



**PEMANTAU ISI KULKAS MENGGUNAKAN ETHERNET SHIELD R3
BERBASIS ARDUNO UNO R3**

PROYEK AKHIR

**Diajukan Kepada Fakultas Teknik Universitas Negeri Yogyakarta untuk
Memenuhi Sebagian Persyaratan Guna Memperoleh Gelar Ahli Madya
Program Studi Teknik Elektronika**



OLEH:

AHMAD HARIS

NIM. 13507134018

**PROGRAM STUDI TEKNIK ELEKTRONIKA
JURUSAN PENDIDIKAN TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2016**

**LEMBAR PERSETUJUAN
PROYEK AKHIR**

**PEMANTAU ISI KULKAS MENGGUNAKAN ETHERNET SHIELD R3
BERBASIS ARDUINO UNO R3**

Oleh

AHMAD HARIS

13507134018


Telah diperiksa dan disetujui oleh pembimbing:
Untuk diujikan

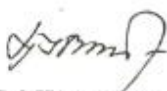


Yogyakarta, 25 November 2016

Mengetahui
Kaprodik Teknik Elektronika

Menyetujui
Pembimbing


Dra. Sri Waluyanti, M.Pd.
NIP. 19581218 198603 2 001


Dra. Sri Waluyanti, M.Pd.
NIP. 19581218 198603 2 001

LEMBAR PENGESAHAN

PROYEK AKHIR

**PEMANTAU ISI KULKAS MENGGUNAKAN ETHERNET SHIELD R3
BERBASIS ARDUINO UNO R3**

Dipersiapkan dan Disusun Oleh:

AHMAD HARIS

13507134018

Telah Dipertahankan didepan Dewan Penguji Proyek Akhir
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA
Pada tanggal 25 November 2016

Dan Dinyatakan Telah Memenuhi Syarat Guna Memperoleh Gelar
Ahli Madya Teknik

SUSUNAN DEWAN PENGUJI

Jabatan	Nama Lengkap dan Gelar	Tandatangan	Tanggal
1. Ketua Penguji	Dr. Sri Wakuyanti, M.Pd.		16 - Des - 2016
2. Sekretaris	Pipit Utami, M.Pd.		16 - Des 2016
3. Penguji Utama	Muhammad Munir, M.Pd.		16 - Des - 2016

Yogyakarta, 25 November 2016

Mengetahui,

Dekan Fakultas Teknik UNY



Dr. Widarto, M.Pd.

NIP. 19631230 198812 1 001 f.

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Ahmad Haris
NIM : 13507134018
Program Studi : Teknik Elektronika DIII
Judul Proyek Akhir : Pemantau Isi Kulkas Menggunakan Ethernet Shield
R3 Berbasis Arduino Uno R3

Menyatakan bahwa Proyek Akhir ini adalah hasil pekerjaan saya sendiri, dan sepanjang pengetahuan saya, tidak berisi materi yang ditulis oleh orang lain sebagai persyaratan penyelesaian studi di Universitas Negeri Yogyakarta atau perguruan tinggi lain, kecuali bagian-bagian tertentu saya ambil sebagai acuan dengan mengikuti kaidah penulisan karya ilmiah yang benar. Jika ternyata terbukti pernyataan ini tidak benar, sepenuhnya menjadi tanggung jawab saya.

Yogyakarta, 25 November 2016

Yang menyatakan



Ahmad Haris

NIM. 13507134018

MOTTO

DARE TO DREAM

LEMBAR PERSEMBAHAN

Dengan segala kerendahan hati, ingin kupersembahkan sebuah karya sederhana yang telah berhasil kuselasaikan ini kepada;

Rasa syukur kepada ALLAH SWT atas segala rahmat dan hidayah-Nya. Taburan cinta dan kasih sayang-Mu telah memberikanku kekuatan, membekaliku dengan ilmu serta memperkenalkanku dengan cinta. Atas karunia serta kemudahan yang Engkau berikan akhirnya Laporan Proyek Akhir yang sederhana ini dapat terselesaikan. Sholawat dan salam selalu terlimpahkan kepada Rasulullah Muhammad SAW.

Kupersembahkan karya sederhana ini kepada orang yang sangat kukasihi dan kusayangi. Ibu, Bapak. Sebagai tanda bakti, hormat, dan rasa terima kasih yang tiada terhingga yang telah memberikan kasih sayang, segala dukungan, dan cinta kasih yang tiada terhingga yang tiada mungkin dapat kubalas hanya dengan selembar kertas yang bertuliskan kata cinta dan persembahan. Semoga ini menjadi langkah awal untuk membuat Ibu dan Bapak bahagia karna kusadar, selama ini belum bisa berbuat yang lebih. Untuk Ibu dan Bapak yang selalu membuatku termotivasi dan selalu memberikan kasih sayang, selalu mendoakanku, selalu menasehatiku menjadi lebih baik.

Seluruh Dosen Pengajar Teknik Elektronika, terima kasih banyak untuk semua ilmu, didikan dan pengalaman yg sangat berarti yang telah kalian berikan. Untuk Dosen Pembimbing saya Dessy Irmawati, M.T. terima kasih bimbingan dan bantuannya selama ini, atas nasihat dan pelajaran yang saya dapatkan, saya tidak akan lupa atas bantuan dan kesabaran dari ibu.

Teman-temanku di kelas B Teknik Elektronika 2013, sungguh kebersamaan yang kita bangun selama ini telah banyak merubah kehidupanku. Kemarahanmu telah menuntunku menuju kedewasaan, senyummu telah membuka cakrawala dunia dan melepaskan belenggu-belenggu ketakutanku, tetes air mata yang mengalir di pipimu telah mengajarku arti kepeduliaan yang sebenarnya, dan gelak tawamu telah membuatku bahagia. Sungguh aku bahagia bersamamu, bahagia memiliki kenangan indah dalam setiap bait pada paragraf kisah persahabatan kita. Bila Tuhan memberikanku umur panjang, akan aku bagi harta yang tak ternilai ini (persahabatan) dengan anak dan cucuku kelak.

*“Ya Allah, jadikanlah Iman, Ilmu dan Amal ku sebagai lentera
jalan hidupku keluarga dan saudara seimanku”*

PROYEK AKHIR
PEMANTAU ISI KULKAS MENGGUNAKAN ETHERNET SHIELD R3
BERBASIS ARDUINO UNO R3

Oleh: Ahmad Haris

NIM: 13507134018

ABSTRAK

Tujuan pemantau isi kulkas menggunakan Ethernet *shield* R3 berbasis arduino uno R3 adalah untuk mengetahui informasi di dalam kulkas tanpa harus membuka kulkas yaitu dengan mengetahui kondisi sebenarnya di dalam kulkas dari jarak jauh.

Pemantau isi kulkas dirancang menggunakan Arduino uno R3 sebagai mikrokontroler, module camera VC0706 sebagai pengambil gambar didalam kulkas, Ethernet shield R3 sebagai media pengirim gambar dan led sebagai notifikasi *load* photo. Gambar yang dihasilkan dapat diterima melalui smartphone atau web dari jarak jauh menggunakan komunikasi wifi atau internet.

Berdasarkan hasil pengujian dapat disimpulkan bahwa pemantau isi kulkas dapat bekerja sesuai dengan prinsip yang telah dirancang. Perangkat lunak yang ada di mikrokontroler juga dapat berjalan sesuai sintaksisnya. Selain itu, aplikasi android atau web dapat menerima gambar melalui komunikasi wifi dengan jarak maksimal 15 meter.

Kata Kunci : Pemantau isi kulkas, Arduino Uno R3, *Module Camera* VC0706, Ethernet shield R3.

KATA PENGANTAR

Assamu'alaikum wr. wb.

Puji syukur kehadiran ALLAH SWT atas segala rahmat dan hidayah-Nya, penulis dapat melaksanakan Laporan Proyek Akhir ini dapat terselesaikan tanpa halangan yang berarti. Sholawat dan salam semoga tercurah pada Qudwah kita Rasulullah SAW keluarga, sahabat dan orang-orang yang istiqomah di jalan-Nya.

Dalam menyusun Laporan Proyek Akhir ini penulis merasa banyak kekurangan karena terbatasnya kemampuan dan pengetahuan penulis. Oleh karena itu pada kesempatan ini penulis menyampaikan ucapan terima kasih dan penghargaan yang sebesar-besarnya kepada:

1. Bapak, Ibu, Adik dan seluruh keluarga atas dukungan yang telah diberikan.
2. Dr. Sri Waluyanti, M.Pd. selaku Pembimbing Proyek Akhir, Ketua Program Studi Diploma III Teknik Elektronika dan Koordinator Proyek Akhir Jurusan Pendidikan Teknik Elektronika Universitas Negeri Yogyakarta.
3. Dr. Fatchul Arifin, M.T. selaku Ketua Jurusan Pendidikan Teknik Elektronika Universitas Negeri Yogyakarta.
4. Dr. Widarto M.Pd. selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta.
5. Seluruh Dosen Pengajar Teknik Elektronika Universitas Negeri Yogyakarta atas bekal ilmu yang diberikan kepada penulis.
6. Teman-teman seperjuangan angkatan 2013 khususnya kelas B 2013 yang telah memberikan bantuan sehingga pembuatan proyek akhir ini dapat terselesaikan.
7. Syahrul Awalludin Sidiq, Nugroho Agus Sugandi, Ahmad Nurhari, Ridho Dias Kusumo, Fauzi Hulqi Arin, Muhammad Arif Nuryanto dan kelas B Teknik Elektronika 2013 yang telah membantu menemani dan memberikan dukungan dalam pembuatan proyek akhir.
8. Semua pihak yang tidak dapat kami sebutkan satu persatu yang telah membantu hingga terselesainya laporan ini.

Penulis menyadari bahwa laporan ini masih banyak kekurangan dan jauh dari sempurna walaupun penulis telah berusaha untuk mendekati kesempurnaan, maka penulis berharap para pembaca memberikan saran dan kritik yang membangun.

Akhir kata penulis mohon maaf yang sebesar-besarnya apabila ada kekeliruan didalam penulisan laporan ini.

Wassalamu'alaikum wr. Wb

Yogyakarta, 25 November 2016

Penulis

Ahmad Haris

DAFTAR ISI

	Halaman
JUDUL	i
LEMBAR PERSETUJUAN	Error! Bookmark not defined.
LEMBAR PENGESAHAN	Error! Bookmark not defined.
LEMBAR PERNYATAAN KEASLIAN	Error! Bookmark not defined.
MOTTO	v
LEMBAR PERSEMBAHAN	vi
PROYEK AKHIR	viii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL	xvi
DAFTAR LAMPIRAN	xvii
BAB I	18
PENDAHULUAN	18
A. Latar Belakang Masalah	18
B. Identifikasi Masalah	19
C. Batasan Masalah.....	19
D. Rumusan Masalah	19
E. Tujuan.....	20
F. Manfaat	20
G. Keaslian Gagasan	21
BAB II	23
PENDEKATAN PEMECAHAN MASALAH.....	23
A. Kulkas	23
B. Module Kamera VC0706	25
1. Struktur dan Prinsip Kerja Modul Kamera VC0706	26
2. Spesifikasi Kamera Module VC0706.....	27
3. Spesifikasi Lensa Modul Kamera VC0706	28
C. Ethernet Shield R3	29
1. Fungsi LED pada Ethernet Shield R3:	30
2. Spesifikasi Ethernet Shield R3 :.....	31

D.	Microcontroller Arduino UNO R3	31
1.	Spesifikasi Arduino Uno R3	32
2.	Pemrograman	33
3.	Proteksi.....	34
4.	Power Supply	34
5.	Memori	35
6.	Input dan Output (I/O).....	35
7.	Komunikasi	36
8.	Reset Otomatis (software).....	37
E.	Catu Daya	38
F.	Sistem Operasi Android	40
G.	Smartphone	41
BAB III	43
PERANCANGAN ALAT	43
A.	Analisa Kebutuhan	43
B.	Perancangan.....	44
C.	Pengembangan.....	46
D.	Pengujian Alat	56
1.	Uji Fungsional	57
2.	Uji Unjuk Kerja	58
BAB IV	59
PROSES, HASIL DAN, PEMBAHASAN	59
A.	Perancangan.....	59
1.	Perangkat Keras.....	59
2.	Perangkat Lunak (Software).....	63
3.	Implementasi dan Pengujian	66
a.	Pengujian Tegangan Output Power Supply	67
b.	Pengujian Kamera	70
c.	Pengujian Ethernet Shield R3	72
B.	Hasil Pengujian Pemantau isi kulkas	75
C.	Pembahasan	79
1.	Perangkat Keras (Hardware)	79
2.	Perangkat Lunak (Software).....	80
BAB V	82

KESIMPULAN DAN SARAN	82
A. Kesimpulan.....	82
B. Saran	83
DAFTAR PUSTAKA	84
LAMPIRAN.....	85

DAFTAR GAMBAR

	Halaman
Gambar 1. Kulkas Pintar	22
Gambar 2. Kulkas	24
Gambar 3. Modul Kamera VC0706	26
Gambar 4. struktur modul kamera VC0706.....	27
Sumber: (Itead, 2016)	27
Gambar 5. Ethernet Shield R3	29
Gambar 6. Arduino Uno R3	31
Sumber: (Anggi Adi Putra, 2014)	31
Gambar 7. Software Arduino IDE	33
Gambar 8. Konfigurasi Pin pada 7805	39
Sumber: (Marlin P. Jones, 2016)	39
Gambar 10. Blok Diagram Perancangan Alat.....	45
Gambar 11. Power supply 5V dan 12V	45
Gambar 12. Layout pcb 5V dan 12V	48
Gambar 13. Inisialisasi board pada Arduino Uno R3	49
Gambar 14. Inisialilasi port serial	49
Gambar 15. Flowchart progam keseluruhan.	52
Gambar 16. Pembuatan html menggunakan notepade++	53
Gambar 17. Tampilan html pada web browser.	54
Gambar 18. tampilan design pada app inventor	55
Gambar 19. Tampilan blocks program pada app invetor.....	55
Gambar 20. Tampilan apk pada smarphone android	56
Gambar 22. Layout PC.....	60
Gambar 23. Hasil Penyolderan pada PCB	61
Gambar 24. Desaign box.....	62
Gambar 26. Design Box Rangkain Arduino Uno R3	62
Gambar 27. Screen satu pada aplikasi android	64
Gambar 28. Pembuatan Firmaware Arduino	65
Gambar 29. Pembuatan HTML.....	66
Gambar 30. Pengujian tegangan input dan output regulator IC 7805	67
Gambar 31. Hasil pengujian tegangan input dan output regulator IC 7805	67
Gambar 35. Pengaturan IP pada komputer	73

Gambar 36. Pengecekan Komunikasi dengan PING	74
Gambar 37. Pengujian Menggunakan Aplikasi Smartphone Android dengan Resolusi 160x120 piksel.....	76
Gambar 38. Pengujian Menggunakan Aplikasi Smartphone Android dengan Resolusi 320x240 piksel.....	76
Gambar 39. Pengujian Menggunakan Aplikasi Smartphone Android dengan Resolusi 640x480 piksel.....	77
Gambar 40. Pengujian Menggunakan web browser dengan Resolusi 160x120 piksel.....	77
Gambar 41. Pengujian Menggunakan web browser dengan Resolusi 320x240 piksel.....	78
Gambar 42. Pengujian Menggunakan web browser dengan Resolusi 640x480 piksel.....	78
Gambar 43. Cara Kerja Alat pada Gambar Sebenarnya	81

DAFTAR TABEL

	Halaman
Tabel1. Spesifikasi kulkas.....	25
Tabel2. Spesifikasi Lensa Modul Kamera VC0706	28
Tabel3. Karakteristik Regulator Tegangan Seri 78xx.....	39
Tabel4. Alat yang digunakan	46
Tabel5. Bahan yang digunakan dalam pembuatan alat.....	47
Tabel6. Uji Fungsional.....	57
Tabel7. Uji Unjuk Kerja.....	58
Tabel8. Pengukuran tegangan 5 V IC regulator LM7805	68
Tabel9. Pengukuran tegangan 5 V IC regulator LM7805 dengan % error	69
Tabel10. Pengujian tegangan 12 V IC regulator LM7812	69
Tabel11. Pengujian tegangan 12 V IC regulator LM7812 dengan % error	70
Tabel12. Uji daya jangkauan connectivy dari Ethernet shield R3	75

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Design Rangkaian Alat.....	86
Lampiran 2. Gambar Rangkaian Module Kamera VC0706	87
Lampiran 3. Gambar Rangkaian Arduino Uno R3	88
Lampiran 4. Gambar Rangkaian Ethernet Shield R3.....	89
Lampiran 5. Source Code Arduino Uno R3 pada Pemantau isi kulkas	90

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Pada zaman *modern* seperti sekarang ini, teknologi berkembang sangat cepat. Perkembangan teknologi sangat bermanfaat bagi manusia disegala bidang. Selain itu, teknologi dapat meringankan aktifitas dan kegiatan manusia menjadi lebih mudah. Sehingga mendorong masyarakat untuk mengembangkan suatu teknologi sebagai piranti yang dapat mempermudah kegiatan manusia bahkan menggantikan peran manusia dalam suatu fungsi tertentu.

Perkembangan Teknologi saat ini membuat manusia melakukan sesuatunya dengan mudah, salah satunya adalah kulkas yang dapat mengawetkan makanan hingga beberapa hari dan dapat membekukan air menjadi es. Saat ini masyarakat bisa mengetahui kondisi di dalam kulkas ketika membuka kulkas dan tidak bisa diketahui ketika berada diluar rumah. Masyarakat tidak mempunyai banyak waktu akan sulit mengecek isi kulkas pada saat *realtime*. Hal ini tentu kurang efektif mengingat semakin banyaknya kegiatan di luar rumah. Melihat dari permasalahan yang ada, maka penulis akan membuat suatu alat yang mampu mengatasi masalah tersebut. Teknologi baru yang ada di dalam pemantau kulkas adalah kulkas dapat dilihat melalui *smartphone* berbasis aplikasi dan dapat mengetahui keadaan dari makanan tersebut apakah busuk atau tidak. Kelemahan dari pemantau isi kulkas yang sudah ada adalah tidak menggunakan website

sehingga ketika aplikasi *error* dan berada diluar rumah atau kantor maka tidak bisa dipantau.

B. Identifikasi Masalah

Dari uraian latar belakang masalah di atas, maka dapat dibuat suatu identifikasi masalah sebagai berikut:

1. Keterbatasan akses untuk mengecek isi kulkas pada saat *realtime*,
2. Kulkas pintar yang sudah ada tidak menggunakan website sehingga ketika aplikasi *smartphone error* tidak bisa dipantau,

C. Batasan Masalah

Berdasarkan latar belakang masalah dan identifikasi batasan masalah pada permasalahan tiga, perlu adanya batasan masalah sehingga ruang lingkup permasalahannya terfokus pada satu poin. Poin tersebut peneliti akan membahas tentang pembuatan dan pembahasan pemantau isi kulkas mini yang dimodifikasi dengan *module camera VC0706* sebagai pengambil gambar untuk memberi informasi kepada pemilik kulkas, selanjutnya gambar akan diolah menggunakan Arduino uno R3 dengan komunikasi Ethernet shield R3 agar dapat dilihat melalui aplikasi *smartphone* android dan web melalui komunikasi Wi-Fi, sehingga dapat membantu masyarakat yang mempunyai banyak kegiatan.

D. Rumusan Masalah

Dari identifikasi yang ada, maka dapat ditarik beberapa rumusan masalah, yaitu:

1. bagaimana merancang *hardware* dan *software* aplikasi android Pemantau isi kulkas menggunakan Ethernet shield R3 berbasis Arduino Uno R3?
2. bagaimana cara mengatur dan menyelaraskan Pemantau isi kulkas menggunakan Ethernet shield R3 berbasis Arduino Uno R3?
3. bagaimana unjuk kerja Pemantau isi kulkas menggunakan Ethernet shield R3 berbasis Arduino Uno R3?

E. Tujuan

Adapun tujuan pemantau isi kulkas menggunakan Ethernet shield R3 berbasis Arduino UNO R3 yaitu:

1. merealisasikan pemantau isi kulkas menggunakan Ethernet shield R3 berbasis Arduino UNO R3,
2. merealisasikan pengaturan dan penyelarasan pemantau isi kulkas menggunakan Ethernet shield R3 berbasis Arduino UNO R3,
3. mengetahui unjuk kerja pemantau isi kulkas menggunakan Ethernet shield R3 berbasis Arduino UNO R3.

F. Manfaat

Pembuatan proyek akhir ini diharapkan dapat bermanfaat bagi mahasiswa, lembaga pendidikan, dan industri. Adapun manfaat yang diharapkan dari pembuatan tugas akhir ini antara lain:

1. Bagi mahasiswa
 - a. sebagai tolak ukur individual setelah mendapatkan ilmu dari bangku kuliah dan kehidupan sehari-hari untuk diimplementasikan dalam bentuk suatu alat,

- b. untuk mengaplikasikan ilmu yang didapat selama di bangku kuliah dan menerapkan ilmunya secara nyata,
 - c. dapat digunakan sebagai bahan referensi atau pembelajaran dan penambah wawasan tentang optimasi sel surya menggunakan sampah elektronik bekas berbasis mikrokontroler khususnya Arduino UNO R3 serta sebagai kajian untuk pengembangan selanjutnya,
 - d. sebagai bentuk kontribusi terhadap Universitas dan pengabdian kepada masyarakat dalam bentuk karya alat yang bermanfaat.
2. Bagi prodi Teknik Elektronika
- a. sebagai wujud dari perkembangan Ilmu Pengetahuan dan Teknologi (IPTEK),
 - b. sebagai parameter kualitas dan kuantitas lulusan mahasiswa Fakultas Teknik Universitas Negeri Yogyakarta.
3. Bagi Dunia Industri
- Dapat digunakan sebagai pengembangan produk elektronika yang dapat diaplikasikan pada berbagai bidang khususnya pada pemanfaatan sampah elektronik bekas.

G. Keaslian Gagasan

Pemantau isi kulkas menggunakan Ethernet shield R3 berbasis Arduino UNO R3 sudah ada. Berikut gambar pemantauan kulkas yang sudah ada.



Gambar 1. Kulkas Pintar (Yossie Dwi Pranantojan, 2016)

Perbedaan dari pemantau kulkas yang penulis buat menggunakan Ethernet *shield* R3 sebagai pengirim data dan arduino uno sebagai mikrokontroler.

Berlanjut dari hal tersebut penulis membuat pemantau isi kulkas yang dapat mengirim berupa gambar pada *smartphone* android melalui jaringan Wi-Fi menggunakan Ethernet *shield* R3 dengan memanfaatkan *handphone* sebagai penerimanya.

BAB II

PENDEKATAN PEMECAHAN MASALAH

Dalam pembuatan Pemantau isi kulkas menggunakan Ethernet shield R3 berbasis Arduino UNO R3 ini diperlukan pendekatan pemecahan masalah yang meliputi kajian-kajian teori perangkat *hardware* dan *software* yang digunakan.

A. Kulkas

Kulkas atau lemari es atau lemari pendingin adalah sebuah alat rumah tangga listrik yang menggunakan *refrigerasi* (proses pendingin) untuk menolong pengawetan makanan. Kulkas bekerja menggunakan pompa panas pengubah fase beroperasi dalam sebuah putaran *refrigeration*. Kulkas terdiri dari lemari pendingin atau lemari pembeku atau keduanya. Sistem dua lemari ini diperkenalkan pertama kali oleh General Electric pada 1939. Beberapa kulkas sekarang dibagi menjadi empat ruang untuk penyimpanan jenis makanan yang berbeda: (1) $-18\text{ }^{\circ}\text{C}$ ($-64.4\text{ }^{\circ}\text{F}$) (pembeku), (2) $0\text{ }^{\circ}\text{C}$ ($32\text{ }^{\circ}\text{F}$) (daging), (3) $4\text{ }^{\circ}\text{C}$ ($39.2\text{ }^{\circ}\text{F}$) (pendingin), (4) $10\text{ }^{\circ}\text{C}$ ($50\text{ }^{\circ}\text{F}$) (sayuran) untuk menaruh berbagai jenis makanan (Ihda Amiruddin, 2014).

Pada tahun 1766 pada tanggal 9 Juli, Jacob Perkins adalah seorang ilmuwan dan ahli mesin menemukan sebuah mesin pendingin yang berfungsi untuk menyimpan makanan, daging, sayuran, buah-buahan dan air minum, produk telur dan susu untuk beberapa hari agar makan-makanan tersebut tidak mudah busuk atau alat ini juga dapat mengeraskan air yang biasa kita konsumsi sehari-hari. Alat ini biasanya disebut sebagai kulkas atau *refrigator* (Baidatul Muchlisin Asti dan Junaidi Abdul Munif, 2009).

Saat ini kulkas dirancang semakin canggih dan praktis. Misalnya dirancang sedemikian rupa sehingga aroma makanan dengan daging mentah tidak akan tercampur. Ada juga yang dilengkapi dengan dispenser air dan es batu. Beragam ukuran kulkas dapat dipilih, dari kulkas mimbar hingga kulkas 2 pintu yang sangat besar ukurannya.

Peneletian ini penulis menggunakan kulkas mini untuk dimodifikasi menjadi pemantau isi kulkas yang dapat berkomunikasi dengan *smartphone*. Kulkas mini yang digunakan mempunyai dimensi (WxDxH) 44x45.2x51.1 dengan *volume* 45 l. Berikut gambar kulkas.



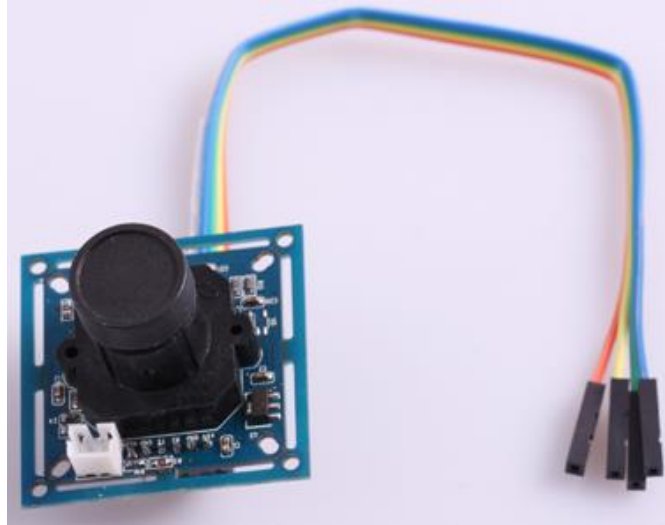
Gambar 2. Kulkas

Tabel.1 Spesifikasi kulkas

Spesifikasi kulkas	
Jenis	kulkas tanpa freezer
Tempat	terpisah
Warna / Coating Material	putih / plastik
Pengelolaan	elektromekanis
Jumlah pintu	1
Dimensi (WxDxH)	44x45.2x51.1 melihat
Mencairkan ruangan pendingin	Panduan
Total volume	45 l
Pemindahan pendingin	40 l
Rak-rak	logam

B. Module Kamera VC0706

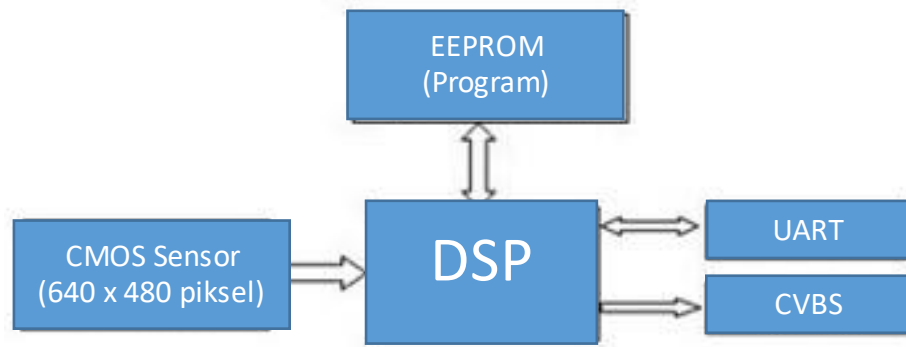
Kamera *module* VC0706 adalah sebuah kamera yang menggunakan *chip controller* VIMICRO VC0706 dan menggunakan komunikasi serial (RS323 atau TTL). Camera VC0706 mempunyai kualitas standar VGA dan 2 output yaitu NTSC video dan *snapshot* yang bisa dikirimkan menggunakan komunikasi serial, selain itu *camera* VC0706 mempunyai 3 ukuran gambar yang bisa diatur diantaranya 640x480 , 320x240 atau 160x120 (*pixels*) yang masing-masing ukuran gambarnya dapat tersimpan dengan *format* .jpg. Berikut struktur *module camera* VC0706 ditunjukkan pada Gambar 3.



Gambar 3. Modul Kamera VC0706 (Dany, 2014)

1. Struktur dan Prinsip Kerja Modul Kamera VC0706

Secara umum struktur module kamera VC0706 terdiri dari EEPROM vimecro vc0706 dapat saling berkamunikasi dengan DSP atau *Host Processor* yang menerima data dari CMOS sensor kemudian bisa komunikasi timbal balik dengan komunikasi UART dan bisa diterima atau ditampilkan oleh CVBS (*Composite Video Blanking dan Sync*). berikut gambar struktur modul kamera VC0706 yang ditunjukkan pada Gambar 3.



Gambar 4. struktur modul kamera VC0706 (Itead, 2016)

2. Spesifikasi Kamera Module VC0706

Video output	: CVBS 30fps
PCB dimensi	: 38 × 38mm / 32 × 32mm
Sensor gambar	: 1/4 CMOS sensor gambar MT9V011
Format gambar	: PAL, 628 × 582; NTSC, 510 × 492; <i>default</i> seperti NTSC untuk modul ini, Pal tanpa resistor R5 (10K), gratis bagi pengguna untuk memilih
Format gambar	: JPEG
Ukuran gambar	: VGA / QVGA / QQVGA CIF / QCIF / QQCIF, umumnya secara <i>default</i> sebagai QVGA (320 * 240), pengguna bebas untuk mengubah ukuran * sesuai dengan protokol komunikasi. Semakin kecil gambar, semakin cepat gambar tersebut dikirim.
Baud rate	: 9600bps-115200bps, <i>default</i> seperti 38400, pengguna bebas untuk mengubah tingkat sesuai dengan protokol komunikasi.

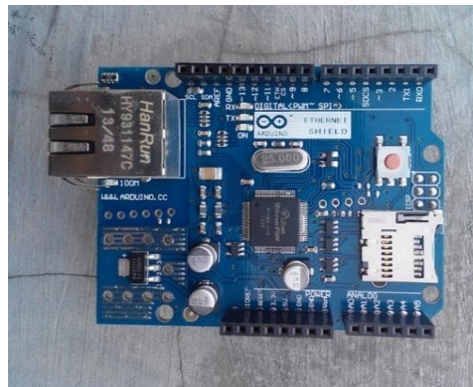
Lens	: FOV 60 ° ~ 120 °, 650 lensa biasa / 850, 940 IR
lensa	
Terminal blok	: 6pin-2.0mm / 2pin-2.0mm (antarmuka papan cahaya inframerah)
Level	: Standar sebagai tingkat CMOS, dapat dimodifikasi untuk tingkat TTL sesuai dengan persyaratan
Bekerja tegangan	: DC4.8V ~ DC6.5V
Bekerja saat ini	: 90mA (ketika cahaya inframerah tidak bekerja)
Bekerja suhu	: -20 °C ~ + 60 °C
Suhu penyimpanan	: -30 °C ~ 70 °C
Kelembaban	: 90% non-kondensasi

3. Spesifikasi Lensa Modul Kamera VC0706

Tabel 2. Spesifikasi Lensa Modul Kamera VC0706

Item	Standar	Pilihan
<i>Baud Rate</i>	38400bps	9600-115200
Level	RS232 CMOS	TTL
Tegangan Input	5V	3.3V
Kualitas Gambar	VGA	QVGA/QQVGA/CIF/QCIF/QQCIF
Lensa	3.6 mm (650)	1.8/2.1/2.5/3.6/4.3/8/12 mm (IR650/850/940/no IR Filter)
Format Video	NTSC	PAL
Ukuran PCB	38x38 mm	32x32 mm

C. Ethernet Shield R3



Gambar 5. Ethernet Shield R3

Ethernet Shield menambah kemampuan arduino *board* agar terhubung ke jaringan komputer. Ethernet shield berbasis *chip* ethernet Wiznet W5100. Ethernet *library* digunakan dalam menulis program agar arduino *board* dapat terhubung ke jaringan dengan menggunakan Arduino ethernet shield. Pada ethernet shield terdapat sebuah slot *micro-SD*, yang dapat digunakan untuk menyimpan *file* yang dapat diakses melalui jaringan. *Onboard micro-SD card reader* diakses dengan menggunakan *SD library*. Arduino *board* berkomunikasi dengan W5100 dan SD card menggunakan bus SPI (Serial Peripheral Interface). Komunikasi ini diatur oleh *library* SPI.h dan Ethernet.h. Bus SPI menggunakan *pin digital* 11, 12 dan 13 pada Arduino Uno. Pin digital 10 digunakan untuk memilih W5100 dan *pin digital* 4 digunakan untuk memilih SD *card*. Pin-pin yang sudah disebutkan sebelumnya tidak dapat digunakan untuk *input/output* umum ketika kita menggunakan Ethernet Shield. Karena W5100 dan SD *card* berbagi bus SPI, hanya salah satu yang dapat aktif pada satu waktu. Jika menggunakan kedua perangkat dalam program, hal ini akan diatasi oleh *library* yang sesuai. Jika tidak menggunakan 15 salah satu

perangkat dalam program kita, kiranya kita perlu secara eksplisit men-*deselect*-nya. Untuk melakukan hal ini pada SD card, set pin 4 sebagai *output* dan menuliskan logika tinggi padanya, sedangkan untuk W5100 yang digunakan adalah pin 10. Ethernet shield ini mempunyai *standard* RJ-45 sebagai koneksi internet.

Perisai saat ini memiliki *Power over* Ethernet (PoE) modul dirancang untuk mengekstrak listrik dari kabel Ethernet *twisted pair* sebagai berikut:

- a. IEEE802.3af
- b. Rendah kebisingan (100mVpp)
- c. Input rentang tegangan 36V ke 57V
- d. *Overload* dan tegangan pendek perlindungan
- e. Keluaran 9V Efisiensi
- f. Tinggi DC / DC converter: typ 75% @ beban 50%
- g. 1500V Isolasi (input ke output)

1. Fungsi LED pada Ethernet Shield R3:

- a. PWR: Menunjukkan bahwa papan dan perisai yang bertenaga
- b. LINK: menunjukkan adanya link jaringan dan berkedip ketika ethernet shield R3 mentransmisikan atau menerima data
- c. FULLD: Menunjukkan bahwa koneksi jaringan adalah full duplex
- d. 100M: Menunjukkan adanya koneksi jaringan 100 Mb / s (sebagai lawan 10 Mb / s)
- e. RX: Berkedip ketika perisai menerima data
- f. TX: Berkedip ketika perisai mengirimkan data
- g. COLL: Berkedip ketika tabrakan jaringan terdeteksi

2. Spesifikasi Ethernet Shield R3 :

- a. *chip* Wiznet W5100 dengan *internal buffer* 16 Kb,
- b. kecepatan koneksi 10 / 100Mb (*Fast-Ethernet*).
- c. Papan ini terhubung dengan Arduino melalui *port* SPI.
- d. Dapat mendukung hingga 4 koneksi simultan

D. Microcontroller Arduino UNO R3

Arduino Uno R3 adalah papan pengembangan (*development board*) mikrokontroler yang berbasis *chip* ATmega328P. *Board* ini berfungsi sebagai arena *prototyping* sirkuit mikrokontroler. Dengan menggunakan papan pengembangan, akan lebih mudah merangkai rangkaian elektronika mikrokontroler dibanding jika anda memulai merakit ATmega328 dari awal di *breadboard*.



Gambar 6. Arduino Uno R3 (Anggi Adi Putra, 2014)

Arduino Uno memiliki 14 *digital pin input / output* (atau biasa ditulis I/O), dimana 6 pin diantaranya dapat digunakan sebagai *output* PWM), 6 pin input analog, menggunakan *crystal* 16 MHz, koneksi USB, *jack* listrik, *header* ICSP dan tombol *reset*. Hal tersebut adalah semua yang diperlukan untuk mendukung

sebuah rangkaian mikrokontroler. Cukup dengan menghubungkannya ke komputer dengan kabel USB atau diberi *power* dengan adaptor AC-DC atau baterai, Arduino UNO sudah dapat digunakan. Kemungkinan paling buruk hanyalah kerusakan pada chip ATmega328, yang bisa diganti sendiri dengan mudah dan dengan harga yang relatif murah.

Kata "Uno" berasal dari bahasa Italia yang berarti "satu", dan dipilih untuk menandai peluncuran *Software* Arduino (IDE) versi 1.0. Arduino. Sejak awal peluncuran hingga sekarang, Uno telah berkembang menjadi versi Revisi 3 atau biasa ditulis REV 3 atau R3. *Software* Arduino IDE, yang bisa di-*install* di Windows maupun Mac dan Linux, berfungsi sebagai *software* yang membantu anda memasukkan (*upload*) program ke *chip* ATmega328 dengan mudah.

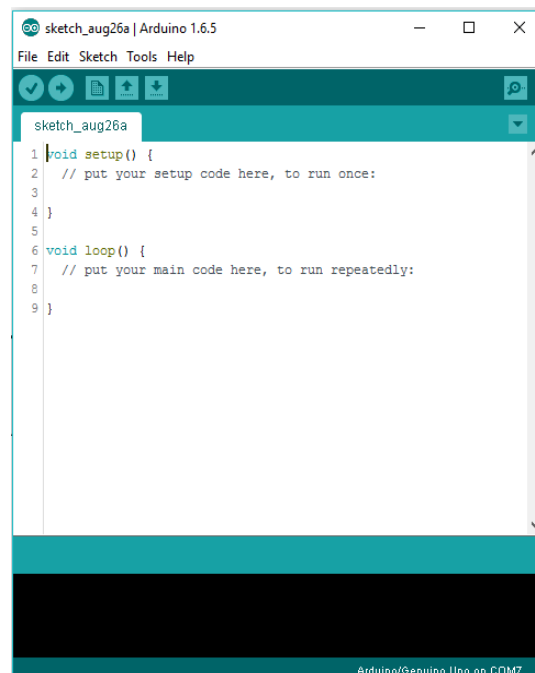
1. Spesifikasi Arduino Uno R3

Chip mikrokontroler	: ATmega328P
Tegangan operasi	: 5V
Tegangan input (yang direkomendasikan, via jack DC)	: 7V - 12V
Tegangan input (limit, via jack DC)	: 6V - 20V
Digital I/O pin	: 14 buah, 6 diantaranya menyediakan PWM
Analog Input pin	: 6 buah
Arus DC per pin I/O	: 20 mA
Arus DC pin 3.3V	: 50 mA
Memori Flash	: 32 KB, 0.5 KB telah digunakan untuk bootloader

SRAM	: 2 KB
EEPROM	: 1 KB
Clock speed	: 16 Mhz
Dimensi	: 68.6 mm x 53.4 mm
Berat	: 25 g

2. Pemrograman

Pemrograman board Arduino dilakukan dengan menggunakan Arduino *Software* (IDE). *Chip* ATmega328 yang terdapat pada Arduino Uno R3 telah diisi program awal yang sering disebut *bootloader*. *Bootloader* tersebut yang bertugas untuk memudahkan pemrograman yang lebih sederhana menggunakan Arduino *Software*, tanpa harus menggunakan tambahan *hardware* lain. Selain itu di dalam Arduino *Software* sudah diberikan banyak contoh program atau *library* yang bisa langsung di *upload* ke *board* Arduino.



```
sketch_aug26a | Arduino 1.6.5
File Edit Sketch Tools Help
sketch_aug26a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

Arduino/Genuino Uno on COM7

Gambar 7. Software Arduino IDE

Arduino Uno R3 telah dilengkapi dengan *chip* ATmega16U2 yang telah diprogram sebagai konverter USB to Serial. *Firmware* ATmega16U2 di *load* oleh DFU *bootloader*, dan untuk merubahnya dapat menggunakan *software* Atmel Flip (*Windows*) atau DFU *programmer* (Mac OSX dan Linux), atau menggunakan header ISP dengan menggunakan *hardware external programmer*.

3. Proteksi

Development board Arduino Uno R3 telah dilengkapi dengan *polyfuse* yang dapat di-*reset* untuk melindungi *port* USB komputer/laptop dari korsleting atau arus berlebih. Meskipun kebanyakan komputer telah memiliki perlindungan port tersebut didalamnya namun sikring pelindung pada Arduino Uno memberikan lapisan perlindungan tambahan menghubungkan Arduino ke komputer dengan aman. Jika lebih dari 500mA ditarik pada port USB tersebut, sirkuit proteksi akan secara otomatis memutuskan hubungan, dan akan menyambung kembali ketika batasan aman telah kembali.

4. Power Supply

Board Arduino Uno dapat ditenagai dengan power yang diperoleh dari koneksi kabel USB, atau *via power supply* eksternal. Pilihan power yang digunakan akan dilakukan secara otomatis

External power supply dapat diperoleh dari adaptor AC-DC atau bahkan baterai, melalui jack DC yang tersedia, atau menghubungkan langsung GND dan pin Vin yang ada di board. *Board* dapat beroperasi dengan power dari *external power supply* yang memiliki tegangan antara 6V hingga 20V. Namun ada beberapa hal yang harus anda perhatikan dalam rentang tegangan ini. Jika

diberi tegangan kurang dari 7V, pin 5V tidak akan memberikan nilai murni 5V, yang mungkin akan membuat rangkaian bekerja dengan tidak sempurna. Jika diberi tegangan lebih dari 12V, *regulator* tegangan bisa over heat yang pada akhirnya bisa merusak pcb. Dengan demikian, tegangan yang di rekomendasikan adalah 7V hingga 12V. Beberapa pin power pada Arduino Uno: (1) GND. Ini adalah ground atau negatif, (2) Vin. Ini adalah pin yang digunakan jika power langsung diberikan ke board Arduino dengan rentang tegangan yang disarankan 7V - 12V, (3) Pin 5V. Ini adalah pin *output* dimana pada pin tersebut mengalir tegangan 5V yang telah melalui regulator, (4) 3V3. Ini adalah pin *output* dimana pada pin tersebut disediakan tegangan 3.3V yang telah melalui regulator, (5) IOREF. Ini adalah pin yang menyediakan referensi tegangan mikrokontroler. Biasanya digunakan pada *board shield* untuk memperoleh tegangan yang sesuai, apakah 5V atau 3.3V.

5. Memori

Chip ATmega328 pada Arduino Uno R3 memiliki memori 32 KB, dengan 0.5 KB dari memori tersebut telah digunakan untuk *bootloader*. Jumlah SRAM 2 KB, dan EEPROM 1 KB, yang dapat di baca-tulis dengan menggunakan EEPROM *library* saat melakukan pemrograman.

6. Input dan Output (I/O)

Seperti yang telah disebutkan sebelumnya, Arduino Uno memiliki 14 buah digital pin yang dapat digunakan sebagai input atau output, sengan menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digital(Read)*. Pin-pin tersebut bekerja pada tegangan 5V, dan setiap pin dapat menyediakan atau menerima arus 20mA, dan memiliki tahanan *pull-up* sekitar 20-50k ohm

(secara *default* dalam posisi *disconnect*). Nilai *maximum* adalah 40mA, yang sebisa mungkin dihindari untuk menghindari kerusakan *chip* mikrokontroller. Beberapa pin memiliki fungsi khusus: (1) Serial, terdiri dari 2 pin : pin 0 (RX) dan pin 1 (TX) yang digunakan untuk menerima (RX) dan mengirim (TX) data serial, (2) External Interrupts, yaitu pin 2 dan pin 3. Kedua pin tersebut dapat digunakan untuk mengaktifkan *interrupts*. Gunakan fungsi *attachInterrupt()*, (3) PWM: Pin 3, 5, 6, 9, 10, dan 11 menyediakan *output* PWM 8-bit dengan menggunakan fungsi *analogWrite()* (4) SPI : Pin 10 (SS), 11 (MOSI), 12 (MISO), dan 13 (SCK) mendukung komunikasi SPI dengan menggunakan *SPI Library*, (5) LED : Pin 13. Pada pin 13 terhubung built-in led yang dikendalikan oleh digital pin no 13, (6) TWI : Pin A4 (SDA) dan pin A5 (SCL) yang mendukung komunikasi TWI dengan menggunakan *Wire Library*.

Arduino Uno memiliki 6 buah input analog, yang diberi tanda dengan A0, A1, A2, A3, A4, A5. Masing-masing pin analog tersebut memiliki resolusi 10 bits (jadi bisa memiliki 1024 nilai). Secara *default*, pin-pin tersebut diukur dari *ground* ke 5V, namun bisa juga menggunakan pin AREF dengan menggunakan fungsi *analogReference()*. Beberapa in lainnya pada board ini adalah AREF sebagai referensi tegangan untuk input analog. *Reset* hubungkan ke *LOW* untuk melakukan *reset* terhadap mikrokontroller. Sama dengan penggunaan tombol *reset* yang tersedia.

7. Komunikasi

Arduino Uno R3 memiliki beberapa fasilitas untuk berkomunikasi dengan komputer, berkomunikasi dengan Arduino lainnya, atau dengan

mikrokontroler lainnya. Chip Atmega328 menyediakan komunikasi serial UART TTL (5V) yang tersedia di pin 0 (RX) dan pin 1 (TX). Chip ATmega16U2 yang terdapat pada board berfungsi menterjemahkan bentuk komunikasi ini melalui USB dan akan tampil sebagai *Virtual Port* di komputer. *Firmware* 16U2 menggunakan driver USB standar sehingga tidak membutuhkan *driver* tambahan.

Pada Arduino *Software* (IDE) terdapat *monitor* serial yang memudahkan data textual untuk dikirim menuju Arduino atau keluar dari Arduino. Led TX dan RX akan menyala berkedip-kedip ketika ada data yang ditransmisikan melalui chip USB to Serial via kabel USB ke komputer. Untuk menggunakan komunikasi serial dari digital pin. Chip ATmega328 juga mendukung komunikasi I2C (TWI) dan SPI. Di dalam Arduino *Software* (IDE) sudah termasuk *Wire Library* untuk memudahkan anda menggunakan bus I2C. Untuk menggunakan komunikasi SPI.

8. Reset Otomatis (*software*)

Pada mikrokontroler lain ketika melakukan pemrograman mikrokontroler, harus menekan tombol reset sesaat sebelum melakukan *upload* program. Pada Arduino Uno, hal ini tidak lagi merepotkan. Arduino Uno telah dilengkapi dengan auto reset yang dikendalikan oleh software pada komputer yang terkoneksi. Salah satu jalur flow control (DTR) dari ATmega16U pada Arduino Uno R3 terhubung dengan jalur reset pada ATmega328 melalui sebuah kapasitor 100nF. Ketika jalur tersebut diberi nilai *LOW*, mikrokontroler akan di reset. Dengan demikian proses *upload*

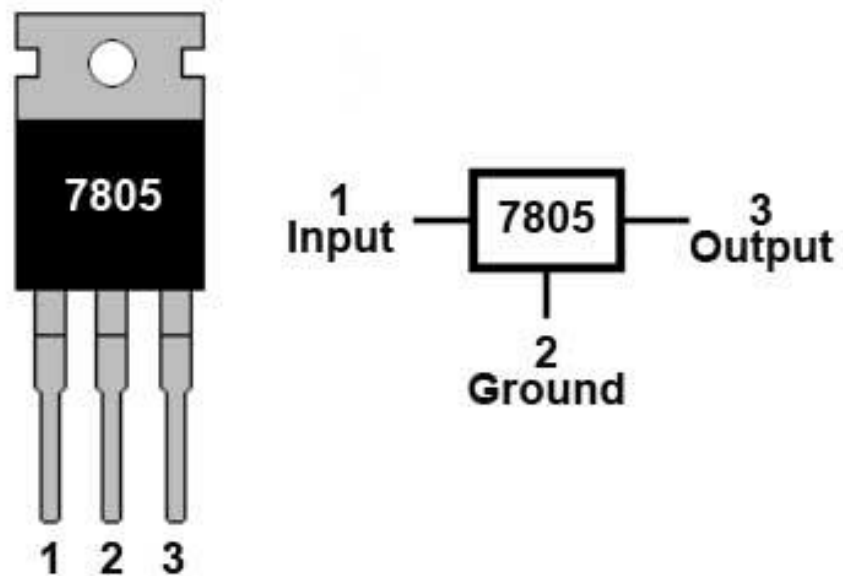
akan jauh lebih mudah dan tidak harus menekan tombol *reset* pada saat yang tepat seperti biasanya.

E. Catu Daya

Catu daya (*power supply*) adalah suatu rangkaian elektronika yang dapat menghasilkan tegangan keluar yang stabil, tegangan keluaran dapat berupa tegangan AC maupun tegangan DC. *Supply* daya atau tegangan catu suatu rangkaian elektronika yang berubah-ubah besarnya (baik berubah membesar maupun mengecil) dapat menyebabkan pengaruh yang sifatnya merusak rangkaian elektronika yang dicatunya (Sunomo, 1996:78).

Fungsi dari regulator tegangan merupakan komponen yang berfungsi untuk menstabilkan tegangan. Seri 78XX adalah regulator tegangan positif dengan tiga terminal seperti yang ditunjukkan pada gambar 2. Regulator ini memiliki kemampuan mengeluarkan arus yang besarnya bervariasi sesuai dengan tipe yang diberikan oleh pabrik.

7805 Pinout



Gambar 8. Konfigurasi Pin pada 7805 (Marlin P. Jones, 2016)

Tabel 3. Karakteristik Regulator Tegangan Seri 78xx

Type	V Out (V)	I Out (A)			V in (V)	
		78xxC	78Lxx	78Mxx	Min	Max
7805	5	1	0,1	0,5	7,5	20
7806	6	1	0,1	0,5	8,6	21
7808	8	1	0,1	0,5	10,6	23
7809	9	1	0,1	0,5	11,7	24
7810	10	1	0,1	0,5	12,7	25
7812	12	1	0,1	0,5	14,8	27
7815	15	1	0,1	0,5	18	30
7818	18	1	0,1	0,5	21	33
7824	24	1	0,1	0,5	27,3	38

Dalam proyek akhir ini besarnya catu daya yang dibutuhkan sebesar +5 volt sehingga menggunakan IC regulator seri 7805 yang dapat diperoleh dalam

kemasan TO-220 plastik dan logam. IC regulator 7805 dapat mengeluarkan arus melebihi 0,5A apabila dilengkapi dengan peredam *heatshink* yang memadai dan pada daya kurang atau sama dengan 15 watt. Karakteristik regulator tegangan seri 78XX ada di tabel 2.

Rangkaian terpadu (*integrate Circuit = IC*) tipe 7805 ini adalah regulator yang dapat menstabilkan tegangan searah positif dengan memasukan +7,5 volt sampai +12 volt dengan keluaran +5 volt. IC regulator seri 7805 sesuai tabel 2 yang mempunyai karakteristik menstabilkan tegangan positif dengan memasukan tegangan +7,5 volt sampai +20 volt, tegangan keluaran IC 7805 adalah +5 volt DC teregulasi, arus keluaran melebihi 0,5A, penggunaan *intern* terhadap pembebasan lebih termik, tidak memerlukan tambahan komponen *ekstern*, pembatas arus hubungan singkat *intern*.

F. Sistem Operasi Android

Sistem operasi Android merupakan sebuah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak.

Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi,



Gambar 9. Logo Android (Aingandra, 2009)

Termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google *Mail Services* (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD).

G. Smartphone

Telepon pintar (*smartphone*) adalah telepon genggam yang mempunyai kemampuan tingkat tinggi, kadang-kadang dengan fungsi yang menyerupai komputer. Belum ada standar pabrik yang menentukan arti telepon pintar (Elcom, 2011). Bagi beberapa orang, telepon pintar merupakan telepon yang bekerja menggunakan seluruh perangkat lunak sistem operasi yang menyediakan hubungan standar dan mendasar bagi pengembang aplikasi. Bagi

yang lainnya, telepon cerdas hanyalah merupakan sebuah telepon yang menyajikan fitur canggih seperti surel (surat elektronik), internet dan kemampuan membaca buku elektronik (*e-book*) atau terdapat papan ketik (baik sebagaimana jadi maupun terhubung keluar) dan penyambung VGA. Dengan kata lain, telepon cerdas merupakan komputer kecil yang mempunyai kemampuan sebuah telepon (Erick Managam, 2012).

BAB III

PERANCANGAN ALAT

Proyek akhir ini dalam pengerjaannya menggunakan metode rancang bangun, langkah-langkah dari metode rancang bangun antara lain yaitu, analisa kebutuhan, perancangan, pembuatan dan pengujian. Data hasil pengujian diperoleh dengan cara observasi menyagkut rancang bangun dan unjuk kerja alat. Teknik analisis data yang digunakan dalam proyek akhir ini adalah secara deskriptif.

A. Analisa Kebutuhan

Berdasarkan identifikasi kebutuhan yang ada, maka diperoleh beberapa analilis kebutuhan terhadap alat yang akan dibuat dengan spesifikasi sebagai berikut:

1. Perangkat Power Supply

Perangkat power supply 5V dan 12V digunakan sebagai sumber daya pada kulkas mini menggunakan Ethernet Shield R3.

2. Perangkat Mikrokontroler

Pada pembuatan pemantau isi kulkas menggunakan Ethernet shield membutuhkan Arduino Uno R3 sebagai mikrokontroler.

3. Smartphone Android

Smartphone dibutuhkan untuk mengirim perintah berupa kode melalui perantara Jaringan web atau html yang kemudian diproses pada mikrokontroler.

4. Perangkat Ethernet Shield R3

Ethernet shield R3 sebagai pengirim data melalui Jaringan web atau html agar dapat ditampilkan melalui *smartphone*.

5. Kulkas Mini Daewoo

Kulkas mini Daewoo yang mempunyai ukuran dengan dimensi (WxDxH) (44x45.2x51.1)

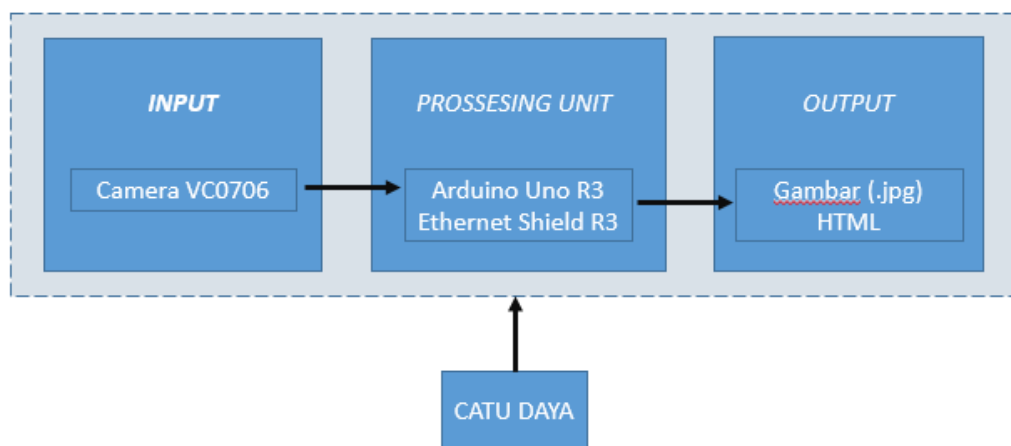
6. *Module Camera VC0706*

Peneliti menggunakan modul kamera VC0706 karena kamera dapat bekerja dalam suhu $-20^{\circ} \sim +60^{\circ}$. Jadi dapat digunakan di dalam kulkas dengan suhu minimal 0° . Camera Module VC0706 yang digunakan sebagai pengambil gambar pada kulkas mini.

B. Perancangan

1. Perancangan Perangkat Keras

Secara garis besar diagram dari rangkaian pemantau isi kulkas menggunakan Ethernet shield R3 berbasis Arduino uno R3 adalah sebagai berikut :



Gambar 10. Blok Diagram Perancangan Alat.

a. Blok *Input*

Blok *input* tersiri dari *module camera* VC0706 yang berfungsi untuk pengambilan gambar pada pemantau isi kulkas yang kemudian gambar tersebut bisa dilihat melalui *smartphone* dalam bentuk *html*.

b. Proses (Mikrokontroler Arduino Uno R3)

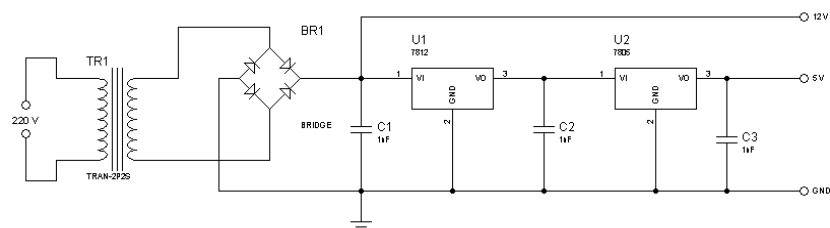
Sistem *control* yang digunakan adalah *system* mikrokontroler Arduino uno R3 dengan rancang bangun yang disesuaikan agar sesuai dengan Ethernet shield R3 dan *module camera* VC0706.

c. Blok *Output*

Pada output ini menggunakan motor servo sebagai penggerak camera VC sesuai keinginan user bagian mana yang ingin dilihat, yaitu pada bagian pintu atau pada bagian kulkas. Gambar yang dihasilkan oleh kamera VC0706 adalah berupa format *.jpg* dalam bentuk *html*.

2. Perancangan Rangkain *Power Supply*

Power supply sangat penting untuk menyuplai tegangan ke *system* mikrokontroler Arduino uno R3, Ethernet shield, *camera* VC0706 dan led. *Power supply* pada pemantau isi kulkas menggunakan 5V dan 12V

Gambar 11. *Power supply* 5V dan 12V

C. Pengembangan

1. Pengembangan Alat

Alat yang digunakan dalam pengembangan pemantau isi kulkas menggunakan Ethernet shield R3 berbasis Arduino uno R3 yaitu:

- 1) Bor tangan digunakan untuk melubangi board PCB sebagai tempat komponen.
- 2) Tang gunting untuk memotong atau mengupas gunting.
- 3) Multimeter untuk mengukur arus, tegangan ataupun hambatan pada alat.
- 4) Solder untuk mensolder (merekatkan) komponen pada layout PCB.
- 5) Obeng untuk memasang atau melepas skrup.
- 6) Gunting untuk memotong kabel-kabel dan bahan lain.
- 7) Tool set digunakan sebagai perlengkapan bahan pembuatana alat.

Untuk lebih jelasnya mengenai rincian alat yang dibutuhkan dapat dilihat pada tabel 4.

Tabel 4. Alat yang digunakan

No	Nama Bahan	Jumlah
1	Bor tangan	1
2	Tang gunting	1
3	Multimeter	1
4	Solder	1
5	Tenol	1
6	Obeng	1
7	Lem Tembak	1
8	Gunting	1
9	Tool set	1

- b. Bahan yang digunakan pada pengembangan alat ini adalah:

Tabel 5. Bahan yang digunakan dalam pengembangan alat

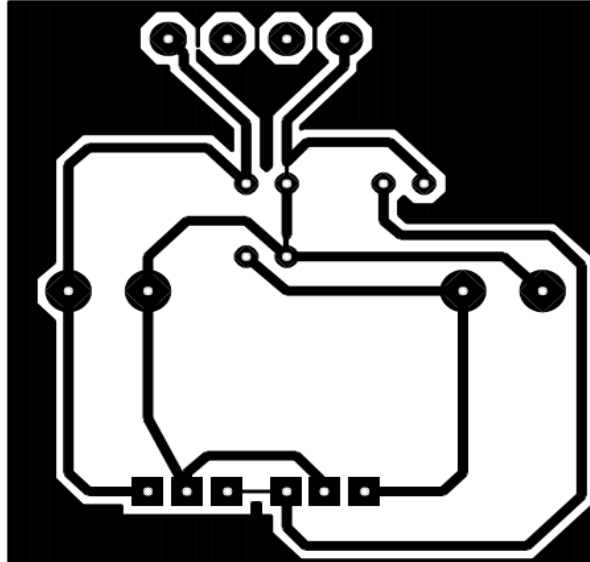
No	Nama Bahan	Jumlah
1	Dioda IN4148	1
2	Travo 1A	1
3	Jack DC	1
4	Jack AC	1
5	Led Merah 3 mm	1
6	Led Merah 5 mm	1
7	Capasitor 22uF	2
8	Regulator 7805	1
9	Regulator 7812	1
10	Pcb	1
11	Akrilik	1

- c. Pengembangan layout dan papan PCB

Dalam pengembangan layout terlebih dahulu memproses skematik menjadi bentuk desain layout pcb yang siap dipakai. Berikut adalah hasil tahap penerapan dari perancangan *layout power supply*. *Layout power supply* yang dibuat dengan menggunakan *software*, setelah *layout* jadi kemudian dicetak menggunakan kertas PCB. Berikut langkah-langkahnya:

1. Menyiapkan alat dan bahan yang dibutuhkan. Dalam hal ini sudah dicantumkan pada tabel.
2. Merancang gambar skematik rangkaian.
3. Mendesain layout PCB dan mencetak layout PCB.
4. Memindahkan hasil cetakan pada kertas ke PCB.
5. Melarutkan PCB menggunakan FeCl₃.
6. Mengebor PCB.
7. Memasang dan meyolder komponen.

8. Memasukan program kedalam mikrokontroller.
9. Melakukan pengujian.



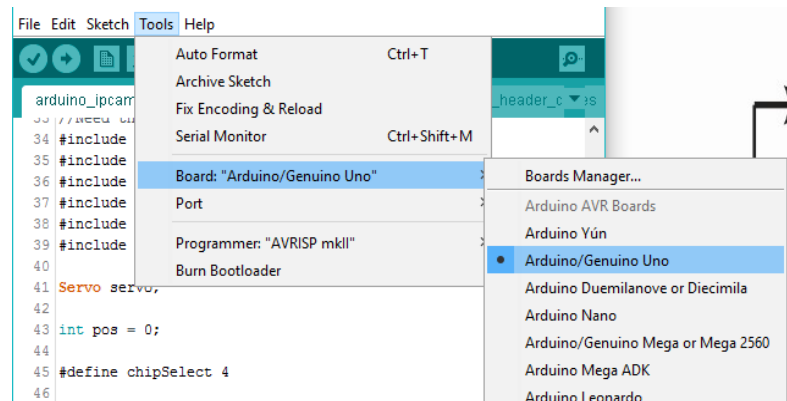
Gambar 12. *Layout* pcb 5V dan 12V

2. Pengembangan Software

1. Program

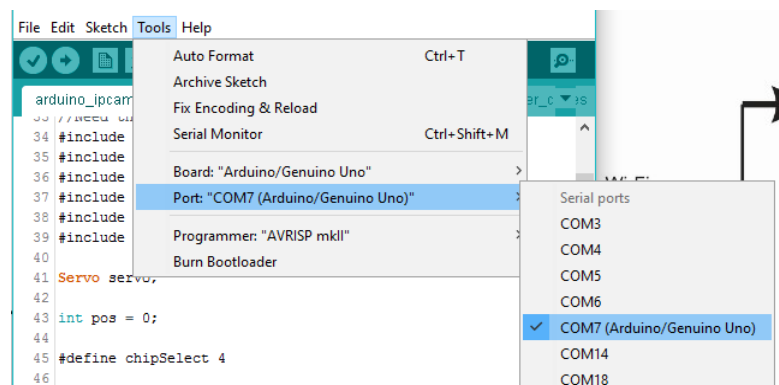
Pembuatan perangkat lunak mikrokontroler Arduino uno R3 ini digunakan untuk memberikan langkah-langkah yang harus dikerjakan mikroprosesor sebagai perangkat kendali rangkaian pada pemantau isi kulkas menggunakan Ethernet Shield R3 berbasis Arduino uno R3. Hal awal yang dilakukan pada setiap pemrograman adalah inisialisasi pin input output. Inisialisasi yang dilakukan adalah sebagai berikut:

1. Inisialisasi model mikrokontroler yang digunakan



Gambar 13. Inisialisasi *board* pada Arduino Uno R3

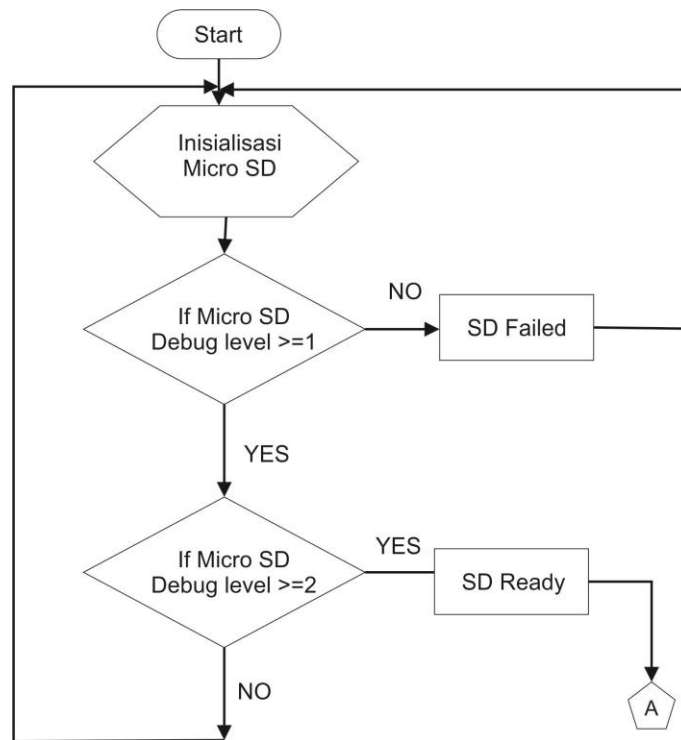
2. Inisialisasi *port* serial

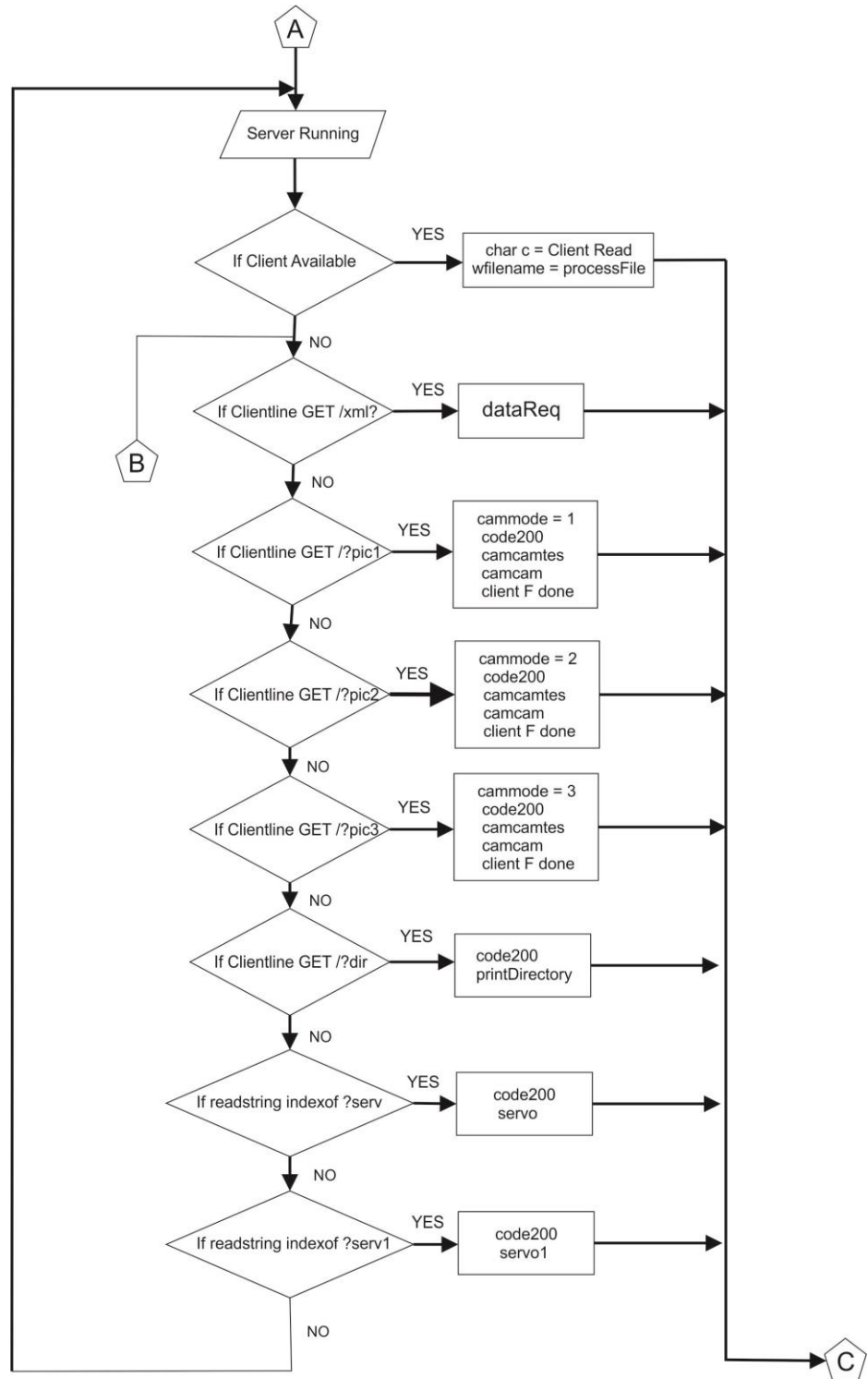


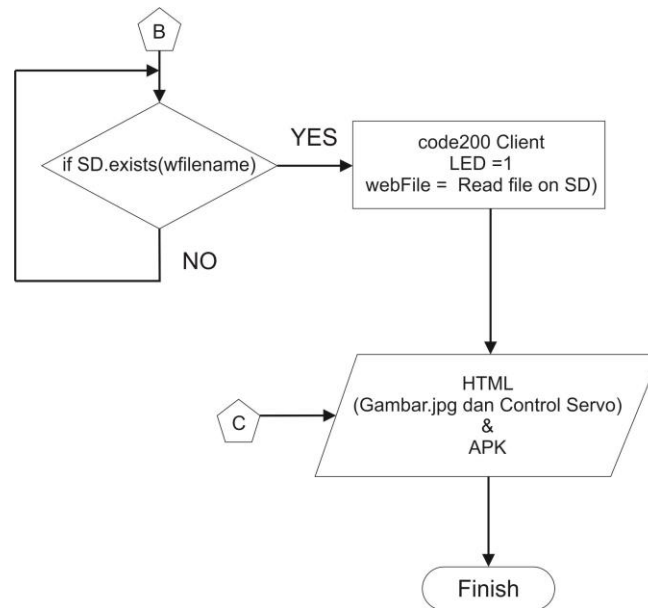
Gambar 14. Inisialisasi port serial

Perancangan menggunakan bahasa pemrograman C pada software arduino, program yang telah dibuat kemudian dicompile sehingga akan diperoleh file dengan ekstensi *.ino. File ini inilah yang akan didownload ke Arduino Uno R3. Perancangan program ini dilakukan dengan membuat diagram alir (*flowchart*) terlebih dahulu. *Flowchart* keseluruhan dari program pemantau isi kulkas menggunakan Ethernet shield berbasis Arduino uno R3 ditunjukkan pada point 7.

2. Flowchart







Gambar 15. Flowchart program keseluruhan.

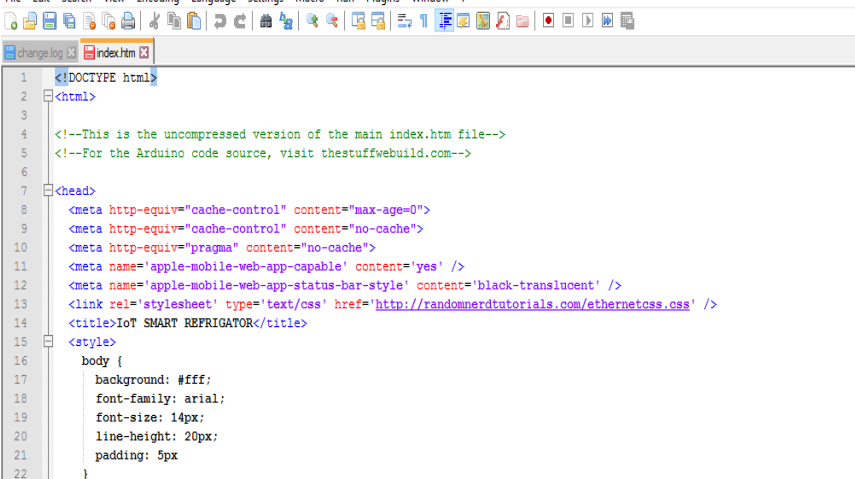
3. Proses pengembangan html menggunakan notepad++

HTML adalah bahasa *markup* internet (web) berupa kode dan simbol yang dimasukkan kedalam sebuah *file* yang ditujukan untuk ditampilkan didalam sebuah *website*. Singkatnya, HTML adalah bahasa *markup* yang digunakan untuk membuat *website*. *Website* yang dibuat dengan HTML ini, dapat dilihat oleh semua orang yang terkoneksi dengan internet. Tentunya dengan menggunakan aplikasi penjelajah internet (*browser*) seperti Internet Explorer, Mozilla Firefox dan Google Chrome.

1. *HyperText* adalah metode dimana kita "berpindah" disekeliling web, dengan mengklik sebuah teks yang bernama *hyperlink*. *Hyperlink* adalah sebuah teks khusus di internet, dimana saat teks tersebut diklik, akan membawa kita ke halaman web selanjutnya/halaman web lain yang telah ditentukan.

2. *Markup* adalah hal yang dilakukan oleh tag HTML kepada teks yang ada didalamnya. HTML menandai teks yang berada didalamnya sebagai tipe teks tertentu. Misalnya saja jika kita menandai sebuah teks dengan tag html `<i>`, maka teks tersebut akan berubah menjadi *italic* (huruf yang miring). Sedangkan jika kita menandainya dengan ``, maka teks tersebut akan berubah menjadi **bold** (huruf tebal).

3. *Language* yang berarti bahasa. HTML adalah sebuah bahasa, yang memiliki kata kata berupa kode dan *syntax* seperti bahasa yang lain. Saat ini bahasa HTML masih terus dikembangkan Hal ini dikarenakan pengguna internet semakin hari semakin berkembang pesat. Oleh karena itu bahasa HTML harus ditingkatkan lagi agar bisa menciptakan halaman web yang lebih berkualitas. Untuk itulah dibentuk sebuah organisasi yang bertanggung jawab mengembangkan bahasa HTML. Organisasi ini bernama W3C.

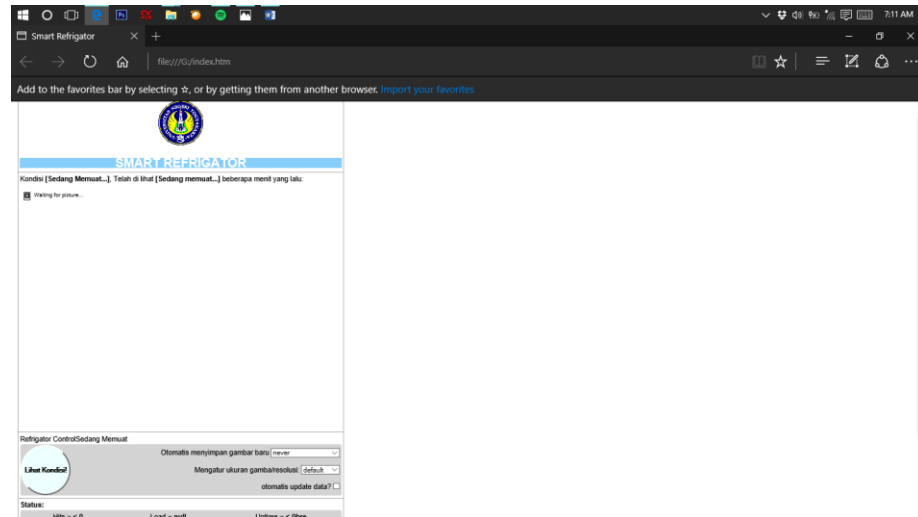


```

1 <!DOCTYPE html>
2 <html>
3
4 <!--This is the uncompressed version of the main index.htm file-->
5 <!--For the Arduino code source, visit thestuffwebuild.com-->
6
7 <head>
8 <meta http-equiv="cache-control" content="max-age=0">
9 <meta http-equiv="cache-control" content="no-cache">
10 <meta http-equiv="pragma" content="no-cache">
11 <meta name="apple-mobile-web-app-capable" content="yes" />
12 <meta name="apple-mobile-web-app-status-bar-style" content="black-translucent" />
13 <link rel="stylesheet" type="text/css" href="http://randomnerdtutorials.com/ethernetcss.css" />
14 <title>IoT SMART REFRIGATOR</title>
15 <style>
16   body {
17     background: #fff;
18     font-family: arial;
19     font-size: 14px;
20     line-height: 20px;
21     padding: 5px
22   }

```

Gambar 16. Pembuatan html menggunakan notepad++

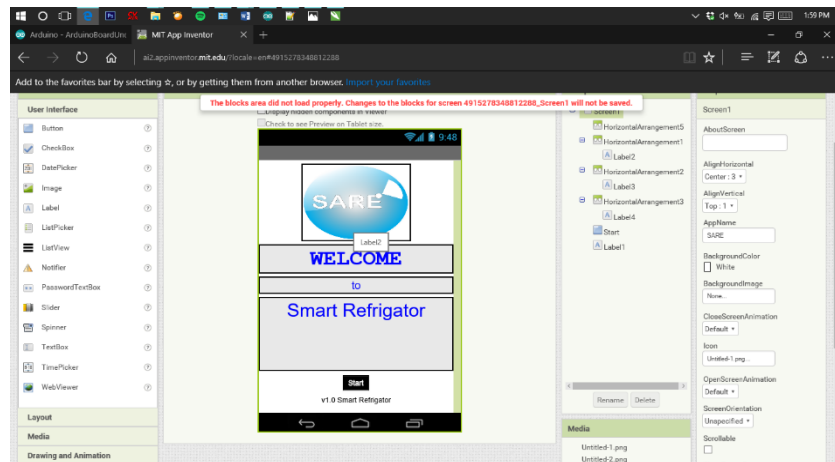


Gambar 17. Tampilan html pada web browser.

4. Pembuatan apk “Sare” menggunakan app inventor

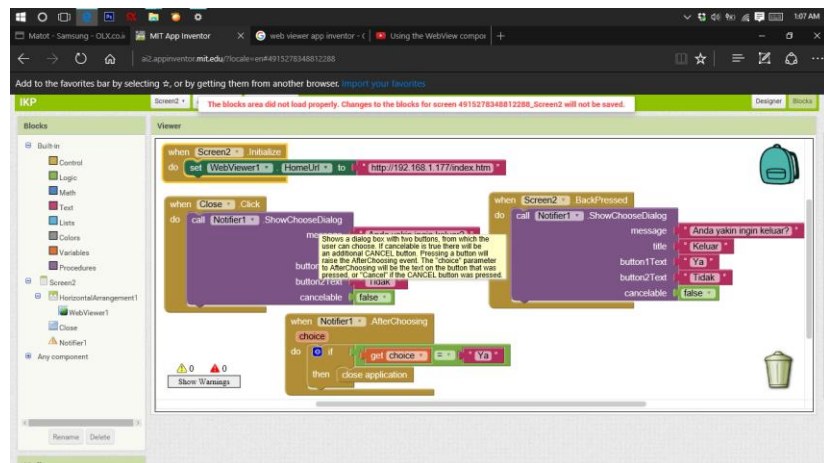
App Inventor adalah sebuah aplikasi web *open-source* yang awalnya disediakan oleh Google, dan sekarang dikelola oleh Massachusetts Institute of Technology (MIT). Didesain dengan sistem visual - konstruksionis dan dapat dioperasikan hanya menggunakan peramban (*browser*) selain itu lebih mudah dalam pembuatan dan pemrograman karena tidak terlalu kompleks. MIT AppInventor, beserta logo dan seluruh sumber daya didalamnya dilisensikan dalam *Creative Commons Attribution-ShareAlike 3.0 Unported License*, © 2012-2015 *Massachusetts Institute of Technology*.

1. Tampilan design apk android pada app inventor



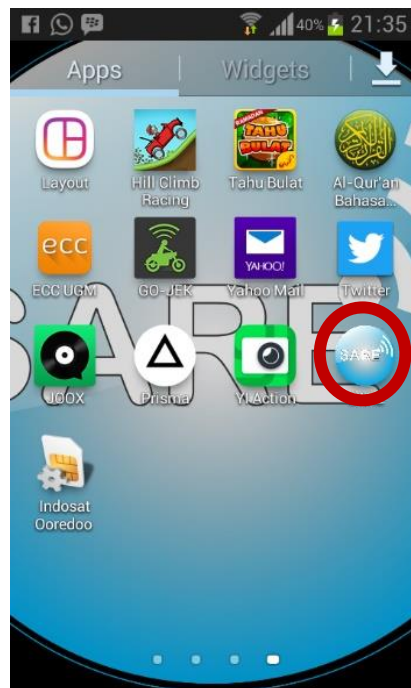
Gambar 18. tampilan design pada app inventor

2. Tampilan block program apk adroid pada app inventor



Gambar 19. Tampilan *blocks* program pada app inventor

3. Hasil apk android pada app inventor



Gambar 20. Tampilan apk pada *smartphone* android

Pada gambar 31 terdapat lingkaran merah yang menunjukkan bahwa lingkaran tersebut adalah hasil dari apk yang telah dibuat melalui app inventor yang diberi nama “SARE” dengan singkatan *Smart Refrigerator*.

D. Pengujian Alat

Pengujian alat dilakukan untuk mendapatkan data penelitian serta mengetahui fungsi alat yang telah dibuat. Dalam pengujian dilakukan dengan dua pengujian yaitu:

1. Uji Fungsional

Pengujian dilakukan dengan cara menguji setiap bagian alat berdasarkan karakteristik dan fungsi masing-masing. Pengujian dilakukan untuk mengetahui apakah setiap bagian dari perangkat telah bekerja sesuai dengan fungsi dan keinginan, hasil uji fungsional ditunjukkan oleh tabel 6.

Tabel 6. Uji Fungsional

No	Alat	Indikator	Kebutuhan/Pengukuran
1	<i>Power Supply</i>	Tegangan <i>output</i> 5 volt	5 volt dan 12 volt
2	Ethernet Shield 3	Dapat mengirimkan gambar dengan format .jpg	Gambar .jpg
3	VC0706	Dapat mengambil gambar dengan ukuran yang berbeda-beda	640x480,320x240, dan 160x120 (<i>pixels</i>)
4	Voltmeter	Dapat membaca tegangan <i>input</i> dari sumber daya	11,45 volt sampai dengan 12 volt
6	<i>SD Card</i>	Dapat menyimpan data yang diambil dari kamera.	Format gambar.jpg

2. Uji Unjuk Kerja

Pengujian unjuk kerja alat dilakukan dengan cara melihat unjuk kerja alat. Hal-hal yang perlu diamati antara lain: rangkaian catu daya, arduino, module camera VC0706 dan komunikasi data menggunakan Ethernet shield R3 berbasis jaringan, hasil uji fungsional ditunjukkan oleh tabel 6.

Tabel 7. Uji Unjuk Kerja

No	Indikator	Langkah pengujian	Standar pengukuran
1	Tegangan kerja 5 volt dan 12 volt	Mengukur <i>output</i> dari <i>power supply</i> menggunakan multimeter.	Multimeter
2	Html sebagai media penerima data untuk menampilkan gambar dan pengontrolan pada servo	Mengecek komunikasi Ethernet shield R3 melalui serial monitor, web browser dan apk android dengan mengubah ip jaringan pada laptop yang digunakan sebagai server.	Serial Monitor laptop dan <i>handphone</i>
3	Dapat membaca tegangan <i>input</i> dari sumber daya	Membandingkan pembacaan voltmeter dengan multimeter.	Multimeter
4	Dapat menyimpan data berupa gambar dalam format gambar.jpg	Mengecek file yang disimpan pada <i>MicroSD Card</i> menggunakan laptop	Laptop

BAB IV

PROSES, HASIL DAN, PEMBAHASAN

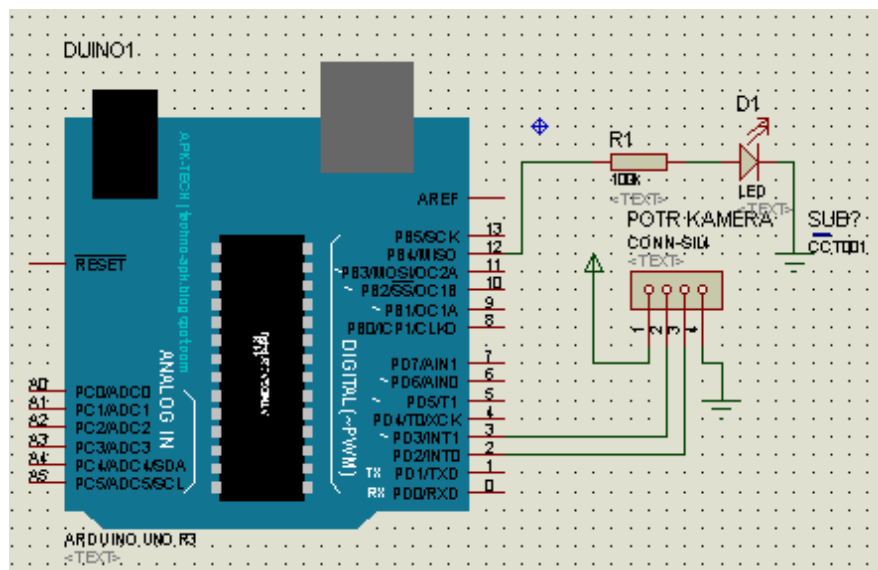
Tujuan dari pengujian dan pembahasan adalah untuk mengetahui kinerja alat baik secara per bagian blok rangkaian maupun sistem keseluruhan apakah sudah seperti yang diharapkan atau belum. Pengujian ini meliputi:

A. Perancangan

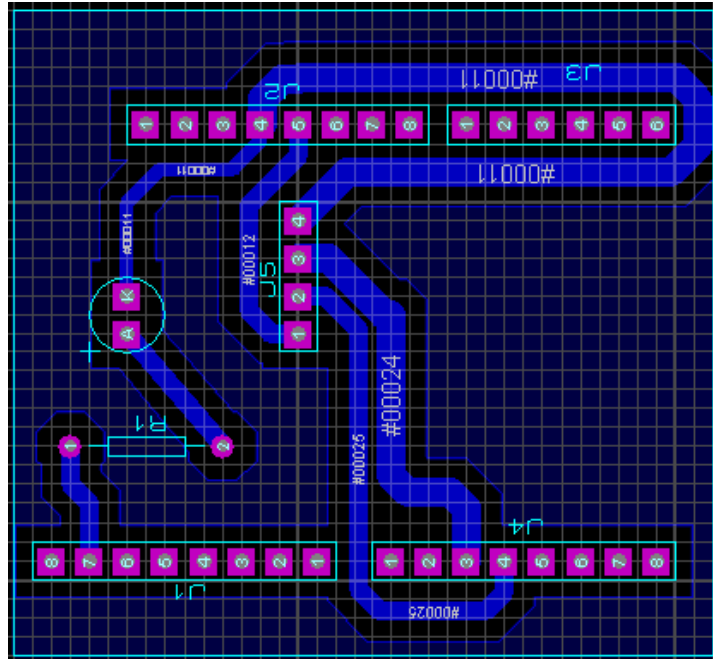
1. Perangkat Keras

a) *Layout PCB*

Perancangan *layout* rangkaian menggunakan software proteus kemudian rancangan *layout* tersebut di *converter* ke ares supaya bisa *men-design layout* di PCB.



Gambar 21. Rangkaian Alat pada ISIS Proteus

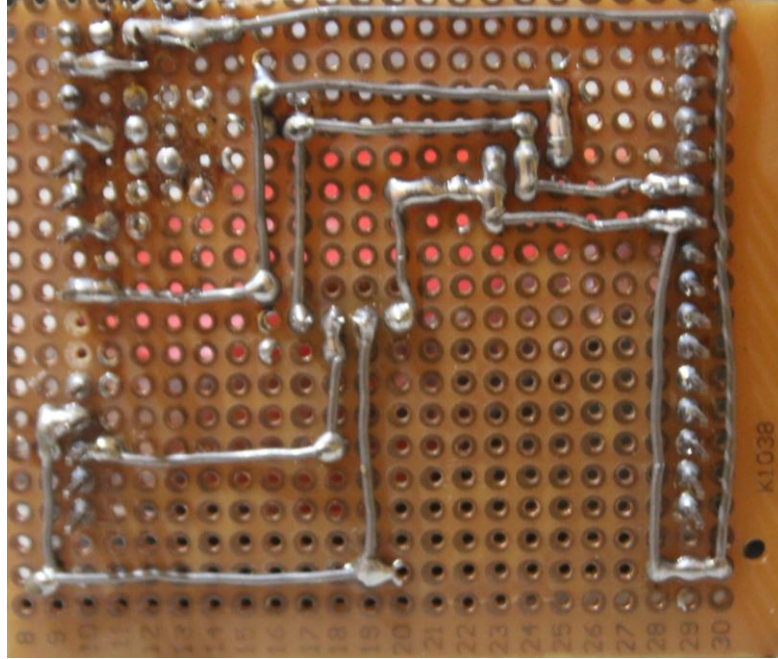


Gambar 22. Layout PC

Sebelum melakukan pencetakan di PCB, terlebih dahulu harus mendesign dengan menggunakan *software* proteus agar hasil jalur pcb terlihat rapi seperti yang terlihat pada gambar , kemudian dilakukan pencetakan di PCB jenis lubang dengan teknik penyolderan langsung pada PCB lubang, hasil pencetakan ke PCB padat langsung dilihat pada gambar .

b) Perakitan Komponen

Langkah selanjutnya melakukan pencetakan pada PCB sekaligus pemasangan komponen pada PCB bolong dengan *layout* yang telah dibuat.

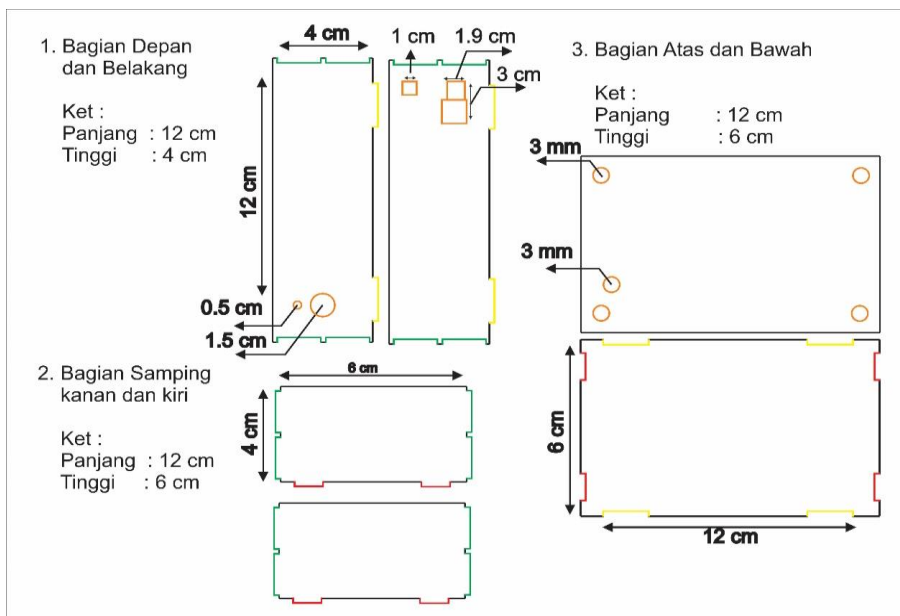


Gambar 23. Hasil Penyolderan pada PCB

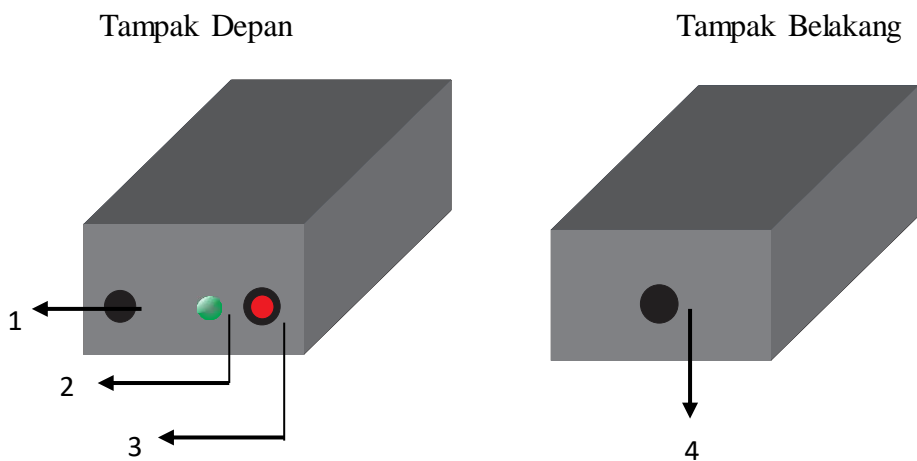
Perakitan komponen PCB dilakukan dengan menyolder terlebih dahulu komponen *white housing* dan *black housing*, Gambar 23 menunjukkan perakitan komponen pada PCB. Perakitan dan penyolderan dilakukan dengan sangat hati-hati agar tidak ada jalur yang *short*.

c) Perancangan *Desaign Box*

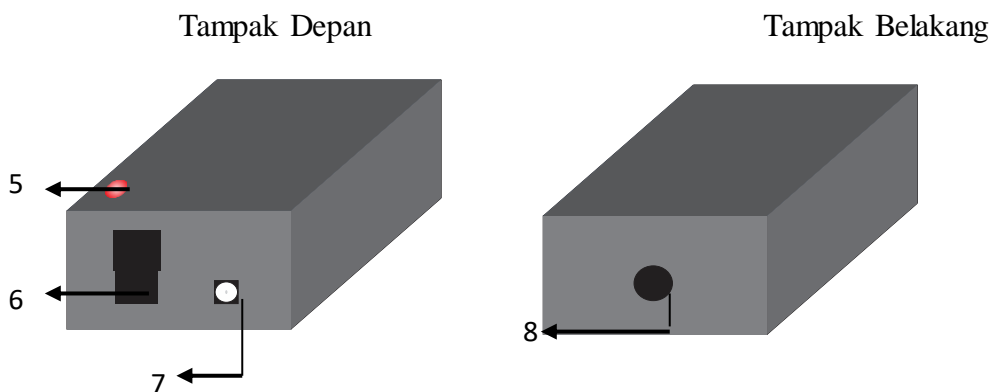
Melakukan perancangan *design box* dengan menggunakan *software* corel draw x7, kemudian merealisasikan *design* dengan menggunakan bahan *box universal*.



Gambar 24. Desain box



Gambar 25. Design Box Power Supply



Gambar 26. Design Box Rangkain Arduino Uno R3

Ket :

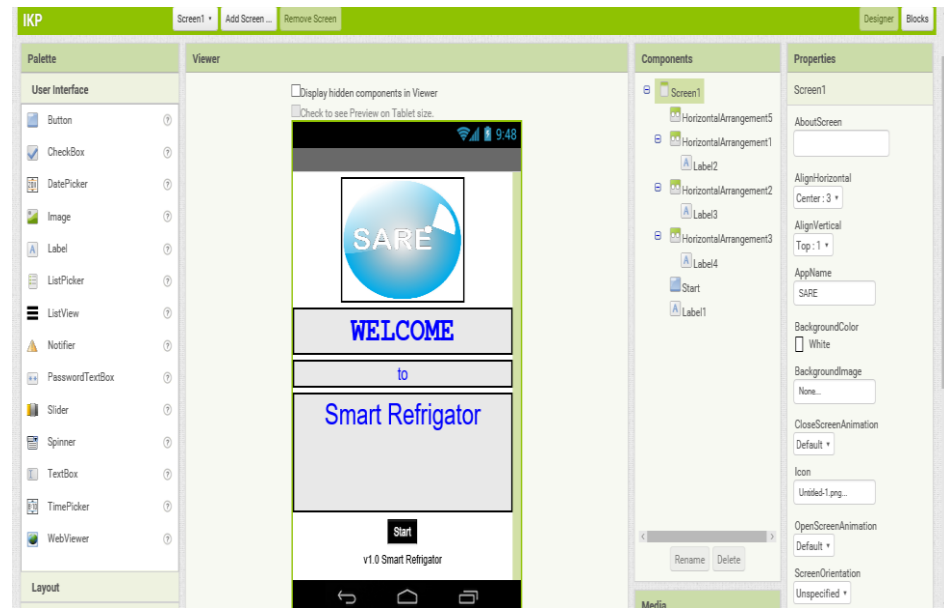
1. Lubang kabel jack dc untuk Arduino Uno R3
2. Lampu indikator bahwa *power supply* dapat bekerja
3. Button *on – off* pada *power supply*
4. Lubang untuk kabel AC dan Lampu pada kulkas mini
5. Lampu *indicator* untuk gambar yang sedang diphoto dengan tanda lampu berkedip.
6. Lubang untuk *jack* RJ45 dan *jack* untuk *upload* ke Arduino Uno R3
7. Lubang untuk *Jack* DC pada Arduino Uno R3
8. Lubang untuk kabel kamera.

2. Perangkat Lunak (*Software*)

Proyek akhir ini menggunakan tiga perangkat lunak yang dibuat dengan beberapa aplikasi berbeda, yaitu;

a) Pembuatan Aplikasi Android

Pembuatan aplikasi android ini menggunakan app inventor. Gambar dibawah ini adalah gambar *screen* satu dan *screen* dua pada aplikasi android.



Gambar 27. Screen satu pada aplikasi android

b) Pembuatan *Firmware* Arduino

Pembuatan *firmware* Arduino uno R3 menggunakan *software* Arduino 1.6.5.

The screenshot shows the Arduino IDE interface with the following code in the editor:

```

1 |
2 #define DebugLevel 2 //(1)+sd,client requests,ambil gambar.
3
4 #define BusyLED 13
5
6
7 #include <Adafruit_VC0706.h>
8 #include <SoftwareSerial.h>
9 #include <SPI.h>
10 #include <Ethernet.h>
11 #include <SD.h>
12
13 SoftwareSerial cameraconnection = SoftwareSerial(2, 3);
14 Adafruit_VC0706 cam = Adafruit_VC0706(&cameraconnection);
15
16 //memulai vars penamaan file dari 0
17 int fname = 0;
18
19 //vars untuk menghitung waktu and time since last pic (tslp)

```

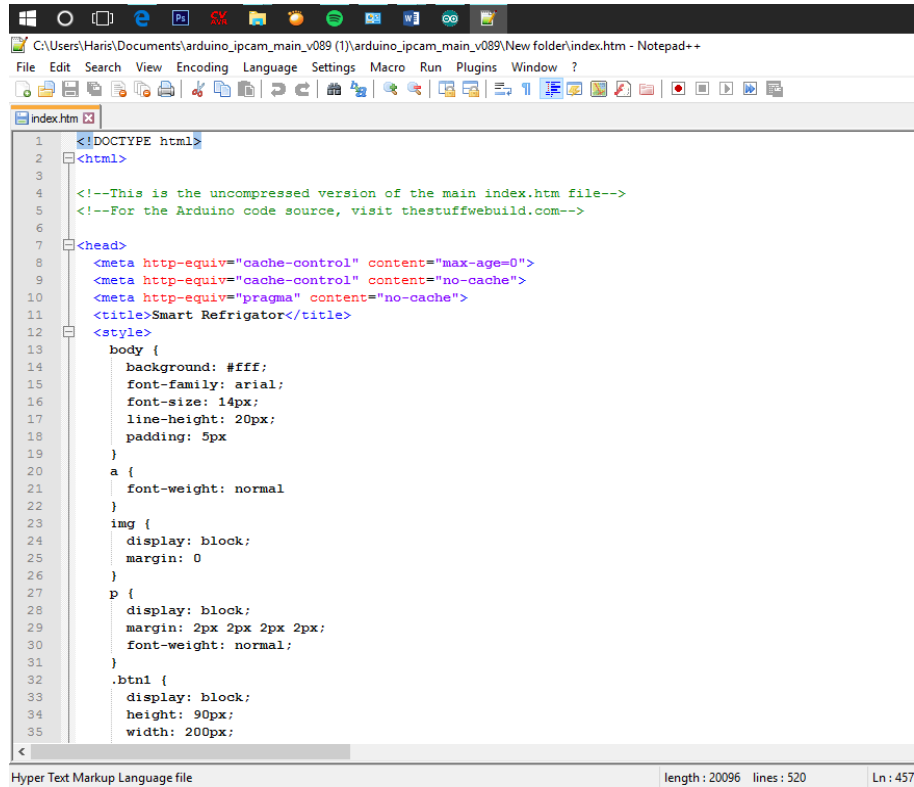
The IDE title bar reads "Smart_Refrigator | Arduino 1.6.5". The bottom status bar indicates "Arduino/Genuino Uno on COM7".

Gambar 28. Pembuatan *Firmware* Arduino

c) Pembuatan HTML

Pembuatan html untuk *web* menggunakan *software* notepad++.

Berikut dibawah ini gambar dari *software* dan pembuatannya.



```

1 <!DOCTYPE html>
2 <html>
3
4 <!--This is the uncompressed version of the main index.htm file-->
5 <!--For the Arduino code source, visit thestuffwebuild.com-->
6
7 <head>
8 <meta http-equiv="cache-control" content="max-age=0">
9 <meta http-equiv="cache-control" content="no-cache">
10 <meta http-equiv="pragma" content="no-cache">
11 <title>Smart Refrigerator</title>
12 <style>
13     body {
14         background: #fff;
15         font-family: arial;
16         font-size: 14px;
17         line-height: 20px;
18         padding: 5px
19     }
20     a {
21         font-weight: normal
22     }
23     img {
24         display: block;
25         margin: 0
26     }
27     p {
28         display: block;
29         margin: 2px 2px 2px 2px;
30         font-weight: normal;
31     }
32     .btn1 {
33         display: block;
34         height: 90px;
35         width: 200px;

```

Hyper Text Markup Language file length : 20096 lines : 520 Ln : 457

Gambar 29. Pembuatan HTML

3. Implementasi dan Pengujian

Pengujian fungsional diperoleh sebelum melakukan uji unjuk kerja karena pengujian fungsional terdiri dari pengujian dari seluruh komponen yang akan dirakit menjadi satu, komponen tersebut terdiri dari hardware dan *software* dan juga memastikan apakah komponen yang digunakan bekerja dengan baik atau tidak.

a. Pengujian Tegangan *Output Power Supply*



Gambar 30. Pengujian tegangan *input* dan *output regulator IC 7805*



Gambar 31. Hasil pengujian tegangan *input* dan *output regulator IC 7805*

Gambar 30 dan Gambar 31 merupakan pengukuran tegangan *output* pada *IC 7805* dan *IC 7812* tanpa beban. Pengujian pada *power supply* sangat diperlukan karena *power supply* merupakan pusat tenaga ataupun sumber tenaga untuk menyuplai seluruh sistem yang ada supaya bekerja dengan maksimal. Pengukuran tegangan dilakukan pada bagian *input* tegangan dan *output* tegangan pada *IC Regulator 7805* dan *IC Regulator 7812*. Hal ini bertujuan untuk mengetahui besarnya tegangan kerja yang masuk sebelum ke perangkat *sismin*, karena *sismin* hanya dapat beroperasi dengan tegangan masukan 5 volt dan 12 volt. Berikut adalah hasil dari pengukuran:

Tabel 8. Pengukuran tegangan 5 V *IC regulator LM7805*

No	Tegangan <i>input</i>	Tegangan <i>output</i> tanpa beban	Tegangan <i>output</i> dengan beban
1	5,00 Volt	4.99 Volt	4.50 Volt
2	5,00 Volt	4.99 Volt	4.50 Volt
3	5,00 Volt	4.99 Volt	4.50 Volt
4	5,00 Volt	4.99 Volt	4.50 Volt
5	5,00 Volt	4.99 Volt	4.50 Volt

Perhitungan *error* tegangan *output* tanpa beban

Nilai sebenarnya = 5.00 V

Nilai terbaca = 4.99 V

$$\text{Error} = \frac{5.00\text{V} - 4.99\text{V}}{5.00\text{V}} = 0.02 \%$$

Perhitungan *error* dengan beban

Nilai sebenarnya = 5.00 V

Nilai terbaca = 4.50 V

$$\text{Error} = \frac{5.0V - 4.9V}{5.0V} = 0.10 \%$$

Tabel 9. Pengukuran tegangan 5 V IC *regulator* LM7805 dengan % *error*

No	Tegangan <i>output</i> tanpa beban	% error Tegangan <i>output</i> tanpa beban	Tegangan <i>output</i> dengan beban	% error Tegangan <i>output</i> dengan beban
1	4.99 Volt	0.02	4.50 Volt	0.10
2	4.99 Volt	0.02	4.50 Volt	0.10
3	4.99 Volt	0.02	4.50 Volt	0.10
4	4.99 Volt	0.02	4.50 Volt	0.10
5	4.99 Volt	0.02	4.50 Volt	0.10

Tabel 10. Pengujian tegangan 12 V IC *regulator* LM7812

No	Tegangan <i>input</i>	Tegangan <i>output</i> tanpa beban	Tegangan <i>output</i> dengan beban
1	12.00 Volt	11.45 Volt	8,80 Volt
2	12.00 Volt	11.45 Volt	8,80 Volt
3	12.00 Volt	11.45 Volt	8,80 Volt
4	12.00 Volt	11.45 Volt	8,80 Volt
5	12.00 Volt	11.45 Volt	8,80 Volt

Perhitungan *error* tegangan *output* tanpa beban

Nilai sebenarnya = 12.00 V

Nilai terbaca = 11.40 V

$$\text{Error} = \frac{12.00V - 11.40V}{12.00V} = 0.05 \%$$

Perhitungan *error* dengan beban

Nilai sebenarnya = 12.00 V

Nilai terbaca = 8.80 V

$$\text{Error} = \frac{12.00\text{V} - 8.80\text{V}}{12.00\text{V}} = 0.27 \%$$

Tabel 11. Pengujian tegangan 12 V IC *regulator* LM7812 dengan % *error*

No	Tegangan <i>output</i> tanpa beban	% <i>error</i> Tegangan <i>output</i> tanpa beban	Tegangan <i>output</i> dengan beban	% <i>error</i> Tegangan <i>output</i> dengan beban
1	11.45 Volt	0.05	8,80 Volt	0.27
2	11.45 Volt	0.05	8,80Volt	0.27
3	11.45 Volt	0.05	8,80 Volt	0.27
4	11.45 Volt	0.05	8,80 Volt	0.27
5	11.45 Volt	0.05	8,80 Volt	0.27

Berdasarkan hasil data hasil dari pengukuran pada tabel 8 sampai tabel 11, bisa dilihat bahwa *power supply* bekerja dengan baik karena tegangan keluaran tanpa beban dan dengan beban yang terukur tetap stabil.

b. Pengujian Kamera

Pengujian kamera dilakukan untuk mengetahui gambar yang dihasilkan oleh kamera dari resousi yang berbeda-beda. *Module* kamera VC0706 mempunyai tiga resolusi atau ukuran gambar yang berbeda-beda, yaitu: 1) resolusi 640x480 piksel, 2) resolusi 320x240 piksel, dan

3) 160x120 piksel. Berikut dibawah ini adalah gambar-gambar yang dihasilkan.



Gambar 32. Resolusi 160x120 piksel



Gambar 33. Resolusi 320x240 piksel



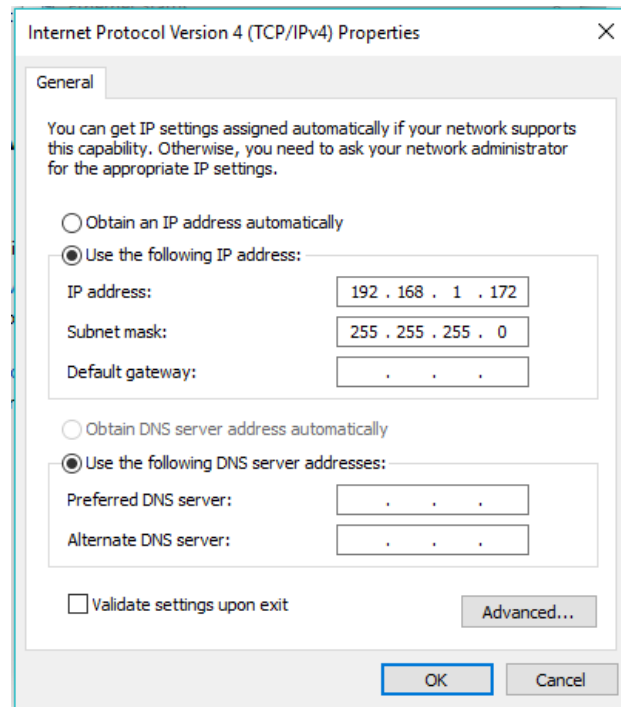
Gambar 34. Resolusi 640x480 piksel

Terlihat perbedaan dari ketiga gambar diatas berarti resolusi yang paling besar menampilkan gambar yang paling baik. Untuk itu, pada penelitian ini peneliti menggunakan tiga resolusi, sehingga dapat memilih resolusi yang paling besar supaya gambar dapat dilihat dengan baik.

c. Pengujian Ethernet Shield R3

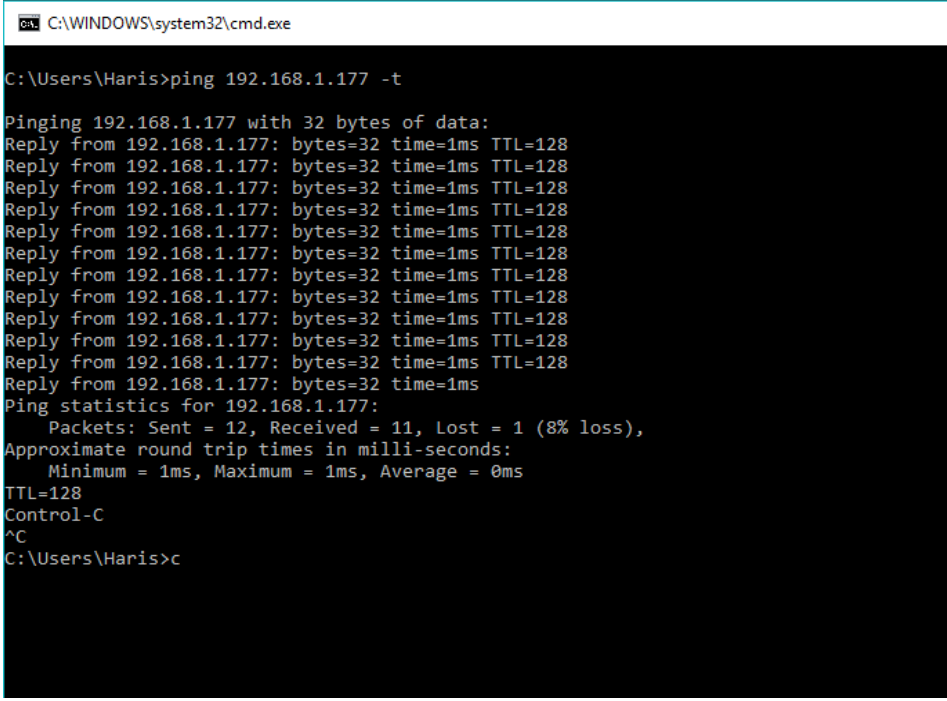
Pengujian rangkaian ini dengan menghubungkan alat ke komputer. Hubungan modul dengan komputer adalah jenis *peer to peer* sehingga sambungan yang digunakan adalah *cross*. Kabel yang digunakan jenis utp dengan konektor RJ45. Setelah tersambung dengan komputer langkah selanjutnya adalah dengan mengatur IP komputer disesuaikan dengan IP modul ethernet. IP modul ethernet menggunakan IP 192.168.1.177 sehingga komputer kita set dalam range IP modul ethernet yaitu 192.168.1.172

sampai dengan 192.168.1.254, kecuali ip 192.168.1.177 karena sudah dipakai oleh modul ethernet. Tampilan pengaturan IP komputer dapat dilihat pada gambar 35.



Gambar 35. Pengaturan IP pada komputer

Pengecekan koneksi antara modul dan komputer dengan menggunakan program DOS yaitu dengan instruksi PING, hasil pengecekan komunikasi antara modul IP dapat dilihat pada gambar 36. Dari hasil pengujian dapat dilihat komunikasi antara modul ethernet dan komputer bagus, dengan ping time < 10 ms.



```
C:\WINDOWS\system32\cmd.exe
C:\Users\Haris>ping 192.168.1.177 -t
Pinging 192.168.1.177 with 32 bytes of data:
Reply from 192.168.1.177: bytes=32 time=1ms TTL=128
Reply from 192.168.1.177: bytes=32 time=1ms TTL=128
Reply from 192.168.1.177: bytes=32 time=1ms TTL=128
Reply from 192.168.1.177: bytes=32 time=1ms TTL=128
Reply from 192.168.1.177: bytes=32 time=1ms TTL=128
Reply from 192.168.1.177: bytes=32 time=1ms TTL=128
Reply from 192.168.1.177: bytes=32 time=1ms TTL=128
Reply from 192.168.1.177: bytes=32 time=1ms TTL=128
Reply from 192.168.1.177: bytes=32 time=1ms TTL=128
Reply from 192.168.1.177: bytes=32 time=1ms TTL=128
Reply from 192.168.1.177: bytes=32 time=1ms TTL=128
Reply from 192.168.1.177: bytes=32 time=1ms
Ping statistics for 192.168.1.177:
    Packets: Sent = 12, Received = 11, Lost = 1 (8% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 0ms
TTL=128
Control-C
^C
C:\Users\Haris>c
```

Gambar 36. Pengecekan Komunikasi dengan PING

Pembuatan Jaringan wifi menggunakan *software* connectivity. Connectivity adalah *software* yang digunakan untuk mempermudah membuat koneksi wifi di komputer yang hanya dapat digunakan untuk windows OS. Pengujian menggunakan *software* Connectivity dilakukan beberapa jarak yang berbeda. Berikut hasil pengujian dapat dilihat pada tabel 10.

Tabel 12. Uji daya jangkauan *connectivity* dari Ethernet *shield* R3

No	Jarak <i>Connectivity</i> dan Penerima (meter)	Keterangan (signal streng)
1	3 m tanpa penghalang	Penuh 4 strip
2	4 m terhalang 1 dinding	3 strip
3	8 m terhalang 2 dinding	2 strip
4	15 m terhalang 3 dinding	1 strip
5	15 m tanpa penghalangan	Penuh 4 strip

B. Hasil Pengujian Pemantau isi kulkas

Pengujian unjuk kerja secara keseluruhan bertujuan untuk mengetahui pemantau isi kulkas menggunakan Ethernet *shield* R3 berbasis Arduino uno R3 secara elektriks bekerja dengan baik dan benar. Pengujian ini dilakukan dengan dua alat yang berbeda yaitu dengan *smartphone* dan web. Berikut di bawah ini adalah gambar untuk pengujian menggunakan *smartphone* dengan tiga resolusi yang berbeda.

a) Hasil pengujian menggunakan *smartphone* android

Hasil pengujian menggunakan *smartphone* dapat berfungsi dengan baik, *power supply* yang masuk dalam rangkaian mikrokontroler dapat bekerja sesuai perencanaan sehingga dapat menghasilkan gambar atau photo dari jarak maksimal 15 m. tanpa penghalang. Berikut hasil pengujian menggunakan aplikasi android dapat dilihat pada gambar 37, 38, dan 39.



Gambar 37. Pengujian Menggunakan Aplikasi *Smartphone* Android dengan Resolusi 160x120 piksel



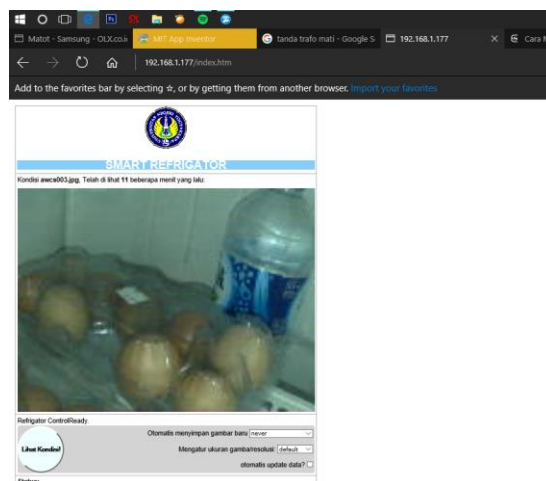
Gambar 38. Pengujian Menggunakan Aplikasi *Smartphone* Android dengan Resolusi 320x240 piksel



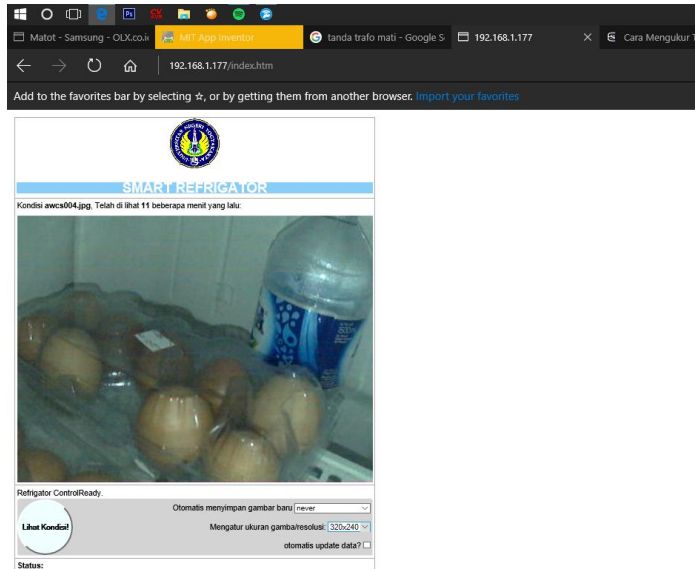
Gambar 39. Pengujian Menggunakan Aplikasi *Smartphone* Android dengan Resolusi 640x480 piksel

b) Hasil pengujian menggunakan web *browser*

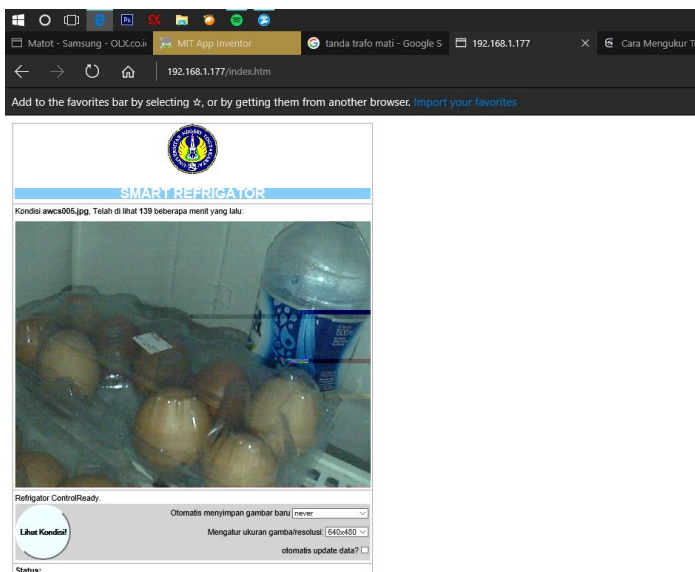
Berikut hasil pengujian menggunakan web *browser* dapat dilihat pada gambar 40, 41 dan 42 dibawah ini.



Gambar 40. Pengujian Menggunakan web *browser* dengan Resolusi 160x120 piksel



Gambar 41. Pengujian Menggunakan web *browser* dengan Resolusi 320x240 piksel



Gambar 42. Pengujian Menggunakan web *browser* dengan Resolusi 640x480 piksel

C. Pembahasan

1. Perangkat Keras (*Hardware*)

Berdasarkan tahapan pengujian yang telah dilakukan maka dapat ditarik kesimpulan bahwa sistem yang telah dirancang dapat bekerja sebagaimana mestinya, meskipun terdapat *error* atau kesalahan pada beberapa bagian. Berikut adalah pembahasan dari pengujian yang telah dilakukan pada:

a) Catu daya

Hasil pengukuran catu daya bekerja dengan baik dan telah dibuktikan pada Tabel 8 sampai Tabel 11. Tegangan masukan terukur sebanyak lima kali mengalami penurunan tegangan *input* dari 12 Volt sampai 11.45 Volt dan 5 Volt sampai 4,5 Volt, disebabkan karena konsumsi rangkaian. Tegangan *output* sudah sesuai yang diharapkan dan memenuhi tegangan input mikrokontroler arduino uno yaitu sebesar 7 Volt sampai 12 Volt dan tegangan kerja dari 5 Volt sampai 4,5 Volt dengan beban. Jika catu daya kurang dari tegangan input yang telah ditentukan maka rangkaian tidak bisa berjalan dengan normal atau rangkaian terjadi *error*.

b) Kamera

Module camera VC0706 dapat mengambil photo cukup baik dan gambar dapat dilihat secara jelas. Hasil pengujian pada Gambar 32, 33, dan 34 menunjukkan bahwa kamera dapat mengambil gambar dengan tiga resolusi atau ukuran gambar yang berbeda, meskipun ketiga gambar tersebut mempunyai gambar dengan ukuran yang

berbeda, tetapi masih bisa dilihat dengan jelas pada ukuran yang paling besar yaitu: 640x480 piksel. Penempatan kamera dalam bentuk statis. Jika kamera tidak statis, maka isi kulkas tidak dapat terlihat seluruhnya, kecuali kamera yang dipakai adalah kamera dengan ukuran kecil dan pergerakannya tidak memakai banyak ruang.

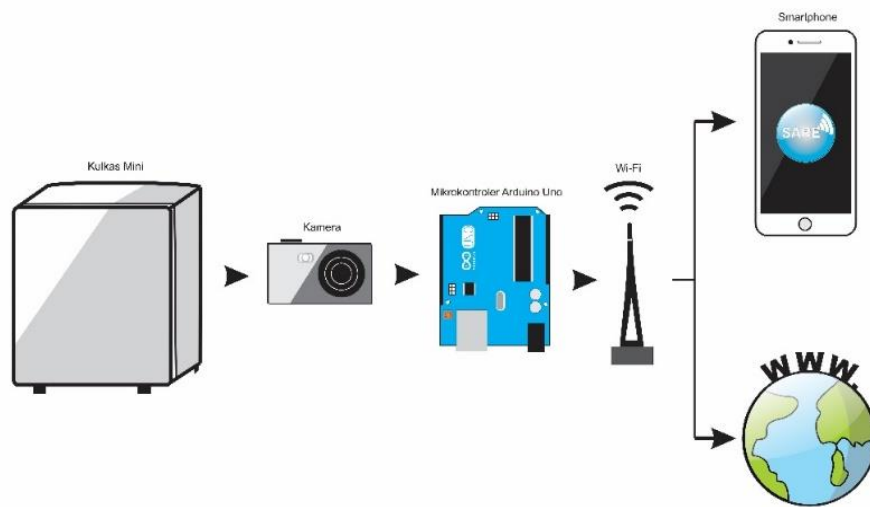
c) **Ethernet Shield R3**

Jaringan internet dibuat menggunakan Ethernet *shield* R3 dapat bekerja dengan baik dan sesuai dengan target. Ethernet dapat digunakan untuk membagi jaringan wifi melalui *software Connectivity* kemudian *smartphone* dapat menerima jaringan wifi tersebut. Tetapi ada batasan jarak yang harus dipenuhi agar selalu terkoneksi dengan wifi karena masih menggunakan IP local, jadi maksimal jarak yang bisa terkoneksi wifi hanya 15 meter. Selain itu penghalang juga mempengaruhi kinerja pada wifi seperti tembok, gedung, dan tinggi rendahnya suatu daratan. Pengujian yang dilakukan menggunakan penghalang tembok. Jarak antara penerima dan pengirim dapat dilihat pada tabel 10. Kemungkinan solusi agar dapat diakses keseluruhan wilayah yaitu menggunakan IP Public dengan begitu pemantau isi kulkas dapat diakses keseluruhan wilayah.

2. Perangkat Lunak (Software)

Berdasarkan tahapan pengujian yang telah dilakukan pada *software* yang telah dibuat untuk *smartphone* dan web dapat ditarik kesimpulan bahwa *software* dapat mengambil gambar dengan tiga resolusi yang berbeda dari dalam kulkas dan bekerja dengan baik dan benar.

D. Mekanisme Pengolahan Informasi Pemantau isi kulkas



Gambar 43. Cara Kerja Alat pada Gambar Sebenarnya

Gambar di atas menjelaskan bahwa *module camera VC0706* mengambil gambar yang ada didalam kulkas dan diolah dengan mikrokontroler Arduino uno R3 kemudian dikirim melalui Ethernet shield R3 yang dapat menghubungkan Jaringan internet atau wifi agar dapat diterima dan melihat kondisi kulkas dengan *smartphone* dan web.

“Pemantau isi kulkas Menggunakan Ethernet Shield R3 dan Arduino Uno R3” memiliki keterbatasan yaitu tidak bisa mengambil gambar pada seluruh permukaan isi kulkas karena kulkas yang berukuran kecil. Selain itu kamera juga tidak bisa *moving* karena ukuran kamera yang terlalu besar dan jika jarak terlalu dekat maka gambar akan terlihat sangat dekat dan isi kulkas tidak bisa terlihat seluruhnya. Ethernet yang digunakan masih menggunakan *IP local*, jadi tidak bisa diakses melebihi jarak yang ada pada pengujian.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan pengujian yang telah dilakukan terhadap “Pemantau isi kulkas Menggunakan Ethernet Shield R3 dan Arduino Uno R3” maka dapat disimpulkan:

1. Perangkat Keras “Pemantau isi kulkas Menggunakan Ethernet Shield R3 dan Arduino Uno R3” dibangun terdiri dari beberapa rangkaian yang saling mendukung. Rangkaian pendukung alat ini adalah catu daya, mikrokontroler Arduino, Ethernet shield R3, *module camera VC0706*, dan LED,
2. perangkat lunak yang digunakan untuk merealisasikan alat ini adalah *software* Arduino untuk memprogram sistem minimum. Proteus7 digunakan untuk membuat *layout* PCB Arduino. Corel draw X7 untuk *men-design box*. Notepad++ untuk membuat html pada web dan App Inventor untuk membuat aplikasi android. Secara keseluruhan program yang dibuat sudah dapat bekerja sesuai dengan tujuan. Hal ini ditunjukkan dengan berhasilnya Arduino melakukan pengambilan gambar pada isi kulkas,
3. pemantau isi kulkas dirancang untuk memantau isi kulkas ketika pemilik kulkas berada di luar rumah agar dapat mempermudah kegiatannya dengan jarak yang ditentukan. Jarak tersebut bisa dilihat melalui tabel 10 dengan maksimal jarak yang ditempuh 15 meter tanpa penghalang.

B. Keterbatasan Alat

1. Pemantau isi kulkas tidak ada komponen tambahan berupa sensor yang dapat mendeteksi kelayakan konsumsi makanan untuk manusia agar dapat mengetahui apakah makanan yang ada di dalam kulkas harus masih baik atau tidak,
2. keterbatasan jarak koneksi pada jaringan wi-fi yaitu jarak maksimal 15 meter,
3. kamera dalam keadaan statis, mengingat dimensi kamera yang besar dan tidak bisa mengambil gambar pada setiap sudut isi kulkas.

C. Saran

Berdasarkan hasil penelitian, penulis mengakui terdapat kekurangan terhadap alat yang dibuat ini, karena keterbatasan kemampuan dan waktu, maka penulis mengajukan saran sebagai berikut:

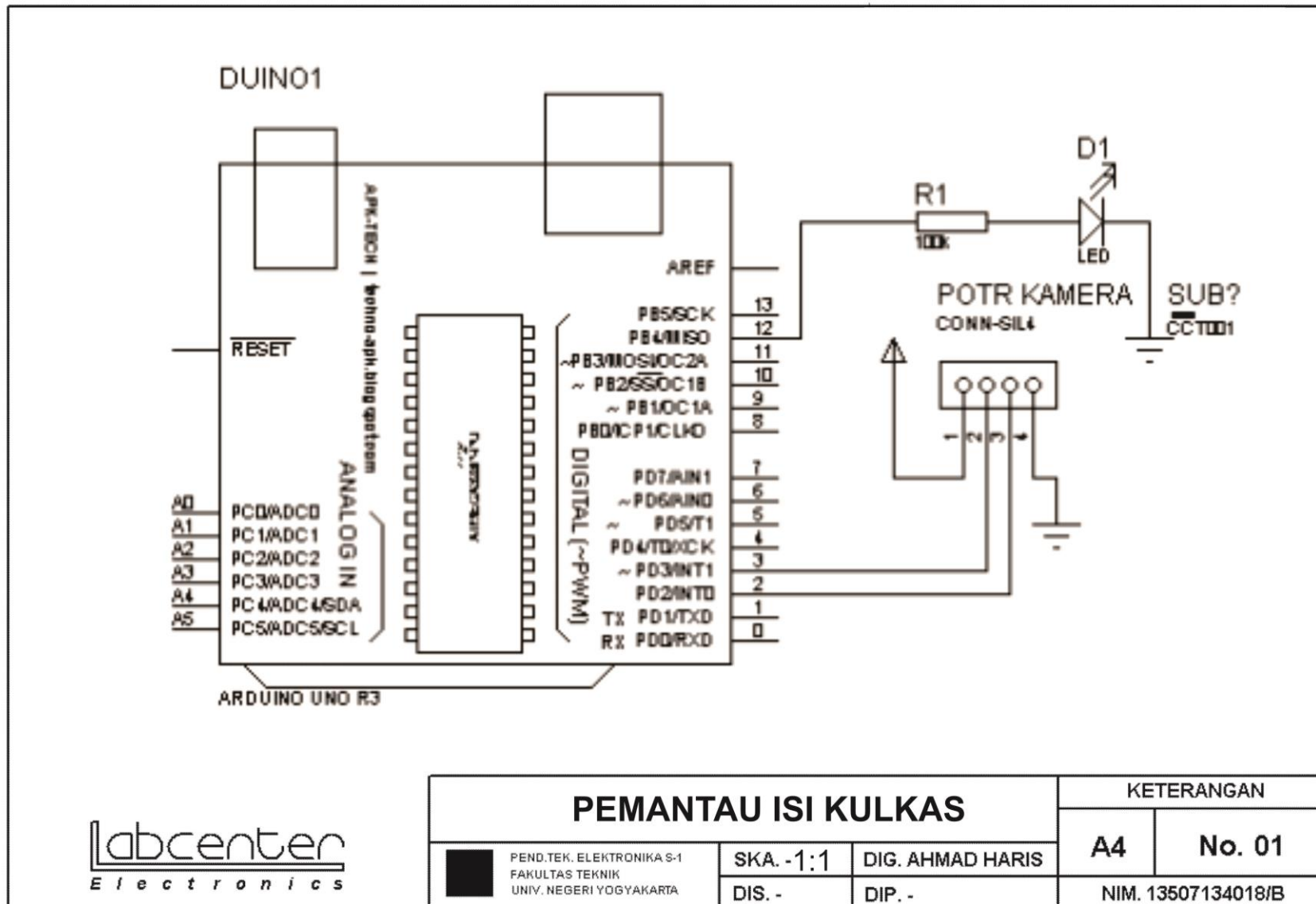
1. Pemantau isi kulkas ini akan lebih baik jika ada komponen tambahan berupa sensor yang dapat mendeteksi kelayakan konsumsi makanan untuk manusia agar dapat mengetahui apakah makanan yang ada di dalam kulkas harus masih baik atau tidak,
2. sebaiknya menggunakan kamera yang lebih kecil dilengkapi dengan penggerak agar dapat *moving* dan bisa melihat seluruh isi kulkas,
3. kulkas dapat diakses seluruh wilayah jika menggunakan IP *public*.

DAFTAR PUSTAKA

- Aingandra (2009). *Android Adalah – Pengertian Android – Sistem Operasi*. Diambil tanggal 25 Mei 2016, dari, <http://www.aingindra.com/android-adalah-pengertian-android-sistem-operasi.html>.
- Ecadio (2016). *Mengenal Arduino uno R3*. Diambil pada tanggal 11 Mei 2016, dari, <http://ecadio.com/mengenal-dan-belajar-arduino-uno-r3>.
- H, Mike (2015). *Arduino Ajax Ip Webcam*. Diambil pada tanggal 6 September 2016, dari, <http://thestuffwebuild.com/projects/small-things/arduino-ajax-ip-webcam/>.
- M, Edward (2014). *Display “Warning” and “Alert” box messages in App Inventor apps*. Diambil pada tanggal 1 September 2016, dari, <http://appinventor.pest.com/?p=81>.
- Igendel (2015). *Arduino And The Vc0706 Jpeg Camera*. Diambil pada tanggal 15 juni 2016, dari, <http://www.idogendel.com/en/archives/385>.
- Managam, Erick (2012). *E-Commerce*. Diambil pada tanggal 10 september 2016, dari, <http://erickmanagam.blogspot.co.id/2012/06/telepon-pintar-smartphone-adalah.html>.
- Taifun (2016). *App Inventor and Multiple Screens*. Diambil pada tanggal 5 September 2016, dari, <http://puravidaapps.com/manager.php>

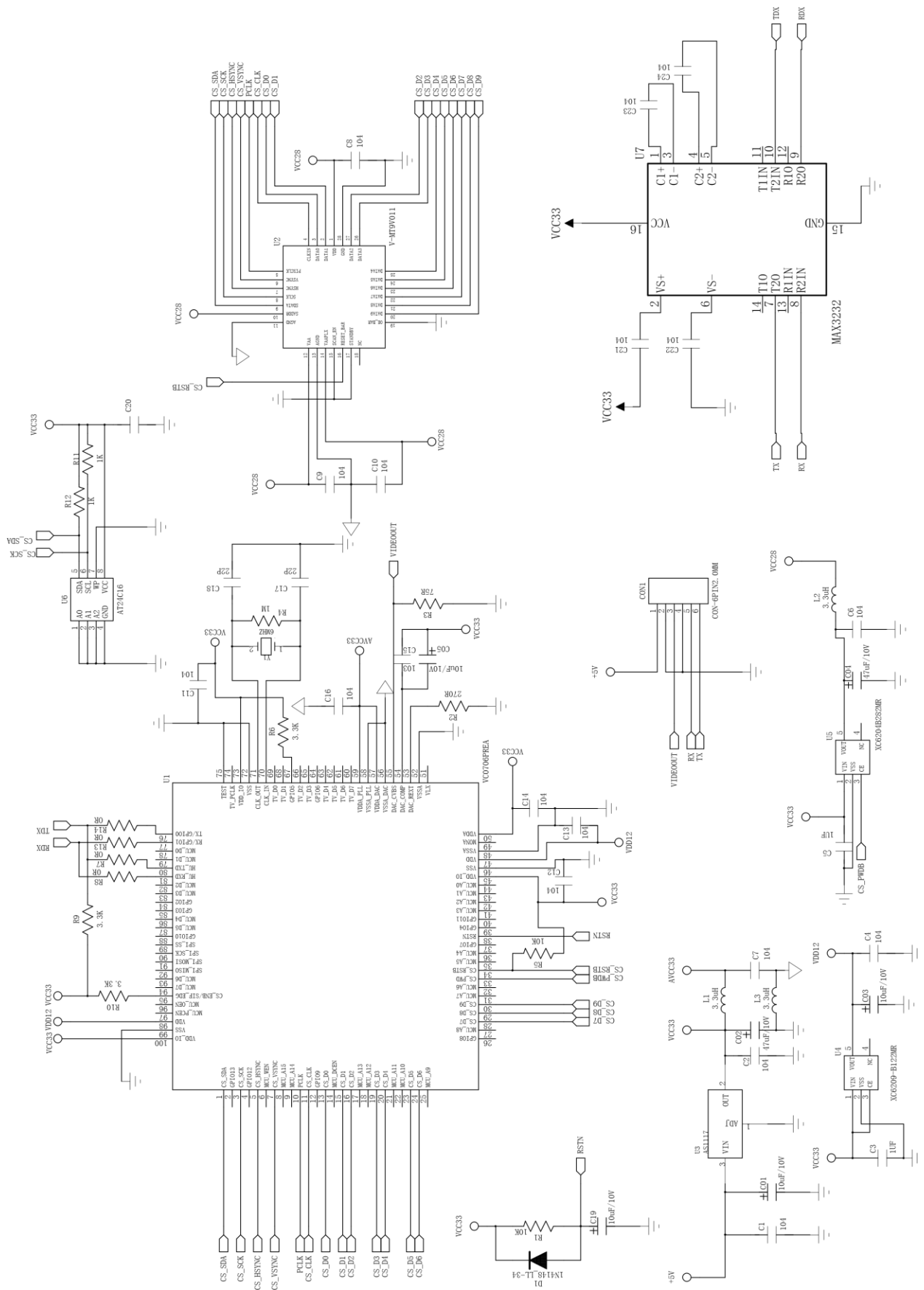
LAMPIRAN

Lampiran 1. Design Rangkaian Alat

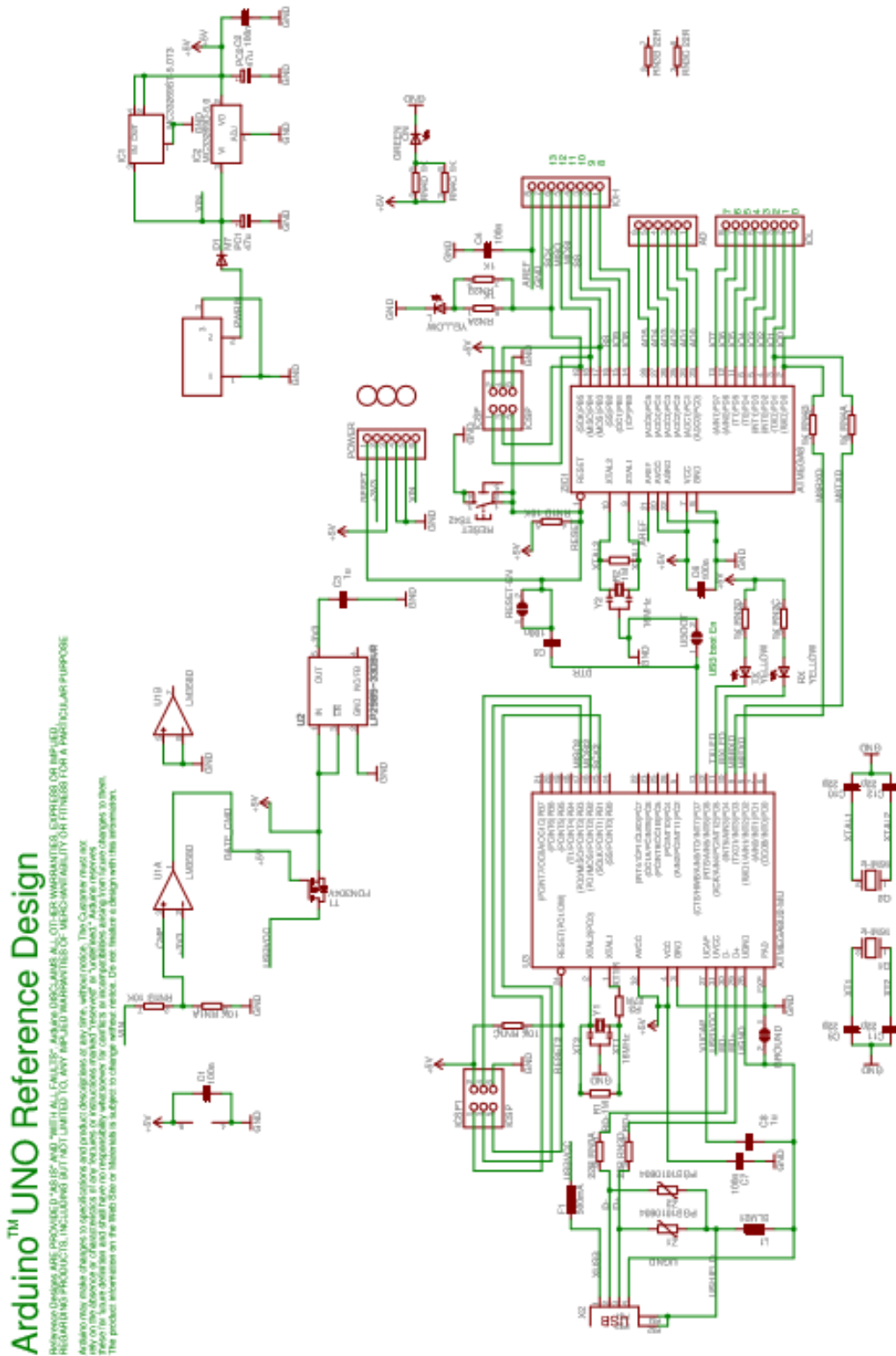


PEMANTAU ISI KULKAS		KETERANGAN	
		PEND. TEK. ELEKTRONIKA S-1 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA. -1:1 DIS. -

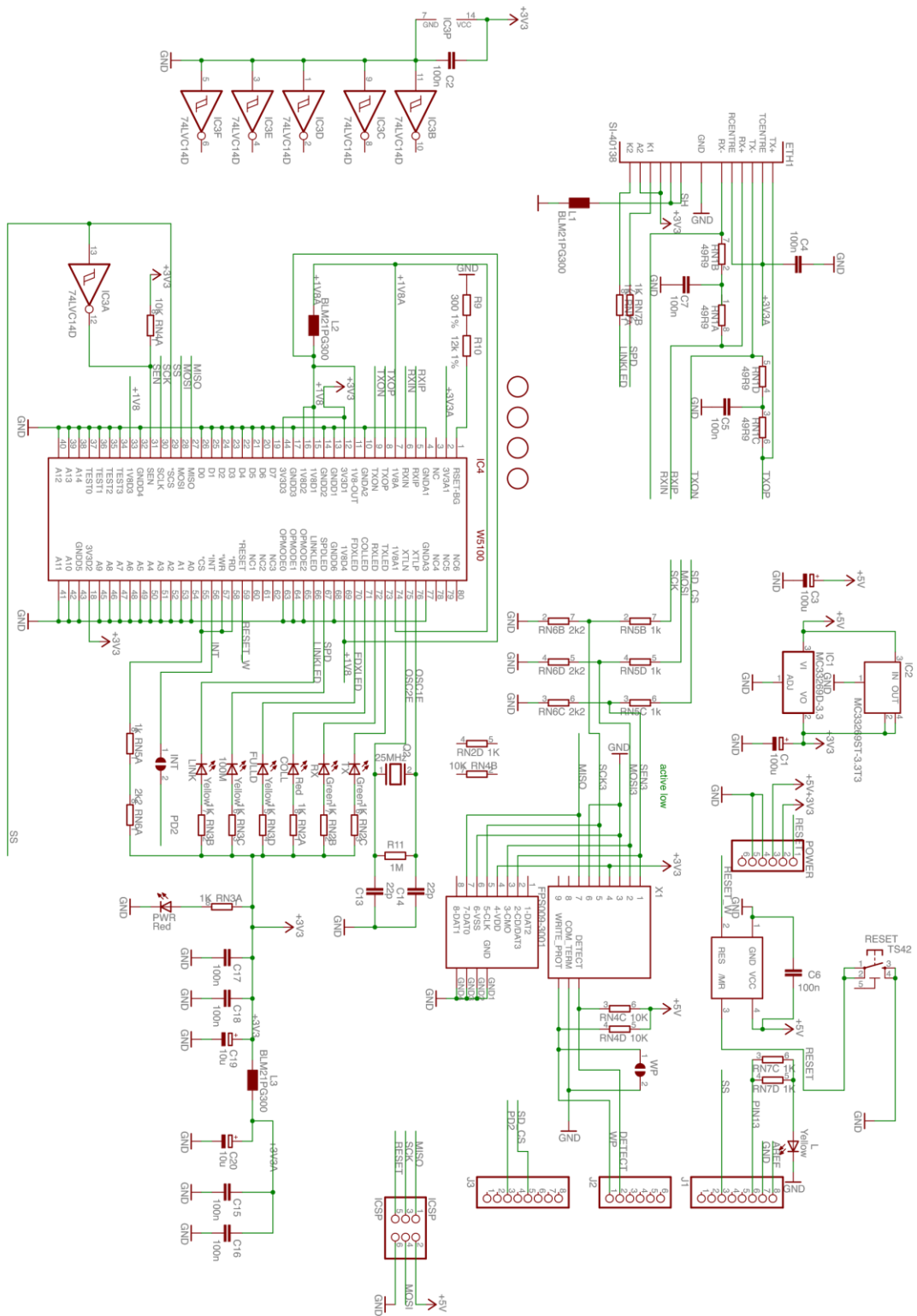
Lampiran 2. Gambar Rangkaian Module Kamera VC0706



Lampiran 3. Gambar Rangkaian Arduino Uno R3



Lampiran 4. Gambar Rangkaian Ethernet Shield R3



Lampiran 5. Source Code Arduino Uno R3 pada Pemantau isi kulkas

```
#define DebugLevel 2 //(1)+sd,client requests,ambil gambar.
#define BusyLED 13
#include <Adafruit_VC0706.h>
#include <SoftwareSerial.h>
#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>

SoftwareSerial cameraconnection = SoftwareSerial(2, 3);
Adafruit_VC0706 cam = Adafruit_VC0706(&cameraconnection);

//memulai vars penamaan file dari 0
int fname = 0;

//vars untuk menghitung waktu and time since last pic (tslp)
unsigned long lastpic = 0;
unsigned long tslp = 0;

//memuali countour hits dari 0
int hits = 0;

//menentukan beban pada cpu
unsigned long preload = 0;
int load = 0;

//maksimal gambar
int maxpics = 256;
```



```
pinMode(BusyLED, OUTPUT);
Ethernet.begin(mac, ip);
server.begin();
Serial.begin(9600);

if (!SD.begin(4)) {
#if DebugLevel >= 1
    Serial.print(F("sd failed! "));
#endif
}
else{
#if DebugLevel >= 2
    Serial.print(F("sd ready. "));
#endif
}

Serial.println(F("server running. "));

kamera_test();

//take a pic to get things primed
Kamera();
}

void loop() {
    int BUFSIZ = 32;
    int index = 0;
    char clientline[BUFSIZ];
    EthernetClient client = server.available();
    if (client) {
```

```
boolean currentLineIsBlank = true;
index = 0;

while (client.connected()) {

    if (client.available()) {

        char c = client.read();
        if (c != '\n' && c != '\r') {
            clientline[index] = c;
            index++;
            if (index >= BUFSIZ)
                index = BUFSIZ - 1;
            continue;
        }
        clientline[index] = 0;

        wfilename = processFile(clientline);

#ifdef DebugLevel >= 2
        Serial.print(F("Request: "));
        Serial.println(wfilename);
#endif

        if (strstr(clientline, "GET /?xml") != 0) {
            dataReq(client);
            break;
        }
        if (strstr(clientline, "GET /?pic1") != 0) {
```

```
    cammode = 1;
    code200(client);
    kamera_test();
    Kamera();
    client.print(F("done.));
    break;
}
if (strstr(clientline, "GET /?pic2") != 0) {
    cammode = 2;
    code200(client);
    kamera_test();
    Kamera();
    client.print(F("done.));
    break;
}

//AJAX request ambil gambar
if (strstr(clientline, "GET /?pic3") != 0) {
    cammode = 3;
    code200(client);
    kamera_test();
    Kamera();
    client.print(F("done.));
    break;
}
if (strstr(clientline, "GET /?dir") != 0) {
    code200(client);
    printDirectory(client, 0);
    break;
}
```

```
}

if (strlen(wfilename) < 2) {
    wfilename = rootFileName;
}

if (SD.exists(wfilename)) {
    code200(client);
    digitalWrite(BusyLED, HIGH);

    webFile = SD.open(wfilename, FILE_READ);
    if (webFile) {
        while(webFile.available()) {
            client.write(webFile.read());
        }
        delay(400);
        webFile.close();
    }
    digitalWrite(BusyLED, LOW);
    break;
}

else {
    code200(client);
    client.print(F("error"));
    break;
}

break;
}
}
```

```
//close kamera
client.println(F("<!--end-->"));
delay(1);
client.flush();
client.stop();
}

load = millis() - preload;
preload = millis();

}

char* processFile(char clientline[63]) {
  char *wfilename;
  wfilename = clientline + 5;
  (strstr(clientline, " HTTP"))[0] = 0;
  return wfilename;
}

void Kamera(){
  if (! cam.takePicture()){
    #if DebugLevel >=1
      Serial.println(F("Image failed"));
    #endif
  }
  else {
    #if DebugLevel >=3
      Serial.println(F("Image taken"));
    #endif
  }
}
```



```

#endif
}

// a filename
strcpy(pfilename, "awcs000.jpg");
pfilename[4] = (fname/100)%10 + '0'; //To get 1st digit from fname
pfilename[5] = (fname/10)%10 + '0'; //To get 2nd digit from fname
pfilename[6] = fname%10 + '0'; //To get 3rd digit from fname
if (SD.exists(pfilename)) {
    SD.remove(pfilename);
}
fname++;
if (fname >= maxpics){
    fname=0;
}
digitalWrite(BusyLED, HIGH);
File imgFile = SD.open(pfilename, FILE_WRITE);
uint16_t jpglen = cam.frameLength();
#if DebugLevel >=2
    Serial.print(jpglen, DEC);
    Serial.print(F(" bytes going to "));
    Serial.print(pfilename);
#endif
while (jpglen > 0) {
    // read 32 bytes at a time;
    uint8_t *buffer;
    uint8_t bytesToRead = min(32, jpglen);
    buffer = cam.readPicture(bytesToRead);
    imgFile.write(buffer, bytesToRead);
}

```

```
    jpglen -= bytesToRead;
}
imgFile.close();

//note the time for xml data
lastpic = millis();

cam.resumeVideo();

#if DebugLevel >=2
Serial.println(F("..done"));
#endif

digitalWrite(BusyLED, LOW);
}

void code200(EthernetClient client) {
    client.println(F("HTTP/1.1 200 OK"));
    if (strstr(wfilename, ".jpg") != 0) client.println(F("Content-Type: image/jpeg"));
    else if (strstr(wfilename, ".txt") != 0) client.println(F("Content-Type:
text/plain"));
    else {
        client.println(F("Content-Type: text/html"));
        hits++;
    }
    client.println(F("Connection: close"));
    client.println(F("Cache-Control: no-cache, max-age=0"));
    client.println();
}

void kamera_test() {
```

```
    if (cam.begin()) {
#if DebugLevel >= 3
        Serial.print(F("Cam on. "));
#endif
    }
    else {
#if DebugLevel >= 1
        Serial.print(F("Cam err. "));
#endif
    }
    delay(500);

    if(cammode == 3){
        cam.setImageSize(VC0706_640x480);    // large
    }
    else if(cammode == 2){
        cam.setImageSize(VC0706_320x240);    // med
    }
    else{
        cam.setImageSize(VC0706_160x120);    // small
    }
    cam.begin();
    for(byte i=0;i<5;i++){
        cam.getImageSize();
    }
    uint8_t imgsize = cam.getImageSize();

#if DebugLevel >= 3
```

```
Serial.print("Img size:");
if (imgsize == VC0706_640x480) Serial.println(F("640x480"));
else if (imgsize == VC0706_320x240) Serial.println(F("320x240"));
else if (imgsize == VC0706_160x120) Serial.println(F("160x120"));
else Serial.println(F("err"));
#endif

delay(500);

}

void printDirectory(EthernetClient client, int numTabs) {

File dir = SD.open("");
dir.rewindDirectory();

while(true){
  entry = dir.openNextFile();
  if (! entry) {

    entry.close();
    break;
  }

String fname = entry.name();
fname.toLowerCase();

if (entry.isDirectory()) {
  //nothing
}
```

```
else{
    client.print(F("<a href=\"");
    client.print(fname);
    client.print(F("\>"));
    client.print(fname);
    client.print(F(" &nbsp; "));
    client.print(entry.size(), DEC);
    client.print(F(" b</a><br />"));
}
entry.close();
}
dir.close();
}

void dataReq(EthernetClient client) {

    client.println(F("HTTP/1.1 200 OK"));
    client.println(F("Content-Type: text/xml"));
    client.println(F("Connection: close"));
    client.println(F("Cache-Control: no-cache, max-age=0"));
    client.println();

    client.print(F("<?xml version = \"1.0\" ?>"));
    client.print(F("<xml>"));

    client.print(F("<id>"));
    client.print(pfilename);
    client.print(F("</id>"));

    client.print(F("<age>"));
```

```
    tslp = (millis() - lastpic) / 1000;
    client.print(tslp);
    client.print(F("</age>"));

    client.print(F("<mp>"));
    client.print(maxpics);
    client.print(F("</mp>"));

    client.print(F("<hit>"));
    client.print(hits);
    client.print(F("</hit>"));

    client.print(F("<up>"));
    client.print(millis());
    client.print(F("</up>"));
    client.print(F("<size>"));
    uint8_t imgsize = cam.getImageSize();
    if (imgsize == VC0706_640x480) client.print(F("640x480"));
    else if (imgsize == VC0706_320x240) client.print(F("320x240"));
    else if (imgsize == VC0706_160x120) client.print(F("160x120"));
    else client.print(F("err"));
    client.print(F("</size>"));
    client.print(F("<lo>"));
    if (load < 50) client.print(F("berhenti"));
    else client.print(F("high"));
    //Serial.print(load);
    client.print(F("</lo>"));
    client.print(F("</xml>"));
}
```