %01u *)\n Vol
                  %02u0ml\n *)1:Nrm  2:Drn",dos[cursor[1]].method,
                  dos[cursor[1]].voldose);
            lcd_puts(lcd);
            switch(cursor[3]){
                  case 0:
                        lcd_gotoxy(0,1);
                        lcd_putsf(">");
                        break;
                  case 1:
                        lcd_gotoxy(0,2);
                        lcd_putsf(">");
                        break;
                  default:
                        cursor[3]=0;
                        break;}
            if((numb!=0)&&(cursor[3]==0)){
                  dos[cursor[1]].method=keypressed;
                  if((keypressed>2)||(keypressed==0)){
                        dos[cursor[1]].method=1;}
                  lcd_gotoxy(9,1);
                  sprintf(lcd,"%u",dos[cursor[1]].method);
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
                  lcd_puts(lcd);
                  eepsetdose[cursor[1]].method=
                        dos[cursor[1]].method;}
            else if((numb!=0)&&(cursor[3]==1)){
                  a[numb-1]=keypressed;
                  lcd_gotoxy(9,2);
                  sprintf(lcd,"%u%u0",a[0],a[1]);
                  lcd_puts(lcd);
                  if(numb==2){
                        dos[cursor[1]].voldose=(a[0]*10)+(a[1]);
                        if(dos[cursor[1]].voldose>25){
                              dos[cursor[1]].voldose=25;}
                        eepsetdose[cursor[1]].voldose=
                              dos[cursor[1]].voldose;
                        numb=a[0]=a[1]=a[2]=0;};};}
      //timer mode
      else if((select==4)&&(cursor[0]==3)&&(cursor[2]==1)){
            dos[cursor[1]].mode=2;eepsetdose[cursor[1]].mode=2;
            if(navi==12){cursor[3]++;numb=0;}  //next
            else if((navi==11)&&(cursor[3]!=0)){cursor[3]--;numb=0;}
            lcd_gotoxy(0,0);
            sprintf(lcd," Set Dose Timer\n 1 %02u:%02u 4 %02u:%02u",
                  dos[cursor[1]].wdose[0].hour,
                  dos[cursor[1]].wdose[0].min,
                  dos[cursor[1]].wdose[3].hour,
                  dos[cursor[1]].wdose[3].min);
            lcd_puts(lcd);
            lcd_gotoxy(0,2);
            sprintf(lcd," 2 %02u:%02u 5 %02u:%02u",
                  dos[cursor[1]].wdose[1].hour,
                  dos[cursor[1]].wdose[1].min,
                  dos[cursor[1]].wdose[4].hour,
                  dos[cursor[1]].wdose[4].min);
            lcd_puts(lcd);
            lcd_gotoxy(0,3);
            sprintf(lcd," 3 %02u:%02u 6 %02u:%02u",
                  dos[cursor[1]].wdose[2].hour,
                  dos[cursor[1]].wdose[2].min,
                  dos[cursor[1]].wdose[5].hour,
                  dos[cursor[1]].wdose[5].min);
            lcd_puts(lcd);
            switch(cursor[3]){
                  case 0:
                        lcd_gotoxy(0,1);
                        lcd_putsf(">");
                        break;
                  case 1:
                        lcd_gotoxy(0,2);
                        lcd_putsf(">");
                        break;
                  case 2:
                        lcd_gotoxy(0,3);
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
573                         lcd_putsf(">");
574                         break;
575                   case 3:
576                         lcd_gotoxy(8,1);
577                         lcd_putsf(">");
578                         break;
579                   case 4:
580                         lcd_gotoxy(8,2);
581                         lcd_putsf(">");
582                         break;
583                   case 5:
584                         lcd_gotoxy(8,3);
585                         lcd_putsf(">");
586                         break;
587                   default:
588                         cursor[3]=0;
589                         break;}
590             if((numb!=0)&&(cursor[3]==0)){
591                   a[numb-1]=keypressed;
592                   lcd_gotoxy(3,1);
593                   sprintf(lcd,"%u%u:%u%u",a[0],a[1],a[2],a[3]);
594                   lcd_puts(lcd);
595                   if(numb==4){
596                         dos[cursor[1]].wdose[0].hour=(a[0]*10)+a[1];
597                         if(dos[cursor[1]].wdose[0].hour>=24){
598                               dos[cursor[1]].wdose[0].hour=0;}
599                         dos[cursor[1]].wdose[0].min=(a[2]*10)+a[3];
600                         if(dos[cursor[1]].wdose[0].min>=60){
601                               dos[cursor[1]].wdose[0].min=0;}
602                         eepsetdose[cursor[1]].wdose[0].hour=
603                               dos[cursor[1]].wdose[0].hour;
604                         eepsetdose[cursor[1]].wdose[0].min=
605                               dos[cursor[1]].wdose[0].min;
606                         numb=a[0]=a[1]=a[2]=a[3]=0;}}
607             else if((numb!=0)&&(cursor[3]==1)){
608                   a[numb-1]=keypressed;
609                   lcd_gotoxy(3,2);
610                   sprintf(lcd,"%u%u:%u%u",a[0],a[1],a[2],a[3]);
611                   lcd_puts(lcd);
612                   if(numb==4){
613                         dos[cursor[1]].wdose[1].hour=(a[0]*10)+a[1];
614                         if(dos[cursor[1]].wdose[1].hour>=24){
615                               dos[cursor[1]].wdose[1].hour=0;}
616                         dos[cursor[1]].wdose[1].min=(a[2]*10)+a[3];
617                         if(dos[cursor[1]].wdose[1].min>=60){
618                               dos[cursor[1]].wdose[1].min=0;}
619                         eepsetdose[cursor[1]].wdose[1].hour=
620                               dos[cursor[1]].wdose[1].hour;
621                         eepsetdose[cursor[1]].wdose[1].min=
622                               dos[cursor[1]].wdose[1].min;
623                         numb=a[0]=a[1]=a[2]=a[3]=0;}}
624             else if((numb!=0)&&(cursor[3]==2)){
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
                    a[numb-1]=keypressed;
                    lcd_gotoxy(3,3);
                    sprintf(lcd,"%u%u:%u%u",a[0],a[1],a[2],a[3]);
                    lcd_puts(lcd);
                    if(numb==4){
                          dos[cursor[1]].wdose[2].hour=(a[0]*10)+a[1];
                          if(dos[cursor[1]].wdose[2].hour>=24){
                                dos[cursor[1]].wdose[2].hour=0;}
                          dos[cursor[1]].wdose[2].min=(a[2]*10)+a[3];
                          if(dos[cursor[1]].wdose[2].min>=60){
                                dos[cursor[1]].wdose[2].min=0;}
                          eepsetdose[cursor[1]].wdose[2].hour=
                                dos[cursor[1]].wdose[2].hour;
                          eepsetdose[cursor[1]].wdose[2].min=
                                dos[cursor[1]].wdose[2].min;
                          numb=a[0]=a[1]=a[2]=a[3]=0;}}
              else if((numb!=0)&&(cursor[3]==3)){
                    a[numb-1]=keypressed;
                    lcd_gotoxy(11,1);
                    sprintf(lcd,"%u%u:%u%u",a[0],a[1],a[2],a[3]);
                    lcd_puts(lcd);
                    if(numb==4){
                          dos[cursor[1]].wdose[3].hour=(a[0]*10)+a[1];
                          if(dos[cursor[1]].wdose[3].hour>=24){
                                dos[cursor[1]].wdose[3].hour=0;}
                          dos[cursor[1]].wdose[3].min=(a[2]*10)+a[3];
                          if(dos[cursor[1]].wdose[3].min>=60){
                                dos[cursor[1]].wdose[3].min=0;}
                          eepsetdose[cursor[1]].wdose[3].hour=
                                dos[cursor[1]].wdose[3].hour;
                          eepsetdose[cursor[1]].wdose[3].min=
                                dos[cursor[1]].wdose[3].min;
                          numb=a[0]=a[1]=a[2]=a[3]=0;}}
              else if((numb!=0)&&(cursor[3]==4)){
                    a[numb-1]=keypressed;
                    lcd_gotoxy(11,2);
                    sprintf(lcd,"%u%u:%u%u",a[0],a[1],a[2],a[3]);
                    lcd_puts(lcd);
                    if(numb==4){
                          dos[cursor[1]].wdose[4].hour=(a[0]*10)+a[1];
                          if(dos[cursor[1]].wdose[4].hour>=24){
                                dos[cursor[1]].wdose[4].hour=0;}
                          dos[cursor[1]].wdose[4].min=(a[2]*10)+a[3];
                          if(dos[cursor[1]].wdose[4].min>=60){
                                dos[cursor[1]].wdose[4].min=0;}
                          eepsetdose[cursor[1]].wdose[4].hour=
                                dos[cursor[1]].wdose[4].hour;
                          eepsetdose[cursor[1]].wdose[4].min=
                                dos[cursor[1]].wdose[4].min;
                          numb=a[0]=a[1]=a[2]=a[3]=0;}}
              else if((numb!=0)&&(cursor[3]==5)){
                    a[numb-1]=keypressed;
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
                    lcd_gotoxy(11,3);
                    sprintf(lcd,"%u%u:%u%u",a[0],a[1],a[2],a[3]);
                    lcd_puts(lcd);
                    if(numb==4){
                          dos[cursor[1]].wdose[5].hour=(a[0]*10)+a[1];
                          if(dos[cursor[1]].wdose[5].hour>=24){
                                dos[cursor[1]].wdose[5].hour=0;}
                          dos[cursor[1]].wdose[5].min=(a[2]*10)+a[3];
                          if(dos[cursor[1]].wdose[5].min>=60){
                                dos[cursor[1]].wdose[5].min=0;}
                          eepsetdose[cursor[1]].wdose[5].hour=
                                dos[cursor[1]].wdose[5].hour;
                          eepsetdose[cursor[1]].wdose[5].min=
                                dos[cursor[1]].wdose[5].min;
                          numb=a[0]=a[1]=a[2]=a[3]=0;};};}
      else if((select==4)&&(cursor[0]==3)&&(cursor[2]==2)){
            dos[cursor[1]].mode=3;
            eepsetdose[cursor[1]].mode=3;select=0;}
      else if((select==5)&&(cursor[0]==2)){ //setlight falltime
            if(navi==12){cursor[3]++;numb=0;}  //next
            else if((navi==11)&&(cursor[3]!=0)){cursor[3]--;numb=0;}
            lcd_gotoxy(0,0);
            sprintf(lcd,"  Set FallTime\n Time   %02u:%02u\n Delay
                  %02u:%02u\n",li[cursor[1]].falltm.hour,
                  li[cursor[1]].falltm.min,li[cursor[1]].dfall.hour,
                  li[cursor[1]].dfall.min);
            lcd_puts(lcd);
            switch(cursor[3]){
                  case 0:
                        lcd_gotoxy(0,1);
                        lcd_putsf(">");
                        break;
                  case 1:
                        lcd_gotoxy(0,2);
                        lcd_putsf(">");
                        break;
                  default:
                        cursor[3]=0;
                        break;}
            if((numb!=0)&&(cursor[3]==0)){
                  a[numb-1]=keypressed;
                  lcd_gotoxy(8,1);
                  sprintf(lcd,"%u%u:%u%u",a[0],a[1],a[2],a[3]);
                  lcd_puts(lcd);
                  if(numb==4){
                        li[cursor[1]].falltm.hour=(a[0]*10)+a[1];
                        if(li[cursor[1]].falltm.hour>=24){
                              li[cursor[1]].falltm.hour=0;}
                        li[cursor[1]].falltm.min=(a[2]*10)+a[3];
                        if(li[cursor[1]].falltm.min>=60){
                              li[cursor[1]].falltm.min=0;}
                        eepsetlight[cursor[1]].falltm.hour=
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
                              li[cursor[1]].falltm.hour;
                        eepsetlight[cursor[1]].falltm.min=
                              li[cursor[1]].falltm.min;
                        numb=a[0]=a[1]=a[2]=a[3]=0;}}
            else if((numb!=0)&&(cursor[3]==1)){
                  a[numb-1]=keypressed;
                  lcd_gotoxy(8,2);
                  sprintf(lcd,"%u%u:%u%u",a[0],a[1],a[2],a[3]);
                  lcd_puts(lcd);
                  if(numb==4){
                        li[cursor[1]].dfall.hour=(a[0]*10)+a[1];
                        if(li[cursor[1]].dfall.hour>=24){
                              li[cursor[1]].dfall.hour=0;}
                        li[cursor[1]].dfall.min=(a[2]*10)+a[3];
                        if(li[cursor[1]].dfall.min>=60){
                              li[cursor[1]].dfall.min=0;}
                        eepsetlight[cursor[1]].dfall.hour=
                              li[cursor[1]].dfall.hour;
                        eepsetlight[cursor[1]].dfall.min=
                              li[cursor[1]].dfall.min;
                        li[cursor[1]].hitd=
                              ((int)li[cursor[1]].dfall.hour*60)+
                              li[cursor[1]].dfall.min;
                        eepsetlight[cursor[1]].hitd=
                              li[cursor[1]].hitd;
                        numb=a[0]=a[1]=a[2]=a[3]=0;};};}
      //timer mode2
      else if((select==5)&&(cursor[0]==3)&&(cursor[2]==1)){
            if(navi==12){cursor[4]++;numb=0;}  //next
            else if((navi==11)&&(cursor[4]!=0)){cursor[4]--;numb=0;}
            lcd_gotoxy(0,0);
            sprintf(lcd," Set Dose Volum\n\n Volume: %02u0ml",
                  dos[cursor[1]].voldose);
            lcd_puts(lcd);
            if(numb!=0){
                  a[numb-1]=keypressed;
                  lcd_gotoxy(9,2);
                  sprintf(lcd,"%u%u0",a[0],a[1]);
                  lcd_puts(lcd);
                  if(numb==2){
                        dos[cursor[1]].voldose=(a[0]*10)+(a[1]);
                        if(dos[cursor[1]].voldose>25){
                              dos[cursor[1]].voldose=25;}
                        eepsetdose[cursor[1]].voldose=
                              dos[cursor[1]].voldose;
                        numb=a[0]=a[1]=a[2]=0;};};}
      else if(select==0){
            goto exit;}
      else {select=0;}
      navi=0;
      PORTA=0xf0;
      eint=(PINA.7&PINA.6&PINA.5&PINA.4);
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
781         PORTE.0=eint;
782         delay_ms(100);
783         goto start;
784         exit:
785   }
786
787   void lighting(){
788         unsigned char j;
789         for(j=0;j<6;j++){
790         //inisialisasi
791         inclight[j]=((float)(100*li[j].peak)/li[j].hiti)/60;
792         sunrise[j]=((int)li[j].risetm.hour*60)+li[j].risetm.min;
793         declight[j]=((float)(100*li[j].peak)/li[j].hitd)/60;
794         sunset[j]=((int)li[j].falltm.hour*60)+li[j].falltm.min;
795
796         //naik:sunrise
797         if((time>=sunrise[j])&&(time<=(sunrise[j]+li[j].hiti))){
798         lightch[j]=(((time-sunrise[j])*60)+detik)*(inclight[j]/40);
799         if(lightch[j]>(li[j].peak*2.5)){
800               lightch[j]=2.5*li[j].peak;};}
801         else if((time>(sunrise[j]+li[j].hiti))&&(time<sunset[j])){
802               lightch[j]=2.5*li[j].peak;}
803
804         //turun:sunset
805         else if((time>=sunset[j])&&(time<=(sunset[j]+li[j].hitd))){
806               lightch[j]=(2.5*li[j].peak)-((((time-
807                     sunset[j])*60)+detik)*declight[j]/40);
808               if(lightch[j]<0){lightch[j]=0;}}
809         else if(time>(sunset[j]+li[j].hitd)){
810               lightch[j]=0;}
811         else {lightch[j]=0;};}
812         OCR1A=(unsigned int)lightch[0];
813         OCR1B=(unsigned int)lightch[1];
814         OCR1CL=(unsigned int)lightch[2];
815         OCR3AL=(unsigned int)lightch[3];
816         OCR0=(unsigned int)lightch[4];
817         OCR3CL=(unsigned int)lightch[5];
818         last_detik=detik;}
819
820   void dosing(){
821         unsigned char j,k,l[4]={1,1,1,1},last_min;
822         l[0]=PIND.4;
823         l[1]=PIND.5;
824         l[2]=PIND.6;
825         l[3]=PINE.1;
826         delay_ms(10);
827         for(j=0;j<4;j++){
828         if(dos[j].mode==1){
829               if(dos[j].method==1){
830                     if(l[j]==0){
831                     if(((tes[0]|tes[1]|tes[2]|tes[3])!=0)&&
832                           (tes[j]!=1)) buff[j]=1;
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
                    else {
                          PORTF=PORTF|((unsigned char)pow(2,j));
                          st[j]=1;tes[j]=1;};};}
              else if(dos[j].method==2){
                    switch(l[j]){
                    case 0:
                          PORTF=PORTF&(0xff^(unsigned char)pow(2,j));
                          st[j]=0;tes[j]=0;
                          break;
                    case 1:
                          PORTF=PORTF|((unsigned char)pow(2,j));
                          st[j]=1;tes[j]=0;
                          break;
                    default:
                          PORTF=PORTF|((unsigned char)pow(2,j));
                          st[j]=1;tes[j]=0;
                          break;};};}
        else if(dos[j].mode==2){
        for(k=0;k<6;k++){
              if((jam==dos[j].wdose[k].hour)&&(menit==
                    dos[j].wdose[k].min)&&(detik==0)){
                    if(((tes[0]|tes[1]|tes[2]|tes[3])!=0)&&
                          (tes[j]!=1)) buff[j]=1;
                    else {
                          PORTF=PORTF|((unsigned char)pow(2,j));
                          st[j]=1;tes[j]=1;}
                          last_min=menit;};};}
        else if(dos[j].mode>=3){
              PORTF=PORTF&(0xff^(unsigned char)pow(2,j));
              st[j]=0;tes[j]=0;}
        if(((tes[0]|tes[1]|tes[2]|tes[3])==0)&&(buff[j]==1)){
              PORTF=PORTF|((unsigned char)pow(2,j));
              tes[j]=1;st[j]=1;buff[j]=0;};}
        ld=detik;}

void temp(){
        suhu=ds18b20_temperature(0);
        if((int)(suhu*1000)>=(int)(setsuhu*1000)+125){
              PORTF=PORTF|0x10;st[4]=1;}
        else if((int)(suhu*1000)<=(int)(setsuhu*1000)-125){
              PORTF=PORTF&(0xff^0x10);st[4]=0;}
        if((int)(suhu*1000)>=(int)(setsuhu*1000)+1000){
              PORTF=PORTF|0x10;PORTF=PORTF|0x20;st[4]=st[5]=1;}
        else if((int)(suhu*1000)<=(int)(setsuhu*1000)-1000){
              PORTF=PORTF&(0xff^0x10);
              PORTF=PORTF|0x20;st[4]=0;st[5]=1;}
        else {
              PORTF=PORTF&(0xff^0x20);st[5]=0;};}

void cek_motor(){
        unsigned char i;
        if(tes[0]==1){
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
885                 if(PIND.0==1)goto exit;
886                 while(PIND.0==0){
887                       if(n_rot[0]==dos[0].voldose+1) goto exit;}
888                 n_rot[0]++;
889                 if(n_rot[0]==dos[0].voldose+1){
890                       PORTF=PORTF&(0xff^(0x01));
891                       n_rot[0]=0;st[0]=0;tes[0]=0;};}
892           if(tes[1]==1){
893                 if(PIND.1==1)goto exit;
894                 while(PIND.1==0){
895                       if(n_rot[1]==dos[1].voldose+1) goto exit;}
896                 n_rot[1]++;
897                 if(n_rot[1]==dos[1].voldose+1){
898                       PORTF=PORTF&(0xff^(0x02));
899                       n_rot[1]=0;st[1]=0;tes[1]=0;}}
900           if(tes[2]==1){
901                 if(PIND.2==1)goto exit;
902                 while(PIND.2==0){
903                       if(n_rot[2]==dos[2].voldose+1) goto exit;}
904                 n_rot[2]++;
905                 if(n_rot[2]==dos[2].voldose+1){
906                       PORTF=PORTF&(0xff^(0x04));
907                       n_rot[2]=0;st[2]=0;tes[2]=0;}}
908           if(tes[3]==1){
909                 if(PIND.3==1)goto exit;
910                 while(PIND.3==0){
911                       if(n_rot[3]==dos[3].voldose+1) goto exit;}
912                 n_rot[3]++;
913                 if(n_rot[3]==dos[3].voldose+1){
914                       PORTF=PORTF&(0xff^(0x08));
915                       n_rot[3]=0;st[3]=0;tes[3]=0;}}
916     exit:
917     }
918
919     // Declare your global variables here
920
921     void main(void)
922     {
923     bit eint;
924     char i,f;
925     setsuhu=eepsetsuhu;
926     for(i=0;i<6;i++){li[i]=eepsetlight[i];}
927     for(i=0;i<4;i++){dos[i]=eepsetdose[i];}
928     PORTA=0x00;DDRA=0x0f;
929     PORTB=0xff;DDRB=0xff;
930     PORTC=0xff;DDRC=0xff;DDRC.3=0;PORTC.3=0;
931     PORTD=0x0f;DDRD=0x00;
932     PORTE=0x00;DDRE=0b00101101;
933     DDRF=0xFF;
934     PORTG=0x00;DDRG=0x00;
935     // External Interrupt(s) initialization
936     // INT0: Off
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
937   // INT1: Off
938   // INT2: Off
939   // INT3: Off
940   // INT4: Off
941   // INT5: Off
942   // INT6: Off
943   // INT7: On
944   // INT7 Mode: Falling Edge
945   EICRA=0xAA;EICRB=0x83;EIMSK=0x80;EIFR=0x80;
946
947   // Timer/Counter 0 initialization
948   // Clock source: System Clock
949   // Clock value: 125.000 kHz
950   // Mode: Phase correct PWM top=FFh
951   // OC0 output: Non-Inverted PWM
952   ASSR=0x00;TCCR0=0x64;TCNT0=0x00;OCR0=0x00;
953
954   // Timer/Counter 1 initialization
955   // Clock source: System Clock
956   // Clock value: 125.000 kHz
957   // Mode: Ph. correct PWM top=00FFh
958   // OC1A output: Non-Inv.
959   // OC1B output: Non-Inv.
960   // OC1C output: Non-Inv.
961   // Noise Canceler: Off
962   // Input Capture on Falling Edge
963   // Timer 1 Overflow Interrupt: Off
964   // Input Capture Interrupt: Off
965   // Compare A Match Interrupt: Off
966   // Compare B Match Interrupt: Off
967   // Compare C Match Interrupt: Off
968   TCCR1A=0xA9;TCCR1B=0x03;TCNT1H=0x00;TCNT1L=0x00;ICR1H=0x00;
969   ICR1L=0x00;OCR1AH=0x00;OCR1AL=0x00;OCR1BH=0x00;OCR1BL=0x00;
970   OCR1CH=0x00;OCR1CL=0x00;
971
972   // Timer/Counter 2 initialization
973   // Clock source: Pin T2 Falling Edge
974   // Clock value: 4096 Hz
975   // Mode: Normal top=FFh
976   // OC2 output: Disconnected
977   // Timer 2 Overflow Interrupt: On
978   TCCR2=0x06;TCNT2=0x00;OCR2=0x00;TIMSK=0x40;
979
980   // Timer/Counter 3 initialization
981   // Clock source: System Clock
982   // Clock value: 125.000 kHz
983   // Mode: Ph. correct PWM top=00FFh
984   // OC3A output: Non-Inv.
985   // OC3B output: Discon.
986   // OC3C output: Non-Inv.
987   // Noise Canceler: Off
988   // Input Capture on Falling Edge
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
// Timer 3 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR3A=0x89;TCCR3B=0x03;TCNT3H=0x00;TCNT3L=0x00;
ICR3H=0x00;ICR3L=0x00;OCR3AH=0x00;OCR3AL=0x00;
OCR3BH=0x00;OCR3BL=0x00;OCR3CH=0x00;OCR3CL=0x00;

//I2C Bus initialization
i2c_init();

// DS1307 Real Time Clock initialization
// Square wave output on pin SQW/OUT: On
// Square wave output frequency: 4096
// SQW/OUT pin state: 0
rtc_init(1,1,0);

// 1 Wire Bus initialization
w1_init();
w1_search(0xf0,rom_codes);
ds18b20_init(0,05,35,DS18B20_11BIT_RES);

// LCD module initialization
lcd_init(16);
_lcd_ready();

// Global enable interrupts
#asm("sei")
rtc_get_time(&jam,&menit,&detik);
time=((int)jam*60)+menit;
temp();
lighting();
while (1)
      {
      // Place your code here
      //scan keypad lewat interrupt
      keypressed=0;navi=0;select=0;numb=0;
      for(i=0;i<10;i++){cursor[i]=0;}
      PORTA=0xf0;
      eint=(PINA.7&PINA.6&PINA.5&PINA.4);
      PORTE.0=eint;
      //ambil data waktu dari DS1307 setiap pukul 00:00:00
      if((jam==0)&&(menit==0)&&(detik==0))
            rtc_get_time(&jam,&menit,&detik);
      time=((int)jam*60)+menit;     //ubah waktu ke bentuk menit
      f=fmod(detik,2);              //cari detik kelipatan 2
      if((f==0)&&(detik!=ld)){      //detik kelipatan 2
            dosing();}
      if((detik==3)||(detik==33)){  //ambil data suhu
            temp();}
      //cek motor saat output aktif
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
1041        if((st[0]|st[1]|st[2]|st[3])==1) cek_motor();
1042        //detik bertambah
1043        if(detik!=last_detik){
1044        lighting();
1045        lcd_clear();
1046        lcd_gotoxy(0,0);
1047        sprintf(lcd,"MAC v1  %02u:%02u:%02u\n Temp=%.3f\xdfC",
1048              jam,menit,detik,suhu);
1049        lcd_puts(lcd);
1050        //sensor suhu tidak terdeteksi
1051        if(suhu==-9999.0){
1052              lcd_gotoxy(0,1);lcd_putsf(" Temp=Unpluged  ");}
1053        lcd_gotoxy(0,2);
1054        //tampilkan status pencahayaan saat detik = genap
1055        if(f==0){
1056              sprintf(lcd,"Lght %03.0f %03.0f %03.0f\nLght %03.0f
1057                    %03.0f %03.0f",(lightch[0]/2.5),(lightch[1]/2.5),
1058                    (lightch[2]/2.5),(lightch[3]/2.5),
1059                    (lightch[4]/2.5),(lightch[5]/2.5));}
1060        //tampilkan status output pompa (4kanal),
1061        //status chiller,status buzzer,
1062        //status putaran motor pompa dosing saat detik = ganjil
1063        else {
1064              sprintf(lcd,"  %02u %02u %02u %02u   \n  %u %u %u %u %u
1065              %u ",n_rot[0],n_rot[1],n_rot[2],n_rot[3],st[0],st[1],
1066              st[2],st[3],st[4],st[5]);}
1067        lcd_puts(lcd);}
1068        if(select!=0){menu();}
1069        delay_ms(50);};}
```