

**PENERAPAN DESAIN INTERAKSI PADA *REMOTE CONTROL*
UNTUK SMART HOME**

SKRIPSI

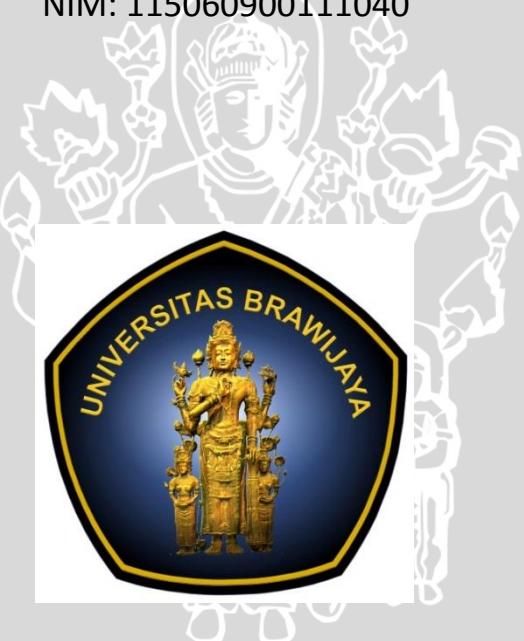
KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Hadistya Bagus A.

NIM: 115060900111040



PROGRAM STUDI TEKNIK INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

PENERAPAN DESAIN INTERAKSI PADA *REMOTE CONTROL UNTUK SMART HOME*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Hadistya Bagus A.
NIM: 115060900111040

Skripsi ini telah diuji dan dinyatakan lulus pada
21 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Hurriyatul Fitriyah, S.T., M.Sc.

NIK. 2013048510012001

Denny Sagita R., S.Kom., M.Kom.

NIP. 19851124015041001

Mengetahui
Ketua Program Studi Teknik Informatika

Drs. Marji, M.T.

NIP. 1970801 199203 1 001



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 21 Januari 2016

Hadistya Bagus A.

NIM: 115060900111040



KATA PENGANTAR

Puji syukur atas kehadiran Allah SWT yang melimpahkan rahmat dan hidayah sehingga saya dapat menyelesaikan skripsi. Skripsi ini adalah salah satu syarat untuk mencapai gelar Sarjana Komputer di Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya yang berjudul "PENERAPAN DESAIN INTERAKSI PADA REMOTE CONTROL UNTUK SMART HOME".

Skripsi ini yang telah disusun tidak lepas dari pihak-pihak yang telah membantu dalam bentuk do'a, kritik, saran, pendapat, dan semangat. Untuk itu saya menyampaikan ucapan terima kasih kepada:

1. Kedua orang tua saya yang telah memberikan dukungan semangat serta doa
2. Ibu Hurruyatul Fitriyah, S.T., M.Sc. sebagai pembimbing I yang telah memberikan saran, bantuan , bimbingan dalam penyelesaian skripsi ini
3. Bapak Denny Sagita R. S.kom, M.kom. sebagai pembimbing II yang telah memberikan banyak saran, kritik, ilmu dan bimbingan dalam penyelesaian skripsi ini
4. Bapak Adharul Muttaqin S.T., M.T. selaku ketua program studi Sistem Komputer Program Teknologi Informasi dan Ilmu Komputer.
5. Segenap dosen Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya yang telah memberikan ilmu pengetahuan, serta staff dan pegawai Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya atas segala bantuan yang bersifat administrative.
6. Mbak put, mbak dhen, mbak sus yang telah menyemangati saya dalam pengerjaan skripsi
7. Brian Angga Kusuma, Rizal Hariansah, Jenar Ramasongko, Ragil Pamungkas, Oferia Chacha, Nurma Febriana, lakmana berca ula dan teman-teman lain yang tidak dapat di sebutkan satu persatu
8. Mas Bhekti, om dar, om wis, brian, om rhama dan Peri-malang yang telah menyupport saya dalam pengerjaan skripsi
9. Teman-teman TKB48 yang telah berjuang bersama-sama mengerjakan skripsi
10. Serta semua pihak yang telah membantu dan memberikan pengalaman bagi saya selama menjalani masa perkuliahan.



Pada akhirnya, saya menyadari bahwa skripsi ini masih belum mencapai kata sempurna. Oleh karena itu saya mengharapkan masukkan berupa kritik dan saran. Saya berharap semoga skripsi saya dapat berguna bagi pengembangan ilmu pengetahuan dan teknologi.

Malang, 21 Januari 2016

Penulis

hadisbagus@gmail.com



ABSTRAK

Smart Home adalah suatu sistem pengendalian dan monitoring rumah yang memiliki sensor, aktuator, display, dan elemen komputer didalamnya. Sistem-sistem yang tertanam tersebut terintegrasi dalam suatu jaringan dan dapat saling berkomunikasi satu sama lainnya. Rumah atau tempat tinggal dengan system otomatis ataupun *remote* yang mengontrol komponen merupakan bentuk dari smart home. Beberapa komponen smart (pintar) membutuhkan sistem yang tertanam dan terintegrasi dalam suatu jaringan dan dapat saling berkomunikasi satu sama lainnya.

Penelitian ini ditujukan untuk membuat desain *remote kontrol smart home* yang lebih interaktif dengan menggunakan *remote* sebagai kontrol dari smart home. *remote control* smart home didesain berdasarkan memenuhi prinsip desain interaksi. hasil dari pengujian verifikasi menyatakan bahwa *remote control* bekerja sesuai dengan perancangannya.

Pada pengujian observasi *remote control smart home*, didapatkan tingkat keberhasilan user dalam menjalankan sekenario sebesar 100% dan membutuhkan waktu rata-rata 9,2 detik. analisis hasil pengujian usabilitas menggunakan questioner diberikan kepada 30 responden, 15 responden dengan dasar ilmu IT dan 15 responden dengan dasar ilmu non-IT dengan metode penilaian likert scale. untuk kategori usabilitas, kemudahan penggunaan, dan kepuasan mendapatkan nilai 4 (setuju), sedangkan untuk kategori kemudahan untuk dimengerti mendapatkan nilai 5 (sangat setuju). dari penelitian ini secara keseluruhan dapat disimpulkan bahwa alat ini memiliki tingkat usabilitas dengan nilai modus 4 yang berarti bahwa user setuju dengan adanya *remote control smart home* ini dapat membantu mereka dalam mengontrol rumah.

Kata kunci : *Smart home, remote control, desain interaksi, likert scale*



ABSTRACT

Smart Home is a home monitoring and controlling system that has sensors, actuators, display and computer elements within. Embedded systems that are integrated in a network and can communicate with each other. Home or residence with automated system or remote system that controls each components is an example form of smart home. Some smart components require an embedded systems that integrated in a network and can communicate with each others.

This research aimed to make a smart home remote control more interactive by using a remote as a controller of smart home. Smart home remote control designed by fulfilling the principles of interaction design. the results of the verification test declares that remote control work in accordance with its design.

In smart home remote control observation test, obtained a success rate of user in carrying out a specific task (scenario) of 100% and takes an average of 9.2 seconds. analysis of the results of usability testing using questionnaires given to 30 respondents, 15 respondents with basic IT knowledge and 15 respondents with basic science of non-IT, with assessment method using Likert scale. For usability category, ease of use and satisfaction scores 4 (agree), while for category of ease of understand getting a score of 5 (highly agree/totaly agree). This research (atau sstudy. terserah kamu) as a whole can be concluded that this device have a usability rates with mode value of 4, which means that the user agrees with smart home remote control can assist them in controlling the house

Keyword: *Smart home, remote control, interaction design, likert scale*



DAFTAR ISI

PENGESAHANii
PERNYATAAN ORISINALITASiii
KATA PENGANTAR.....	.iv
ABSTRAK.....	.vi
ABSTRACT.....	.vii
DAFTAR ISIviii
DAFTAR TABEL.....	.xi
DAFTAR GAMBAR.....	.xii
DAFTAR LAMPIRANxiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	1
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan Masalah.....	2
1.6 Sistematika pembahasan.....	2
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Tinjauan Pustaka	4
2.1.1 Definisi <i>Smart Home</i>	4
2.1.2 Contoh Pengendali <i>Smart Home</i> Yang Sudah Ada	5
2.2 Dasar Teori	7
2.2.1 Desain interaksi	7
2.2.2 Arduino Nano	10
2.2.3 Arduino Uno	13
2.2.4 LCD Nokia 5510	17
2.2.5 <i>Push Button Switch</i>	18
2.2.6 LED.....	19
2.2.7 Arduino IDE	19
2.2.8 Paint	20
2.2.9 LCD Assistant.....	21



2.2.10 Eagle	22
BAB 3 METODOLOGI	24
3.1 Metodologi Penelitian	24
3.1.1 Studi Literatur	25
3.1.2 Analisis Kebutuhan Sistem	25
3.1.3 Perancangan Sistem	25
3.1.4 Implementasi Sistem	26
3.1.5 Pengujian dan Analisa Sistem	27
BAB 4 rekayasa kebutuhan	28
4.1 Analisa Kebutuhan	28
4.2 <i>System requirement</i>	28
4.3 Desain Model	29
4.3.1 <i>Use Case</i>	29
4.3.2 Identifikasi <i>Use Case</i>	29
4.3.3 Identifikasi Aktor	29
BAB 5 PERANCANGAN DAN IMPLEMENTASI	30
5.1 Perancangan	30
5.1.1 Desain Interaksi Pada <i>Remote Smart Home</i>	30
5.1.2 Perancangan Perangkat Keras	35
5.1.3 Perancangan Perangkat Lunak	42
5.1.4 Flowchart System	43
5.2 Implementasi	43
5.2.1 Implementasi Perancangan Desain Interaksi.....	43
5.2.2 Implementasi Case	45
5.2.3 Implementasi Perangkat Keras	46
5.2.4 Implementasi Perangkat Lunak.....	48
BAB 6 PENGUJIAN DAN ANALISIS.....	60
6.1 Pengujian Dan Analisis	60
6.2 Pengujian Verifikasi Alat	60
6.2.1 Pengujian Dan Analisis <i>Remote</i>	60
6.2.2 Pengujian Dan Analisis Fungsionalitas Pengiriman Data Dari <i>Remote Control</i> Ke Maket.....	65



6.2.3 Pengujian Dan Analisis Kesesuaian Nyala-Mati Led Pada Maket Berdasarkan Input Dari <i>Remote</i>	67
6.3 Pengujian dan Analisis Usabilitas.....	70
6.3.1 Pengujian Menggunakan Observasi.....	70
6.3.2 Pengujian Menggunakan Quesioner.....	73
BAB 7 PENUTUP	81
7.1 Kesimpulan.....	81
7.2 Saran	81
DAFTAR PUSTAKA.....	82
Lampiran	83



DAFTAR TABEL

Tabel 2.1 Spesifikasi Arduino nano	10
Tabel 2.2 Spesifikasi Arduino UNO.....	13
Tabel 4.1 Kebutuhan	28
Tabel 4.2 Spesifikasi Kebutuhan	28
Tabel 5.1 Perancangan Tampilan Icon	37
Tabel 5.2 Perancangan Pin LCD	39
Tabel 5.3 Implementasi Pin Tombol	48
Tabel 6.1 Verifikasi Alat.....	60
Tabel 6.2 Hasil Pengujian Tombol	63
Tabel 6.3 Hasil Pengujian LCD	65
Tabel 6.4 Hasil Pengujian Pengiriman Data	66
Tabel 6.5 Hasil Pengujian Maket.....	67
Tabel 6.6 Hasil Verifikasi Alat	69
Tabel 6.7 Tabel Pertanyaan Usefullness	73
Tabel 6.8 Tabel Pertanyaan Satisfaction	73
Tabel 6.9 Tabel Pertanyaan Easy To Use	74
Tabel 6.10 Tabel Pertanyaan Easy To Learn.....	74
Tabel 6.11 Penilaian Jumlah Pada Tingkat Kegunaan	77
Tabel 6.12 Penilaian Jumlah Pada Tingkat Kemudahan Penggunaan.....	78
Tabel 6.13 Penilaian Jumlah Pada Tingkat Kemudahan Untuk Dimengerti.....	78
Tabel 6.14 Penilaian Jumlah Pada Tingkat Kepuasan	79



DAFTAR GAMBAR

Gambar 2.1 Poin-poin pada <i>Smart Home</i>	4
Gambar 2.2 WeMo.....	6
Gambar 2.3 Logitech Harmony	6
Gambar 2.4 Proses Desain Interaksi	7
Gambar 2.5 Arduino Nano	10
Gambar 2.6 Arduino Uno	13
Gambar 2.7 LCD Nokia 5510	18
Gambar 2.8 <i>Button</i>	18
Gambar 2.9 LED.....	19
Gambar 2.10 Arduino IDE	20
Gambar 2.11 Paint	21
Gambar 2.12 LCD Assistant.....	22
Gambar 2.13 Schematic Editor	22
Gambar 2.14 Library Tool.....	23
Gambar 2.15 Eagle Board	23
Gambar 3.1 Diagram Alir.....	24
Gambar 4.1 <i>Use Case Remote</i>	29
Gambar 5.1 HTA <i>Remote Control Smart Home</i>	32
Gambar 5.2 Icon Perangkat.....	32
Gambar 5.3 Penerapan <i>Synthesizability</i>	33
Gambar 5.4 Penerapan <i>Familiarity</i>	33
Gambar 5.5 Penerapan <i>Generalizability</i>	33
Gambar 5.6 Penerapan <i>Cosistensy</i> Icon.....	34
Gambar 5.7 Penerapan <i>Cosistensy</i> Geser	34
Gambar 5.8 Penerapan <i>Observability</i>	34
Gambar 5.9 <i>Diagram Blok</i>	35
Gambar 5.10 Rangkaian Keseluruhan	36
Gambar 5.11 Icon Tombol.....	36
Gambar 5.12 Rangkaian Tombol.....	37
Gambar 5.13 Rangkaian LCD	39

Gambar 5.14 Rangkaian Maket.....	40
Gambar 5.15 Rancangan Maket.....	41
Gambar 5.16 Rancangan <i>Remote</i>	42
Gambar 5.17 Flowchart.....	43
Gambar 5.18 Implementasi Menu	44
Gambar 5.19 Implementasi Menu Lampu	44
Gambar 5.20 Implementasi Menu Kunci	44
Gambar 5.21 Implementasi Menu Elektronik.....	45
Gambar 5.22 Implementasi Menu Penyiram Tanaman.....	45
Gambar 5.23 <i>Remote</i> Tampak Depan	45
Gambar 5.24 <i>Remote</i> Tampak Samping.....	46
Gambar 5.25 <i>Remote</i> Tampak Belakang	46
Gambar 5.26 Implementasi <i>Remote Control</i>	47
Gambar 5.27 Implementasi Maket	48
Gambar 6.1 Program Pengujian Arduino	61
Gambar 6.2 Hasil Pada Arduino IDE	62
Gambar 6.3 Hasil Pada Arduino	62
Gambar 6.4 Hasil Keseluruhan Pengujian Observasi	71
Gambar 6.5 Grafik Rata-rata Penggerjaan Skenario.....	72
Gambar 6.6 Hasil Keseluruhan Pengujian Questioner	76

DAFTAR LAMPIRAN

A.1 Source Code	83
A.1.1 Uji coba tombol.....	83
A.1.2 Uji coba maket	83
A.1.3 Uji coba tombol2.....	84
A.1.4 Menu.....	85
A.1.5 Icon.....	112
A.1.6 Maket	124
A.2 Sample Observasi.....	129
A.3 Sample Quesioner	131
A.4 Dokumentasi	132



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Smart Home adalah suatu sistem pengendalian dan monitoring rumah yang memiliki sensor, aktuator, display, dan elemen komputer didalamnya. Tujuan utama dari dibuatnya suatu smart home adalah untuk meningkatkan kenyamanan, kemudahan, keamanan dan kegunaannya. Sistem-sistem yang tertanam tersebut terintegrasi dalam suatu jaringan dan dapat saling berkomunikasi satu sama lainnya (Christoffer Björkskog, 2010).

Untuk mengatur kumpulan sistem pintar ini, diperlukan suatu desain *remote control* yang dapat diakses secara jarak jauh [Santosh.M.Nejakar. 2014]. *Remote control* adalah sebuah alat yang dapat mengontrol perangkat elektronik dari jarak jauh. Dibandingkan dengan pengendali smart home berbasis web/aplikasi, *remote control* memiliki keunggulan dapat mengakses smart home tanpa ada jaringan internet. Namun tentu saja *remote* ini hanya dapat digunakan untuk pengendalian di dalam rumah.

Beberapa alat *remote control smart home* yang sudah berada dipasaran adalah berupa aplikasi yang dapat di-install di smartphone seperti apps weemo [<http://www.belkin.com/us/>], Logitech harmony [<https://support.myharmony.com/en-us/ultimate-home>]. Kedua sistem control tersebut memiliki kelebihan. Namun pengendali ini memiliki kekurangan dimana pengoperasiannya membutuhkan koneksi internet, *remote control* yang memiliki banyak tombol dan tidak mempunyai indikator untuk kondisi perangkat sehingga memiliki tingkat usabilitas alat rendah.

Untuk meningkatkan usabilitas *remote control*, penelitian ini dibuat berdasarkan disiplin ilmu Interaksi Manusia dan Komputer (IMK) yang merupakan suatu ilmu yang mempelajari manusia dan pengalamannya dalam berinteraksi dengan sistem komputasi yang menyediakan suatu layanan. Penerapan konsep Interaksi Manusia dan Komputer ini akan memaksimalkan usabilitas pada *remote control* yakni dengan menggunakan proses desain interaksi. Proses desain interasi adalah suatu proses yang sering digunakan pada IMK.

Penelitian ini ditujukan untuk membuat desain *remote control smart home* dengan menggunakan *remote* sebagai kontrol dari *smart home*. Perancangan *remote control* akan menggunakan Arduino sebagai mikrokontroler utamanya dan Arduino IDE untuk software coding dan compiler. *Remote* akan terhubung dengan LCD dan user dapat memilih mengendalikan alat elektronik tertentu dengan memilih pilihan serta dapat melihat kondisi perangkat yang tampil di LCD.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka rumusan masalah pada skripsi ini adalah:



1. Bagaimana membuat desain interaksi *remote control* untuk *smart home* yang dilengkapi dengan LCD?
2. Bagaimana menguji fungsionalitas dan *usabilitas* dari *remote control* untuk *smart home* tersebut?

1.3 Tujuan

Tujuan dari penelitian ini adalah untuk mendesain dan menguji usabilitas *remote control* yang mengontrol *smart home* dengan menggunakan maket sebagai interaksi user.

1.4 Manfaat

Manfaat dari penelitian ini adalah terdapat alternatif alat pengendali *smart home* yang dapat memberikan pilihan yang lebih luas kepada pengguna sesuai dengan kebutuhannya, dalam hal ini pengendalian dari dalam rumah.

1.5 Batasan Masalah

Berdasarkan rumusan masalah yang telah diuraikan di atas, maka batasan masalah dalam penelitian ini adalah:

1. *Remote control* ini menggunakan mikrokontroller arduino
2. *Remote control* ini dibuat untuk prototype 1
3. Implementasi *remote control* menggunakan maket
4. Implementasi sebatas *prototype* fungsional dasar, sehingga bentuk, ke-ergonomisan dari tampilan *remote control* tidak diteliti
5. Pengaturan hanya nyala dan mati
6. Penelitian ini tidak memperhatikan aspek keamanan jaringan
7. Penelitian ini menggunakan pengiriman data kabel *transciever* dan *reciever* antar arduino

1.6 Sistematika pembahasan

Untuk mencapai tujuan yang diharapkan, maka sistematika penulisan yang disusun dalam skripsi ini adalah sebagai berikut:

Bab I Pendahuluan

Bab ini berisi latar belakang, rumusan masalah, tujuan, batasan masalah, manfaat dan sistematika penulisan.

Bab II Tinjauan Pustaka

Bab ini membahas mengenai kajian pustaka dan dasar teori yang digunakan dalam penelitian desain *remote control*.

Bab III Metodologi Penelitian

Bab ini membahas metode yang digunakan dalam penelitian ini, terdiri dari penjelasan tentang studi literatur, analisis kebutuhan, perancangan sistem, pengujian, analisis hasil pengujian dan pengambilan kesimpulan.

Bab IV Rekayasa Kebutuhan

Bab ini menjelaskan persyaratan minimal yang harus dipenuhi untuk perancangan hingga implementasi. Dengan harapan perancangan dan implementasi program ini bisa berjalan dengan baik.

Bab V Perancangan dan Implementasi

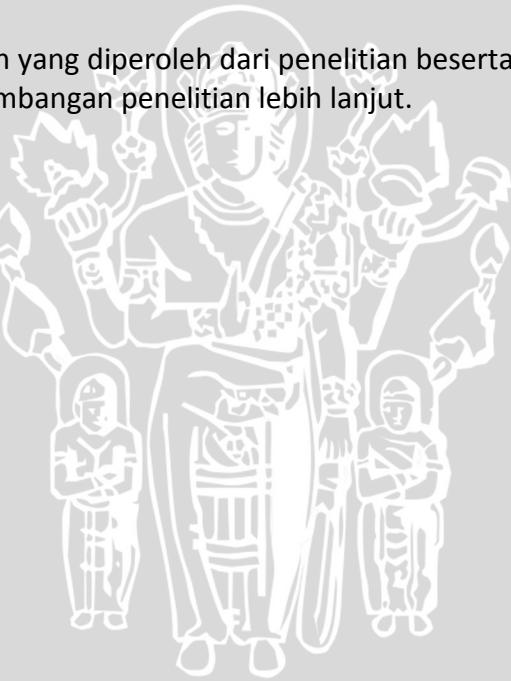
Bab ini berisi tentang algoritma/*coding*/pemrograman yang digunakan pada desain *remote* kontrol *smart home*, yaitu tombol yang di gunakan, algoritma system, coding keseluruhan *remote* yang mencangkup menu, kunci lampu, alat elektronik dan penyiram tanaman

Bab VI Pengujian Dan Analisis

Bab ini berisi hasil pengujian yang dilakukan terhadap pemrograman yang dibangun beserta analisisnya.

Bab VII Penutup

Bab ini berisi kesimpulan yang diperoleh dari penelitian beserta saran yang dapat digunakan untuk pengembangan penelitian lebih lanjut.



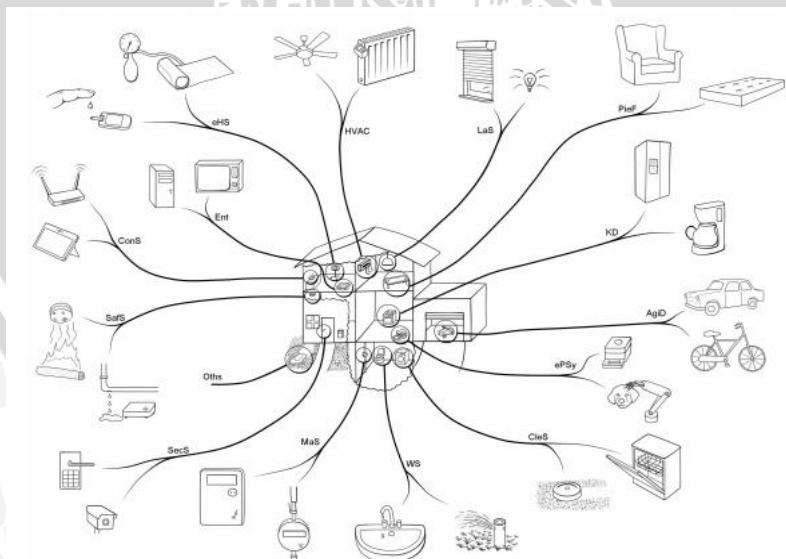
BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Pada bagian ini dijelaskan tentang definisi *smart home*, penelitian terkait, dan dasar teori.

2.1.1 Definisi Smart Home

Michael Schiefer menyatakan bahwa ada beberapa definisi dari *smart home*. Yang pertama dari web *smart home energy*, adalah sebuah rumah pintar yang menggabungkan sistem otomatisasi canggih untuk memberikan kemudahan pada masyarakat dengan pemantauan canggih dan kontrol atas fungsi pada rumah. Misalnya smart home dapat mengontrol pencahayaan, suhu, multi-media, keamanan, jendela dan pintu operasi, serta banyak fungsi lainnya. Disisi lain mengatakan bahwa smart home adalah tempat tinggal yang membantu manusia dengan teknologi modern dan layanan elektronik baru. Sebuah aspek penting dalam definisi ini adalah integrasi sistem perangkat serta koneksi dari tempat tinggal dengan *remote control*. Dari beberapa pengertian smart home yang telah disebutkan, dapat disimpulkan bahwa Smart Home adalah suatu perangkat dengan fungsi-fungsi tertentu terdapat 15 point (Michael Schiefer, 2015) dengan tujuan untuk *control* jarak jauh melalui instalasi jaringan yang telah ditentukan sebelumnya. Bentuk dari smart home dapat digunakan sebagai *alternative* atau infrastruktur tambahan sebagai otomatisasi perangkat. Smart home dapat diimplementasikan pada rumah pribadi maupun bangunan industri. Namun pada penelitian ini hanya akan berfokus pada rumah-rumah pribadi saja, skema dapat dilihat pada Gambar 2.1.



Gambar 2.1 Poin-poin pada *Smart Home*

Sumber: Michael Schiefer

Adapun 15 Poin Yang Ada Adalah:

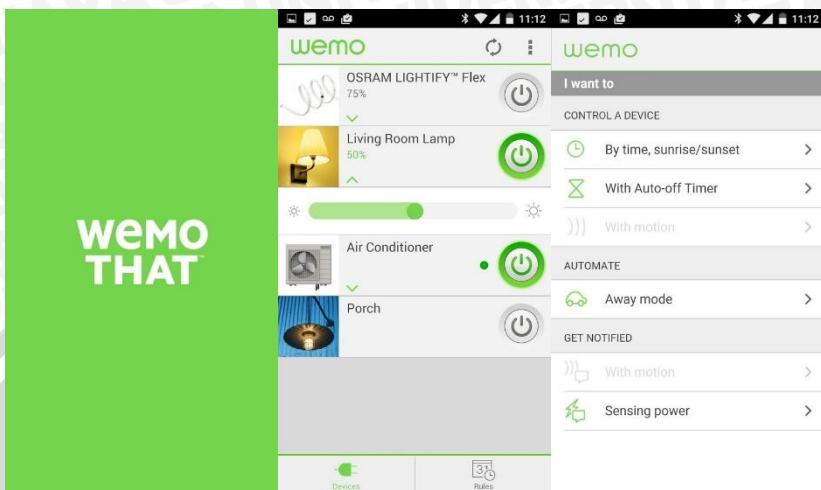
1. *Controlling Systems (ConS)* seperti *control* utama dalam alat, dimana system ini hanya dapat mengontrol smart home seperti melalui sebuah tablet, smartphone dan aplikasi untuk tujuan yang serupa.
2. *Security Systems (SecS)* category ini mencangkup sistem kunci, camera pengintai, dan system sejenisnya. Dari beberapa klasifikasi yang ada ini adalah bagian dari keselamatan dan keamanan
3. *Safety Systems (SafS)* produk ini termasuk categori penggunaan deteksi dan menghindari ancaman untuk keselamatan atau konsisi fisik, seperti pendeteksi dari gas atau kebocoran air.
4. *eHealth Systems (eHS)* perangkat untuk pemeriksaan medis, tapi bantuan medis juga masuk dalam category ini. eHealth itu topic yang sangat luas, tapi pada konteks smart home hal ini bersangkutan.
5. *Measurement and Sensors (MaS)* meteran air, meteran listrik dan perangkat yang sejenisnya adalah bagian grup ini. Sensor diskrit dan pendeteksi juga masuk, selama mereka bukan bagian dari SafS atau SecS.
6. *Heating, Ventilation and Airconditioning (HVAC)* kategori dari sistem ini digunakan untuk mengendalikan suhu ruangan dan ventilasi udara seperti thermostats, climate control units atau ventilator
7. *Light and Shadow (LaS)* sebuah perangkat pemancar atau mencegah cahaya seperti lampu, tenda dan rol buta merupakan bagian dari kategori ini.
8. *Kitchen Devices (KD)* berarti produk yang di gunakan di dapur, seperti penanak nasi, kulkas dan coffee maker.
9. *Water Systems (WS)* keran, bak mandi dan toilet demikian juga penyiram tanaman dan banyak lagi.
10. *Cleaning Systems (CleS)* system untuk kebersihan seperti mesin cuci, pencuci piring.
11. *e-Pet Systems (ePSy)* kategori ini terdiri dari perangkat yang di sekeliling hewan peliharaan, seperti kalung untuk mendeteksi keberadaan, system pemberi makan otomatis.
12. *Entertainment (Ent)* system audio, televisi, game, dan lainnya.
13. *Pieces of Furniture (PieF)* akomodasi tempat duduk dan tempat tidur, seperti pemesanan meja dan lemari dan sebagainya.
14. *agility Devices (AgiD)* perangkat untuk transportasi seseorang, seperti mobil, sepeda, dan lain-lain.
15. *Others (Oths)* memungkinkan untuk menemukan system yang tidak dapat menjadi category lain.

2.1.2 Contoh Pengendali *Smart Home* Yang Sudah Ada

2.1.2.1 WeMo

WeMo merupakan salah satu produksi pengontrol rumah yang berbasis aplikasi android. WeMo ini memiliki keunggulan mengetahui perhitungan konsumsi listrik pada perangkat yang terintegrasi. WeMo bekerja dengan jaringan internet dengan

menghubungkan aplikasi pada smartphone dengan perangkat-perangkat menggunakan WeMo *Controller*. Kelemahan pada WeMo ini adalah pengiriman perintah lewat internet sering kali gagal atau pengaturan ulang setiap aplikasi mengalami restart.



Gambar 2.2 WeMo

Sumber: dokumentasi aplikasi WeMo

2.1.2.2 Logitech harmony

Logitech harmony merupakan pengontrol rumah yang dapat mengakses perangkat-perangkat rumah dengan praktis. Logitech harmony bekerja pada *remote* dan aplikasi pada android. Keunggulan *remote* dapat mengakses berbagai perangkat yang dihubungkan dengan hub harmony yang bertujuan untuk komunikasi antar perangkat, sayangnya dalam pengiriman data pada *remote* sering lemot dan *remote* tersebut harus diarahkan ke hub harmony dikarenakan memakai IR(infrared). Pada aplikasi harmony dapat mengakses dari jarak jauh dan dapat mengakses beberapa perangkat dengan menggunakan jaringan internet, sayangnya pada pengiriman data pada aplikasi ini terkadang salah dan sering meminta sinkronisasi pada perangkat karena terjadi kesalahan ip perangkat.



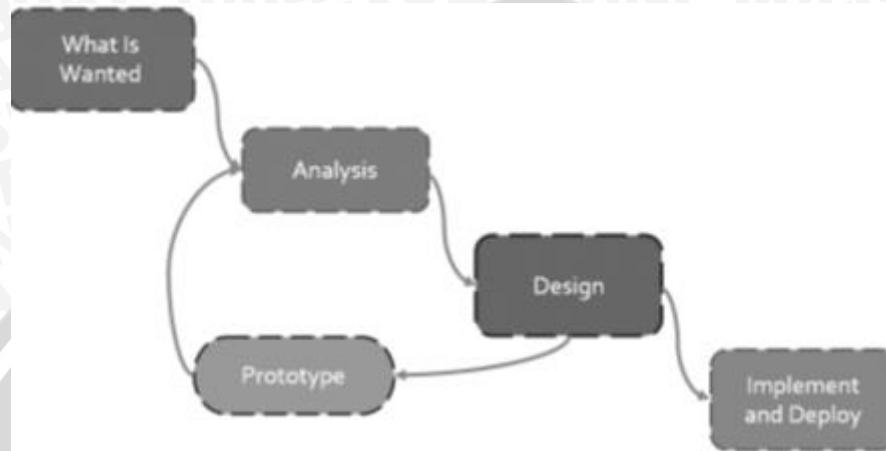
Gambar 2.3 Logitech Harmony

Sumber: Dokumentasi Logitech Harmony

2.2 Dasar Teori

2.2.1 Desain interaksi

Pada proses desain interaksi terdapat 5 bagian yaitu *what is wanted*, *analysis*, *design*, *prototype*, dan *implement/deploy* (Dix Alan et al, 2004). Bagian tersebut digambarkan pada gambar 2.4 proses desain interaksi.



Gambar 2.4 Proses Desain Interaksi

Sumber: HCI 3rd Edition

2.2.1.1 *What is wanted*

Bagian ini merupakan penentuan dari, untuk siapa dan dimana alat ini digunakan. Poin ini sangat penting, karena merupakan gagasan pertama alat untuk tindakan interaksi desain selanjutnya. Sehingga ketika alat selesai maka dapat diimplementasikan sesuai dengan tujuan yang telah ditetapkan sebelumnya.

2.2.1.2 *Analysis*

Ditinjau dari pembentuk katanya, Task Analysis berasal dari kata Analisis yang berarti proses pencarian jalan keluar (pemecahan masalah) yang berangkat dari dugaan akan kebenarannya, sedangkan Task berarti pekerjaan atau perintah atau melakukan sesuatu fungsi. Jadi, dapat disimpulkan Task Analysis adalah proses pencarian jalan keluar terhadap sebuah pekerjaan dan juga Proses menganalisis dan menggambarkan bagaimana manusia melaksanakan tugas / pekerjaannya, apa saja yang dilakukan /peralatan apa yang digunakan dan hal-hal apa saja yang perlu diketahui. Task analysis ini merupakan proses menganalisis tentang cara pengguna dalam mengerjakan, menyelesaikan dan bereaksi terhadap tugas dari suatu sistem(Dix Alan et al, 2004). *Hierarchical task analysis* (HTA) adalah metode untuk menganalisa pekerjaan user: tentang apa yang dikerjakan user, dengan apa user bekerja dan apa yang harus diketahui user. Task analysis ini merupakan proses menganalisa skenario tentang cara user dalam mengerjakan, menyelesaikan dan bereaksi terhadap tugas dari suatu system dan keingintahuan user terhadap system. Bentuk dari runtutan cerita atau tahapan yang dapat

berupa gambar maupun cerita dalam bentuk poin merupakan definisi dari skenario. Skenario berfungsi untuk mempermudahkan perancang desain interaksi dalam mengenali kinerja alat melalui gambar atau cerita yang telah disiapkan.

2.2.1.3 Design

Untuk meningkatkan sisi usabilitas dari alat maka diperlukan adanya suatu design rule yang mencangkup 3 type, yakni Principles, Standard, dan Guideline dalam proses perancangan alat. Pengertian Prinsip Usability Interaksi Manusia-Komputer (IMK) adalah suatu masalah optimasi penggunaan sistem yang digunakan oleh pengguna. Sistem akan bekerja dengan baik apabila dipergunakan secara maksimal oleh pengguna sehingga semua kemampuan sistem dapat dimanfaatkan secara maksimal (Dix Alan et al, 2004). Pada poin principles terdapat 3 kategori utama yakni Learnability, Flexibility dan Robustness.

1. Learnability

Pada fungsi ini mempunyai tujuan agar pengguna pemula mampu mempelajari sistem dan memanfaatkannya secara optimal, pada fungsi ini masih dibagi menjadi 5 fungsi utama yakni:

- a. *Predictability*: Pengguna dapat menebak hasil akhir dari tindakan yang mereka lakukan.
- b. *Synthesizability*: Pengguna dapat melihat hasil atau tanda adanya proses dengan sesegera mungkin.
- c. *Familiarity*: melakukan analogi dalam desain sistem dengan kehidupan sehari-hari atau dengan konsep yang telah dianggap populer.
- d. *Generalizability*: desain operasi sistem yang pada umumnya sudah berlaku dan sama dengan aplikasi lain yang sejenis.
- e. *Consistency*: konsisten dalam penggunaan berbagai istilah maupun ukuran.

2. Flexibility

Berbeda dengan fungsi – fungsi sebelumnya yang digunakan untuk mengatur, pada fungsi ini digunakan untuk sebuah aplikasi se bisa mungkin dapat dioperasikan dengan prosedur yang tidak kaku atau ada sebuah ke fleksibilitas. Ada beberapa konsep dalam flexibility antara lain:

- a. *Dialogue initiative*: pengguna memiliki kebebasan dalam sebuah dialog atau komunikasi dengan alat.
- b. *Multi-Threading*: pengguna dapat menjalankan aplikasi lain ataupun proses lain di saat sebuah proses atau sistem sedang dijalankan, atau dapat diartikan juga sebagai pengguna memiliki beberapa pilihan untuk memberikan suatu task kepada alat.
- c. *Task Migratability*: kemampuan alat untuk diberikan migrasi suatu task dari pengguna.
- d. *Substitutivity*: sebuah perintah dapat diganti dengan padanan lain.
- e. *Customizability*: desain dapat dimodifikasi oleh pengguna secara adaptif atau sesuai dengan tujuan utama masing-masing.

3. Robustness

Prinsip ini diartikan sebagai kehandalan sebuah sistem dalam mencapai tujuan, khususnya dari sudut pandang pengguna. *Help system* merupakan salah satu contoh dari *robustness*, *help system* ini biasanya digunakan ketika pengguna dalam kesulitan mungkin berupa sistem yang gagal atau tidak sesuai dengan harapan. *help system* yang membantu pengguna dalam pengoperasian ini harus menerapkan fungsi *error handling* dan *predictable behavior*. Dalam pencapaian poin ini dibutuhkan empat kriteria yaitu:

- a. *Observability*: pengguna bisa melakukan observasi pendahuluan sebelum benar-benar melakukan proses yang sesungguhnya, maupun saat proses tersebut berlangsung.
- b. *Recoverability*: kemampuan koreksi dari sistem jika pengguna melakukan kesalahan.
- c. *Responsiveness*: sebuah sistem yang responsif berarti mampu menerima tindakan *user* dengan stabil dan segera tanpa ada kendala yang timbul akibat komunikasi dari pengguna.
- d. *Task Conformance*: kesesuaian eksekusi perintah, sesuai dengan Task/tugas yang diberikan pengguna.

2.2.1.4 *Prototype*

Prototype adalah proses pembuatan model sederhana dari sebuah sistem atau alat yang mengijinkan pengguna memiliki gambaran dasar tentang program serta melakukan pengujian awal (Raymond McLeod (2001 : 11). Prototype memberikan fasilitas bagi pengembang dan pemakai untuk saling berinteraksi selama proses pembuatan, sehingga pengembang dapat dengan mudah memodelkan perangkat lunak yang akan dibuat. Model - model prototype:

1. Prototype kertas atau model berbasis komputer yang menjelaskan bagaimana interaksi antara pemakai dan komputer.
2. Prototype yang mengimplementasikan beberapa bagian fungsi dari alat yang dibuat yang sesungguhnya. Dengan cara ini pemakai akan mendapat gambaran tentang program, bentuk maupun desain yang akan dihasilkan, sehingga dapat menjabarkan lebih rinci kebutuhannya.
3. Menggunakan sistem yang sudah ada. Seringkali pembuat sistem atau alat memiliki beberapa ide atau gagasan yang sebagian dari ide tersebut mirip dengan program yang akan dibuat.

2.2.1.5 *Implement and Deploy*

implement and deploy merupakan tahapan terakhir dari bagan desain interaksi yang digagas oleh alan dix. Sesuai dengan namanya ketika sistem atau alat sudah finish, dalam arti telah mengalami banyak perubahan dari sisi desain analisis dan berpengaruh pada hasil jika masih belum sesuai maka akan dibuat prototype, dianalisis dan didesain ulang sampai menemukan desain yang paling tepat. Ketika sudah didapatkan desain yang tepat dan sesuai maka alat siap diimplementasikan dan dipasarkan sesuai dengan poin awal *what is wanted* yang telah ditentukan sebelumnya.

2.2.2 Arduino Nano

Arduino Nano adalah salah satu papan pengembangan mikrokontroler yang berukuran kecil, lengkap dan mendukung penggunaan breadboard. Arduino Nano diciptakan dengan basis mikrokontroler ATmega328 (untuk Arduino Nano versi 3.x) atau ATmega 168 (untuk Arduino versi 2.x).



Gambar 2.5 Arduino Nano

Sumber: <https://www.arduino.cc/>

Arduino Nano kurang lebih memiliki fungsi yang sama dengan Arduino Duemilanove, tetapi dalam paket yang berbeda. Arduino Nano tidak menyertakan colokan DC berjenis Barrel Jack, dan dihubungkan ke komputer menggunakan port USB Mini-B. Arduino Nano dirancang dan diproduksi oleh perusahaan Gravitech. Spesifikasi dari Arduino Nano terdapat pada Tabel 2.1.

Tabel 2.1 Spesifikasi Arduino nano

Sumber: <https://www.arduino.cc/>

Microcontroller	Atmel ATmega168 or ATmega328
Tegangan operasi	5 V
Input Voltage (disarankan)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (6 pin digunakan sebagai output PWM)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) atau 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) atau 1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"
Length	45 mm
Width	18 mm

Weigth	5 g
--------	-----

Arduino Nano dapat diaktifkan melalui koneksi USB Mini-B, atau melalui catu daya eksternal dengan tegangan belum teregulasi antara 6-20 Volt yang dihubungkan melalui pin 30 atau pin VIN, atau melalui catu daya eksternal dengan tegangan teregulasi 5 volt melalui pin 27 atau pin 5V. Sumber daya akan secara otomatis dipilih dari sumber tegangan yang lebih tinggi.

ATmega168 memiliki 16 KB flash memory untuk menyimpan kode (2 KB digunakan untuk bootloader); Sedangkan ATmega328 memiliki flash memory sebesar 32 KB, (juga dengan 2 KB digunakan untuk bootloader). ATmega168 memiliki 1 KB memory pada SRAM dan 512 byte pada EEPROM (yang dapat dibaca dan ditulis dengan EEPROM library); Sedangkan ATmega328 memiliki 2 KB memory pada SRAM dan 1 KB pada EEPROM.

Masing-masing dari 14 pin digital pada Arduino Nano dapat digunakan sebagai input atau output, dengan menggunakan fungsi “pinMode()”, “digitalWrite()”, dan “digitalRead()”. Semua pin beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor pull-up internal (yang terputus secara default) sebesar 20-50 KOhm. Selain itu beberapa pin memiliki fungsi khusus, yaitu:

- **Serial** : 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. Pin ini terhubung ke pin yang sesuai dari chip FTDI USB-to-TTL Serial.
- **External Interrupt** (Interupsi Eksternal): Pin 2 dan pin 3 ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau perubahan nilai.
- **PWM** : Pin 3, 5, 6, 9, 10, dan 11. Menyediakan output PWM 8-bit dengan fungsi analogWrite().
- **SPI** : Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI. Sebenarnya komunikasi SPI ini tersedia pada hardware, tapi untuk saat belum didukung dalam bahasa Arduino.
- **LED** : Pin 13. Tersedia secara built-in pada papan Arduino Nano. LED terhubung ke pin digital 13. Ketika pin diset bernilai HIGH, maka LED menyala, dan ketika pin diset bernilai LOW, maka LED padam.

Arduino Nano memiliki 8 pin sebagai input analog, diberi label A0 sampai dengan A7, yang masing-masing menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara default pin ini dapat diukur/diatur dari mulai Ground sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan fungsi analogReference(). Pin Analog 6 dan 7 tidak dapat digunakan sebagai pin digital. Selain itu juga, beberapa pin memiliki fungsi yang dikhususkan, yaitu:

- **I2C** : Pin A4 (SDA) dan pin A5 (SCL). Yang mendukung komunikasi I2C (TWI) menggunakan perpustakaan Wire.

Masih ada beberapa pin lainnya pada Arduino Nano, yaitu:



- **AREF** : Referensi tegangan untuk input analog. Digunakan dengan fungsi `analogReference()`.
- **RESET** : Jalur LOW ini digunakan untuk me-reset (menghidupkan ulang) mikrokontroler. Biasanya digunakan untuk menambahkan tombol reset pada shield yang menghalangi papan utama Arduino.

Arduino Nano juga memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, dengan Arduino lain, atau dengan mikrokontroler lainnya. ATmega168 dan ATmega328 menyediakan komunikasi serial UART TTL (5 Volt), yang tersedia pada pin digital 0 (RX) dan pin 1 (TX). Sebuah chip FTDI FT232RL yang terdapat pada papan Arduino Nano digunakan sebagai media komunikasi serial melalui USB dan driver FTDI (tersedia pada software Arduino IDE) yang akan menyediakan COM Port Virtual (pada Device komputer) untuk berkomunikasi dengan perangkat lunak pada komputer.

Perangkat lunak Arduino termasuk didalamnya serial monitor memungkinkan data tekstual sederhana dikirim ke dan dari papan Arduino. LED RX dan TX yang tersedia pada papan akan berkedip ketika data sedang dikirim atau diterima melalui chip FTDI dan koneksi USB yang terhubung melalui USB komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

Sebuah library “SoftwareSerial” memungkinkan komunikasi serial pada beberapa pin digital Nano. ATmega168 dan ATmega328 juga mendukung komunikasi I2C (TWI) dan SPI. Perangkat lunak Arduino termasuk library Wire digunakan untuk menyederhanakan penggunaan bus I2C. Untuk komunikasi SPI, silakan lihat datasheet ATmega168 atau ATmega328.

Arduino Nano dapat diprogram dengan software Arduino IDE. Pilih “Arduino Diecimila, Duemilanove, atau Nano w/ ATmega168” atau “Arduino Duemilanove atau Nano w/ ATmega328” melalui menu Tools > Board yang disesuaikan dengan jenis mikrokontrolernya.

ATmega168 dan ATmega328 pada Arduino Nano sudah dipaket preburned dengan bootloader yang memungkinkan peng-upload-an kode baru tanpa menggunakan programer hardware eksternal. Hal ini karena komunikasi yang terjadi menggunakan protokol asli STK500. Dan dapat melewati (bypass) bootloader dan program mikrokontroler melalui pin header ICSP (In-Circuit Serial Programming) menggunakan Arduino ISP atau yang sejenis.

Pengaturan ini juga memiliki implikasi lain. Ketika Arduino Nano terhubung dengan komputer yang menggunakan sistem operasi Mac OS X atau Linux, papan Arduino akan di-reset setiap kali dihubungkan dengan software komputer (melalui USB). Dan setengah detik kemudian atau lebih, bootloader berjalan pada papan Arduino Nano. Proses reset melalui program ini digunakan untuk mengabaikan data yang cacat (yaitu apapun selain meng-upload kode baru), ia akan memotong dan membuang beberapa byte pertama dari data yang dikirim ke papan setelah sambungan terbuka. Jika sebuah sketsa dijalankan pada papan untuk menerima satu kali konfigurasi atau menerima data lain ketika pertama kali dijalankan,



pastikan bahwa perangkat lunak diberikan waktu untuk berkomunikasi dengan menunggu beberapa detik setelah terkoneksi dan sebelum mengirim data.

2.2.3 Arduino Uno



Gambar 2.6 Arduino Uno

Sumber: www.arduino.cc

Arduino UNO adalah sebuah board mikrokontroler yang didasarkan pada ATmega328. Arduino UNO mempunyai 14 pin digital input/output (6 di antaranya dapat digunakan sebagai output PWM), 6 input analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah power jack, sebuah ICSP header, dan sebuah tombol reset. Arduino UNO memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah computer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya.

Arduino Uno berbeda dari semua board Arduino sebelumnya, Arduino UNO tidak menggunakan chip driver FTDI USB-to-serial. Sebaliknya, fitur-fitur Atmega16U2 (Atmega8U2 sampai ke versi R2) diprogram sebagai sebuah pengubah USB ke serial. Revisi 2 dari board Arduino Uno mempunyai sebuah resistor yang menarik garis 8U2 HWB ke ground, yang membuatnya lebih mudah untuk diletakkan ke dalam DFU mode. Revisi 3 dari board Arduino UNO memiliki fitur-fitur baru sebagai berikut:

- Pinout 1.0: ditambah pin SDA dan SCL yang dekat dengan pin AREF dan dua pin baru lainnya yang diletakkan dekat dengan pin RESET, IOREF yang memungkinkan shield-shield untuk menyesuaikan tegangan yang disediakan dari board. Untuk ke depannya, shield akan dijadikan kompatibel/cocok dengan board yang menggunakan AVR yang beroperasi dengan tegangan 5V dan dengan Arduino Due yang beroperasi dengan tegangan 3.3V. Yang ke-dua ini merupakan sebuah pin yang tak terhubung, yang disediakan untuk tujuan kedepannya. Atmega 16U2 mengantikan 8U2.

Tabel 2.2 Spesifikasi Arduino UNO

Sumber: www.arduino.cc

Mikrokontroler	ATmega328
----------------	-----------

Tegangan pengoperasian	5V
Tegangan input yang disarankan	7-12V
Batas tegangan input	6-20V
Jumlah pin I/O digital	14 (6 di antaranya menyediakan keluaran PWM)
Jumlah pin input analog	6
Arus DC tiap pin I/O	40 mA
Arus DC untuk pin 3.3V	50 mA
Memori Flash	32 KB (ATmega328), sekitar 0.5 KB digunakan oleh bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Length	68,6 mm
Width	53.4 mm
Weight	25 g

Arduino UNO dapat disuplai melalui koneksi USB atau dengan sebuah power suplai eksternal. Sumber daya dipilih secara otomatis. Suplai eksternal (non-USB) dapat diperoleh dari sebuah adaptor AC ke DC atau battery. Adaptor dapat dihubungkan dengan mencolokkan sebuah center-positive plug yang panjangnya 2,1 mm ke power jack dari board. Kabel lead dari sebuah battery dapat dimasukkan dalam header/kepala pin Ground (Gnd) dan pin Vin dari konektor POWER.

Board Arduino UNO dapat beroperasi pada sebuah suplai eksternal 6 sampai 20 Volt. Jika disuplai dengan yang lebih kecil dari 7 V, kiranya pin 5 Volt mungkin mensuplai kecil dari 5 Volt dan board Arduino UNO bisa menjadi tidak stabil. Jika menggunakan suplai yang lebih dari besar 12 Volt, voltage regulator bisa kelebihan panas dan membahayakan board Arduino UNO. Range yang direkomendasikan adalah 7 sampai 12 Volt.

Pin-pin dayanya adalah sebagai berikut:

- VIN. Tegangan input ke Arduino board ketika board sedang menggunakan sumber suplai eksternal (seperti 5 Volt dari koneksi USB atau sumber tenaga lainnya yang diatur). Kita dapat menyuplai tegangan melalui pin ini, atau jika penyuplaiannya tegangan melalui power jack, aksesnya melalui pin ini.
- 5V. Pin output ini merupakan tegangan 5 Volt yang diatur dari regulator pada board. Board dapat disuplai dengan salah satu suplai dari DC power jack (7-12V), USB connector (5V), atau pin VIN dari board (7-12). Penyuplaiannya tegangan melalui pin 5V atau 3,3V membypass regulator, dan dapat membahayakan board. Hal itu tidak dianjurkan.
- 3V3. Sebuah suplai 3,3 Volt dihasilkan oleh regulator pada board. Arus maksimum yang dapat dilalui adalah 50 mA.
- GND. Pin ground.

ATmega328 mempunyai 32 KB (dengan 0,5 KB digunakan untuk bootloader). ATmega 328 juga mempunyai 2 KB SRAM dan 1 KB EEPROM (yang dapat dibaca dan ditulis (RW/read and written) dengan EEPROM library).

Setiap 14 pin digital pada Arduino Uno dapat digunakan sebagai input dan output, menggunakan fungsi pinMode(), digitalWrite(), dan digitalRead(). Fungsi-fungsi tersebut beroperasi di tegangan 5 Volt. Setiap pin dapat memberikan atau menerima suatu arus maksimum 40 mA dan mempunyai sebuah resistor pull-up (terputus secara default) 20-50 kOhm. Selain itu, beberapa pin mempunyai fungsi-fungsi spesial:

- Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan memancarkan (TX) serial data TTL (Transistor-Transistor Logic). Kedua pin ini dihubungkan ke pin-pin yang sesuai dari chip Serial Atmega8U2 USB-ke-TTL.
- External Interrupts: 2 dan 3. Pin-pin ini dapat dikonfigurasikan untuk dipicu sebuah interrupt (gangguan) pada sebuah nilai rendah, suatu kenaikan atau penurunan yang besar, atau suatu perubahan nilai. Lihat fungsi "attachInterrupt()" untuk lebih jelasnya.
- PWM: 3, 5, 6, 9, 10, dan 11. Memberikan 8-bit PWM output dengan fungsi analogWrite().
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin-pin ini mensupport komunikasi SPI menggunakan SPI library.
- LED: 13. Ada sebuah LED yang terpasang, terhubung ke pin digital 13. Ketika pin bernilai HIGH LED menyala, ketika pin bernilai LOW LED mati.

Arduino UNO mempunyai 6 input analog, diberi label A0 sampai A5, setiapnya memberikan 10 bit resolusi (contohnya 1024 nilai yang berbeda). Secara default, 6 input analog tersebut mengukur dari ground sampai tegangan 5 Volt, dengan itu mungkin untuk mengganti batas atas dari rangenya dengan menggunakan pin AREF dan fungsi analogReference(). Di sisi lain, beberapa pin mempunyai fungsi spesial:

- TWI: pin A4 atau SDA dan pin A5 atau SCL. Mensupport komunikasi TWI dengan menggunakan Wire library.
Ada sepasang pin lainnya pada board:
- AREF: Referensi tegangan untuk input analog. Digunakan dengan "analogReference()".
- Reset: Membawa saluran ini LOW untuk mereset mikrokontroler. Secara khusus, digunakan untuk menambahkan sebuah tombol reset untuk melindungi yang memblock sesuatu pada board.

Arduino UNO mempunyai sejumlah fasilitas untuk komunikasi dengan sebuah komputer, Arduino lainnya atau mikrokontroler lainnya. Atmega 328 menyediakan serial komunikasi UART TTL (5V), yang tersedia pada pin digital 0 (RX) dan 1 (TX). Sebuah Atmega 16U2 pada channel board serial komunikasinya melalui USB dan muncul sebagai sebuah port virtual ke software pada komputer. Firmware 16U2 menggunakan driver USB COM standar, dan tidak ada driver eksternal yang dibutuhkan. Bagaimanapun, pada Windows, sebuah file inf pasti

dibutuhkan. Software Arduino mencakup sebuah serial monitor yang memungkinkan data tekstual terkirim ke dan dari board Arduino. LED RX dan TX pada board akan menyala ketika data sedang ditransmit melalui chip USB-to-serial dan koneksi USB pada komputer (tapi tidak untuk komunikasi serial pada pin 0 dan 1).

Sebuah “SoftwareSerial” library memungkinkan untuk komunikasi serial pada beberapa pin digital UNO. Atmega328 juga mensupport komunikasi I2C (TWI) dan SPI. Software Arduino mencakup sebuah “Wire library” untuk memudahkan menggunakan bus I2C.

Arduino Uno dapat diprogram dengan software Arduino IDE. Pilih “Arduino Uno dari menu Tools > Board (termasuk mikrokontroler pada board). Untuk lebih jelas, lihat referensi atau turorial.

ATmega328 pada Arduino Uno hadir dengan sebuah bootloader yang memungkinkan untuk mengupload kode baru ke ATmega328 tanpa menggunakan pemrogram hardware eksternal. ATmega328 berkomunikasi menggunakan protokol STK500 asli (referensi, file C header).

Kita juga dapat membypass bootloader dan program mikrokontroler melalui kepala/header ICSP (In-Circuit Serial Programming).

Sumber kode firmware ATmega16U2 (atau 8U2 pada board revisi 1 dan revisi 2) tersedia. ATmega16U2/8U2 diload dengan sebuah bootloader DFU, yang dapat diaktifkan dengan:

- Pada board Revisi 1: Dengan menghubungkan jumper solder pada belakang board (dekat peta Italy) dan kemudian mereset 8U2
- Pada board Revisi 2 atau setelahnya: Ada sebuah resistor yang menarik garis HWB 8U2/16U2 ke ground, dengan itu dapat lebih mudah untuk meletakkan ke dalam mode DFU. Kita dapat menggunakan software Atmel’s FLIP (Windows) atau pemrogram DFU (Mac OS X dan Linux) untuk meload sebuah firmware baru. Atau kita dapat menggunakan header ISP dengan sebuah pemrogram eksternal (mengoverwrite bootloader DFU).

Arduino Uno didesain pada sebuah cara yang memungkinkannya untuk direset dengan software yang sedang berjalan pada komputer yang sedang terhubung tanpa membutuhkan penekanan fisik dari tombol reset. Salah satu garis kontrol aliran hardware (DTR) dari ATmega8U2/16U2 sihubungkan ke garis reset dari ATmega328 melalui sebuah kapasitor 100 nanofarad. Ketika saluran ini dipaksakan (diambil rendah), garis reset jatuh cukup panjang untuk mereset chip. Software Arduino menggunakan kemampuan ini untuk memungkinkan kita untuk mengupload kode dengan mudah menekan tombol upload di software Arduino. Ini berarti bahwa bootloader dapat mempunyai sebuah batas waktu yang lebih singkat, sebagai penurunan dari DTR yang dapat menjadi koordinasi yang baik dengan memulai penguploadan.

Pengaturan ini mempunyai implikasi. Ketika Arduino Uno dihubungkan ke sebuah komputer lain yang sedang running menggunakan OS Mac X atau Linux, Arduino Uno mereset setiap kali sebuah koneksi dibuat dari software (melalui USB). Untuk berikutnya, setengah-detik atau lebih, bootloader sedang berjalan



pada Arduino UNO. Ketika Arduino UNO diprogram untuk mengabaikan data yang cacat/salah (contohnya apa saja selain sebuah penguploadan kode baru) untuk menahan beberapa bit pertama dari data yang dikirim ke board setelah sebuah koneksi dibuka. Jika sebuah sketch sedang berjalan pada board menerima satu kali konfigurasi atau data lain ketika sketch pertama mulai, memastikan bahwa software yang berkomunikasi menunggu satu detik setelah membuka koneksi dan sebelum mengirim data ini.

Arduino Uno berisikan sebuah jejak yang dapat dihapus untuk mencegah reset otomatis. Pad pada salah satu sisi dari jejak dapat disolder bersama untuk mengaktifkan kembali. Pad itu diberi label “RESET-RN” Kita juga dapat menonaktifkan reset otomatis dengan menghubungkan sebuah resistor 110 ohm dari tegangan 5V ke garis reset.

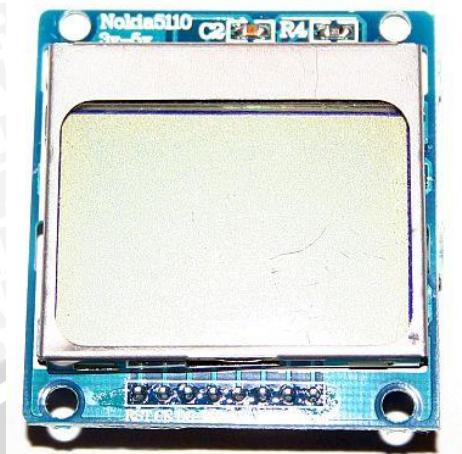
Arduino UNO mempunyai sebuah sebuah sekring reset yang memproteksi port USB komputer dari hubungan pendek dan arus lebih. Walaupun sebagian besar komputer menyediakan proteksi internal sendiri, sekring menyediakan sebuah proteksi tambahan. Jika lebih dari 500 mA diterima port USB, sekring secara otomatis akan memutuskan koneksi sampai hubungan pendek atau kelebihan beban hilang.

Panjang dan lebar maksimum dari PCB Arduino UNO masing-masingnya adalah 2.7 dan 2.1 inci, dengan konektor USB dan power jack yang memperluas dimensinya. Empat lubang sekrup memungkinkan board untuk dipasangkan ke sebuah permukaan atau kotak. Sebagai catatan, bahwa jarak antara pin digital 7 dan 8 adalah 160 mil. (0.16"), bukan sebuah kelipatan genap dari jarak 100 mil dari pin lainnya.

2.2.4 LCD Nokia 5510

Nokia 5110 adalah grafis layar LCD dasar untuk banyak aplikasi. Layar ini pada awalnya ditujukan untuk sebagai layar ponsel. Layar ini ini dipasang dengan mudah untuk solder PCB.





Gambar 2.7 LCD Nokia 5510

Sumber:

Nokia 5510 menggunakan kontroler PCD8544, yang sama yang digunakan dalam Nokia 3310 LCD. PCD8544 adalah CMOS daya LCD controller / driver rendah, yang dirancang untuk mengarahkan tampilan grafis dari 48 baris dan 84 kolom. Semua fungsi yang diperlukan untuk layar disediakan dalam satu chip, termasuk generasi on-chip pasokan LCD dan bias tegangan, sehingga minimal komponen eksternal dan konsumsi daya yang rendah. Interface PCD8544 ke mikrokontroler melalui antarmuka serial bus.

2.2.5 Push Button Switch

Push button switch adalah perangkat yang berfungsi untuk memutuskan atau menghubungkan aliran arus listrik. Push button switch hanya memiliki 2 kondisi, yaitu On dan Off (1 dan 0).



Gambar 2.8 Button

Berdasarkan fungsi kerjanya, push button switch mempunyai 2 tipe kontak yaitu NC (Normally Close) dan NO (Normally Open).

- NO (Normally Open), merupakan kontak terminal dimana kondisi normalnya terbuka (aliran arus listrik tidak mengalir). Dan ketika tombol saklar ditekan, kontak yang NO ini akan menjadi menutup (Close) dan mengalirkan atau menghubungkan arus listrik. Kontak NO digunakan sebagai penghubung atau menyalakan sistem circuit (Push Button ON).
- NC (Normally Close), merupakan kontak terminal dimana kondisi normalnya tertutup (mengalirkan arus listrik). Dan ketika tombol saklar push button

ditekan, kontak NC ini akan menjadi membuka (Open), sehingga memutus aliran arus listrik. Kontak NC digunakan sebagai pemutus atau mematikan sistem circuit (Push Button Off).

2.2.6 LED

Light Emitting Diode atau sering disingkat dengan LED adalah komponen elektronika yang dapat memancarkan cahaya monokromatik ketika diberikan tegangan maju. LED merupakan keluarga Dioda yang terbuat dari bahan semikonduktor. Warna-warna Cahaya yang dipancarkan oleh LED tergantung pada jenis bahan semikonduktor yang dipergunakannya. LED juga dapat memancarkan sinar inframerah yang tidak tampak oleh mata seperti yang sering kita jumpai pada *Remote Control TV* ataupun *Remote Control* perangkat elektronik lainnya.

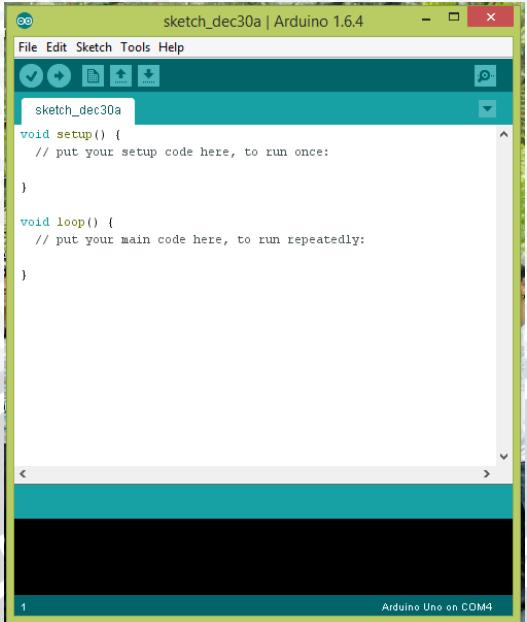


Gambar 2.9 LED

Bentuk LED mirip dengan sebuah bohlam (bola lampu) yang kecil dan dapat dipasangkan dengan mudah ke dalam berbagai perangkat elektronika. Berbeda dengan Lampu Pijar, LED tidak memerlukan pembakaran filamen sehingga tidak menimbulkan panas dalam menghasilkan cahaya.

2.2.7 Arduino IDE

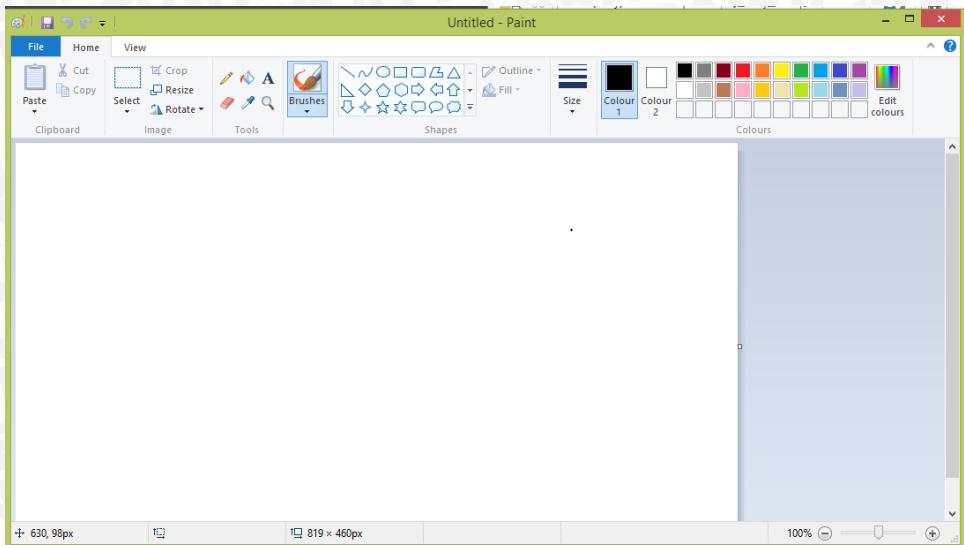
Arduino IDE adalah Software untuk pemrograman pada arduino yang menggunakan bahasa pemrograman dasar C++ yang telah dipermudah melalui library. Arduino menggunakan software processing yang telah digunakan untuk menulis program ke dalam arduino. Processing sendiri merupakan penggabungan antara bahasa C++ dan java. Software ini dapat di install di berbagai OS, seperti: Mac OS, Windows, LINUX.



Gambar 2.10 Arduino IDE

2.2.8 Paint

Paint adalah media gambar dalam pembuatan icon pada alat, software ini dapat digunakan untuk membuat gambar sederhana atau rumit, dapat pula berupa gambar-gambar hitam-putih atau warna, dan dapat disimpan sebagai file bitmap. software ini juga dapat digunakan sebagai media mencetak gambar, menggunakan sebagai latar belakang desktop, maupun copy-paste ke dokumen lain. software paint juga memiliki fitur edit gambar yang dapat digunakan untuk menambah pengalaman dalam edit foto.

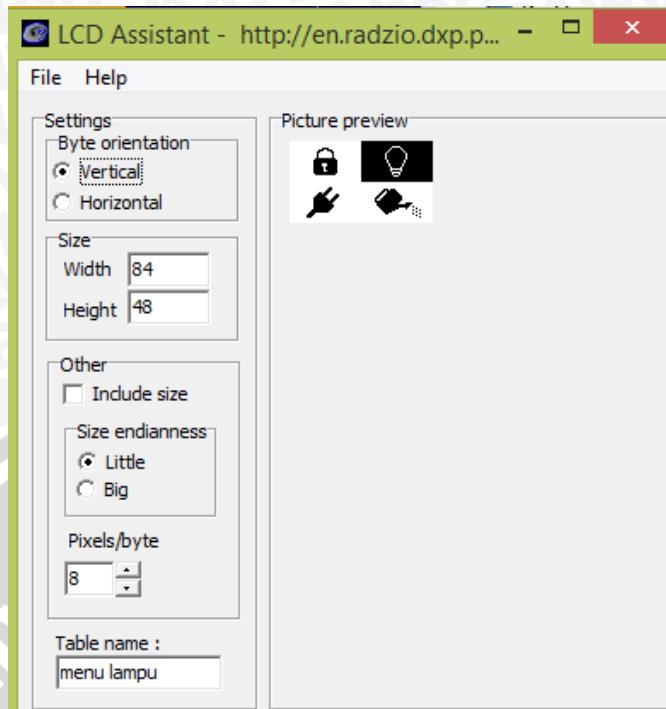


Gambar 2.11 Paint

paint dapat juga digunakan untuk bekerja dengan format gambar, seperti .jpg, .gif, atau .bmp dan juga dapat menyisipkan gambar Cat ke dokumen lain yang telah dibuat.

2.2.9 LCD Assistant

LCD Asisten adalah perangkat gratis untuk mengkonversi bitmap monokromatik untuk array data untuk mudah digunakan dengan program untuk embedded system dengan mikrokontroler dan grafis LCD menampilkan monokromatik seperti T6963C, KS0108, SED1335. Program membuat file untuk digunakan dengan compiler C: untuk AVR, ARM, PIC, 8051 dan mikrokontroler lainnya. Anda dapat menggunakan langsung dengan Arduino atau hardware lainnya yang berbasis C. Jika Anda menggunakan LCD grafis dan Anda ingin menampilkan gambar, Anda dapat menarik gambar dalam Anda terbaik editor grafis dan simpan di file *.bmp.

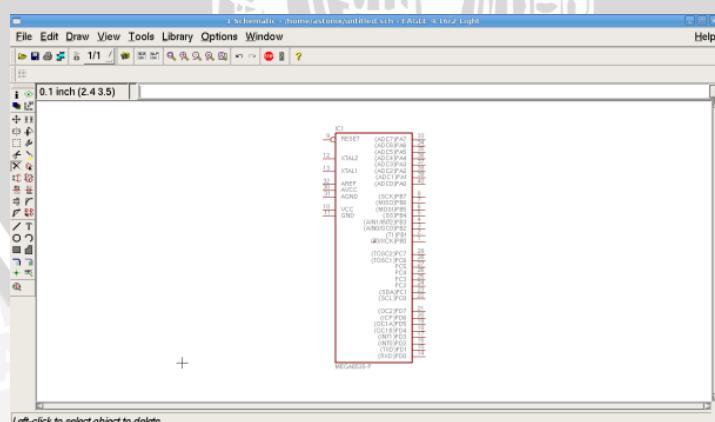


Gambar 2.12 LCD Assistant

2.2.10 Eagle

Eagle (Easily Applicable Graphical Layout Editor), merupakan sebuah aplikasi gratis untuk mendesain skematik Elektronika maupun *PCB (Printed Circuit Board)*. Dengan aplikasi EAGLE user bisa merancang, memodifikasi, dan mencetaknya untuk kemudian realisasikan ke dalam bentuk PCB. Aplikasi ini tersedia untuk Sistem Operasi GNU/Linux, Macintosh, maupun Ms.windows.

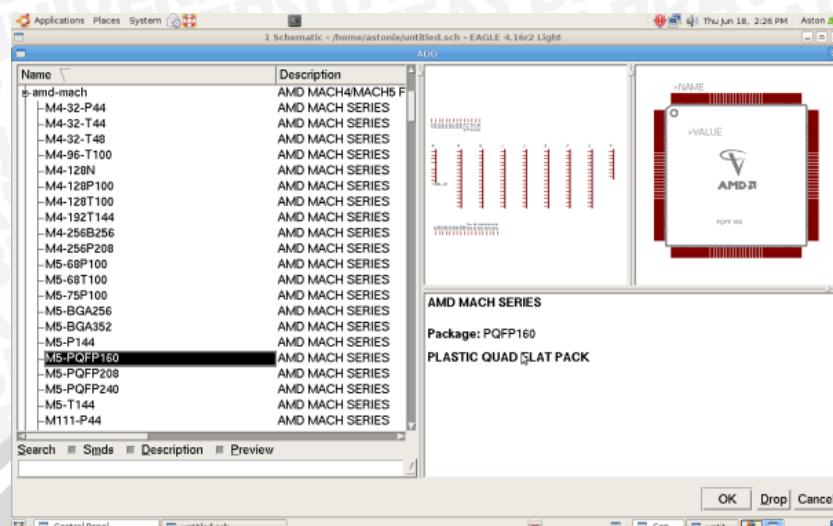
2.2.10.1 Schematic Editor



Gambar 2.13 Schematic Editor

Schematic editor Berfungsi untuk menggambar dan mengedit rangkaian skematik elektronika. Di sini pengguna bisa menggunakan komponen – komponen yang ada dalam Library "Add".

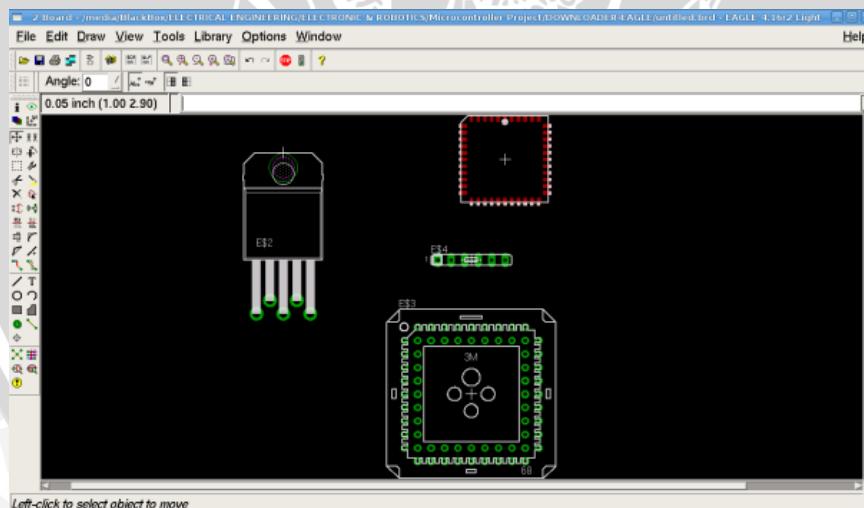
2.2.10.2 Library Tool



Gambar 2.14 Library Tool

Library Tool berfungsi untuk menambahkan berbagai komponen elektronika ke dalam schematic editor. Tersedia bermacam-macam komponen mulai dari Komponen Diskret (Resistor, kapasitor,dioda) sampai IC Mikrokontroler dan Mikroprosesor.

2.2.10.3 Eagle Board



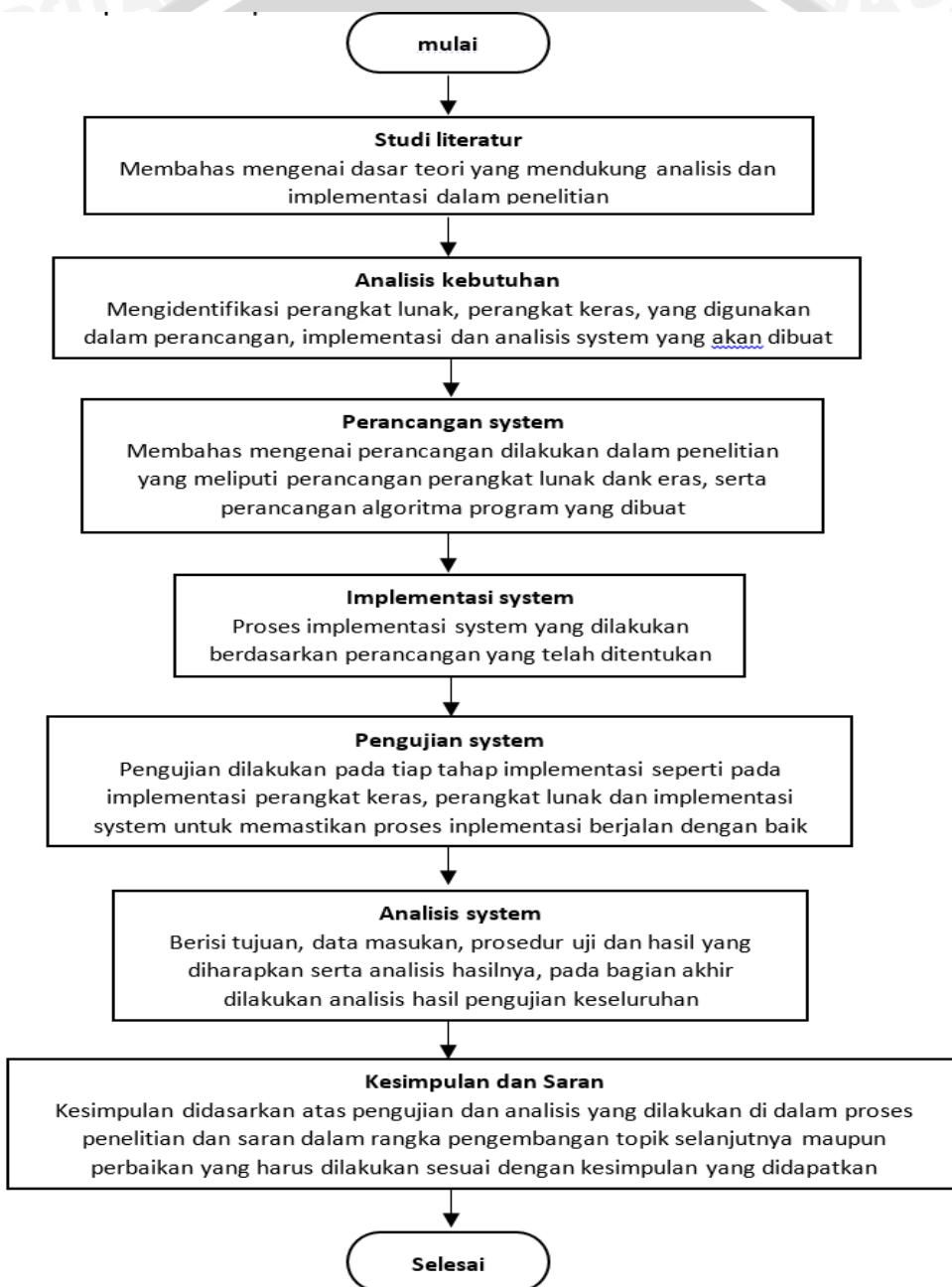
Gambar 2.15 Eagle Board

Eagle board Berfungsi untuk mentransfer rangkaian skematik ke rangkaian PCB. Di sini pengguna bisa menata komponen agar sesuai dengan yang diinginkan. Anda juga harus menggunakan fitur ini untuk me”routing” / membuat jalur pad PCB.

BAB 3 METODOLOGI

3.1 Metodologi Penelitian

Pada bab ini menjelaskan proses perancangan yang akan dilakukan dan dijelaskan dalam implementasi alat. Metode penelitian pada sistem ini akan dijelaskan langkah-langkah yang akan dilakukan dalam perancangan, implementasi, analisis dan pengujian dan penarikan kesimpulan. Untuk pengembangan sistem ini akan dituliskan pada saran. Bagan metode penelitian ini dapat dilihat pada Gambar



Gambar 3.1 Diagram Alir

3.1.1 Studi Literatur

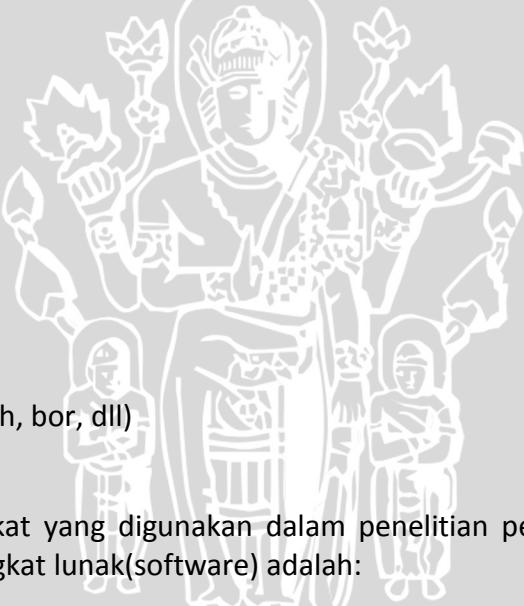
Pada studi literatur ini dijelaskan mengenai dasar teori yang mendukung analisis dan implementasi dalam penelitian desain *remote control* ini. Dalam studi literatur digunakan sebagai dasar implementasi pada desain sistem yang akan dibuat. Dengan menerapkan hal tersebut diharapkan desain sistem nantinya akan lebih akurat.

3.1.2 Analisis Kebutuhan Sistem

Analisis kebutuhan pada penelitian ini mempunyai tujuan untuk mendapatkan apa yang dibutuhkan. Analisis kebutuhan terbagi menjadi 2 (dua) kategori yaitu analisis kebutuhan perangkat keras (hardware) dan analisis kebutuhan perangkat lunak (software). Analisis kebutuhan sistem yang digunakan akan diimplementasikan pada sistem yang dibuat. Dengan adanya analisis kebutuhan sistem, diharapkan dalam hasil desain dan implementasi yang dibuat lebih akurat.

Adapun komponen yang digunakan dalam penelitian pembuatan *remote control* dari segi perangkat keras(hardware) adalah:

1. Arduino uno
2. Ardiuno nano
3. LCD 5510
4. LED
5. Switch
6. Laptop
7. Pcb
8. Toolkit(solder, timah, bor, dll)
9. Power



Sedangkan perangkat yang digunakan dalam penelitian pembuatan *remote control* dari segi perangkat lunak(software) adalah:

1. arduino IDE (pembuatan program arduino)
2. paint (membuat gambar)
3. LCD asisstant (mengubah gambar bitmap ke code biner)
4. eagle (pembuatan rangkaian)

3.1.3 Perancangan Sistem

Pada tahap perancangan sistem terdapat perancangan *remote* dan maket rumah sebagai output dari *remote*. Pada perancangan *remote* akan dibagi menjadi 3 yaitu pemograman pada LCD *remote*, pembuatan dan memprogram tombol, dan mendesain casing *remote* sehingga user dapat menggunakan layaknya *remote* pada dasarnya. Pada maket rumah yang berfungsi sebagai output *remote*, akan dirancang seperti layaknya rumah sehingga user dapat melihat fungsi *remote* yang

bekerja secara langsung. Untuk Desain *remote* ini menggunakan poin-poin yang ada pada prinsip *design rule* pada Interaksi Manusia dan Komputer (IMK) yang terdiri dari *learnability*, *flexibility*, dan *robustness*, dengan acuan dari ketiga aturan itu akan didapatkan desain *prototype* yang diharapkan. Untuk memahami aktifitas user terhadap alat, maka dibuatnya HTA yang berfungsi sebagai panduan terhadap apa saja yang dapat dilakukan oleh user.

3.1.4 Implementasi Sistem

Perancangan sistem direalisasikan dalam sebuah implementasi. Pada tahap ini terdiri dari 2 hal yang penting yaitu dari segi hardware dan software. Supaya hardware dan software ini bekerja dengan yang di inginkan, maka pada implementasi sistem terdapat beberapa tahapan yaitu:

3.1.4.1 Implementasi Perangkat Lunak/Software

Pada pembuatan sistem ini, terdapat beberapa perangkat lunak/software yang diperlukan untuk mendukung kinerja sistem. Perangkat lunak/ software yang dibutuhkan yaitu arduino IDE, paint, lcd_asisstant, dan eagle. Pada implementasi ini harus dapat menjalankan fungsi-fungsi antaralain:

1. Pada arduino IDE, software ini berfungsi sebagai pembuatan program utama yang nantinya akan diupload ke hardware arduino
2. Pada paint, software ini berfungsi untuk menggambar icon-icon dengan warna hitam-putih yang diperlukan *remote* dalam tampilan pada layar
3. Pada lcd_asisstant, software ini berfungsi sebagai mengubah gambar bitmap ke code biner
4. Pada eagle, software ini berfungsi untuk membuat rangkaian yang diperlukan untuk pembuatan *remote*

3.1.4.2 Implementasi Perangkat Keras/Hardware

Pada perangkat keras ini diimplementasikan pendukung jalan kerjanya sistem. Perangkat keras yang dibutuhkan yaitu arduino uno, arduino nano, LCD, tombol, dan LED. Pada implementasi ini harus dapat menjalankan fungsinya, antaralain:

1. Arduino uno, dalam sistem yang dibuat arduino uno akan digunakan sebagai maket rumah, yaitu menerima data dan mengaktifkan LED
2. Arduino nano, dalam sistem ini arduino nano akan bekerja sebagai *remote*, yang kinerja utamanya mengirim data yang ingin di kirim oleh user
3. LCD, berfungsi sebagai tampilan untuk *remote*
4. button, berfungsi sebagai tombol *remote*
5. LED, berfungsi sebagai indikator pada maket rumah



3.1.5 Pengujian dan Analisa Sistem

Pengujian pada penelitian ini dibagi menjadi 2 (dua) tahap yaitu verifikasi dan pengujian usabilitas alat. Verifikasi ini berisi tentang pengujian hardware. Pengujian ini dimulai dari pengujian tampilan LCD dengan memasukan program ke arduino. Selanjutnya pengujian pada tombol untuk input ke arduino sesuai dengan output ke LCD yang di inginkan. Setelah itu pengujian koneksi dari pengiriman data arduino *remote* sampai di terimanya data oleh arduino maket. Setelah data antar arduino sesuai, selanjutnya pengujian LED pada maket yang menyala sesuai dengan data yang diterima.

Untuk tahap pengujian alat, *remote* yang telah memenuhi verifikasi akan diujikan kepada responden dengan menggunakan sampling. Sample adalah sebagian dari jumlah dan karakteristik yang dimiliki oleh populasi atau bagian kecil dari anggota populasi yang diambil menurut prosedur tertentu sehingga dapat mewakili populasinya. Karena *sampling* yang dilakukan terlalu besar sehingga tidak mungkin mempelajari seluruh populasi, random sampling adalah salah satu teknik samping dengan pengambilan sample dilakukan secara acak, tanpa memperhatikan strata yang terdapat dalam populasi tersebut. Cara ini dilakukan jika anggota populasi dianggap homogen. Roscoe (1975) juga memberikan beberapa panduan untuk menentukan ukuran dari sampel yaitu ukuran sampel lebih dari 30 dan kurang dari 500 adalah ukuran tepat untuk kebanyakan penelitian.

Untuk penelitian sederhana, penelitian yang sukses adalah dengan ukuran sampel kecil antara 10 sampai dengan 20. Sehingga ukuran sample yang digunakan 30 responden dengan klasifikasi 8 responden mahasiswa dengan jurusan T.I (Teknologi Informasi), 7 responden dosen dengan jurusan T.I, 8 responden mahasiswa dengan jurusan non-T.I, 7 responden dosen dengan jurusan non-T.I. Klasifikasi ini ditujukan untuk mengetahui perbedaan tingkat usabilitas antara responden yang memiliki dasar ilmu T.I dengan responden dengan dasar ilmu non-T.I. Pengujian dilakukan ada 2, yaitu dengan observasi dan kuisioner. Pada pengujian menggunakan observasi dilakukan dengan menghitung waktu yang dibutuhkan oleh responden dalam melaksanakan suatu skenario yang telah ditetapkan. Pengujian observasi juga melihat konsep interaksi dan bantuan yang sudah diimplementasikan apakah mampu meningkatkan usabilitas alat. Pada pengujian melibatkan pengguna menggunakan questioner, responden dipersilahkan untuk mengisi questioner yang telah disediakan untuk menilai *remote control* ini. Pengujian verifikasi dan usabilitas ini akan mendapatkan hasil berupa data yang dianalisis untuk menentukan apakah sistem sudah berjalan baik sesuai dengan yang diharapkan.



BAB 4 REKAYASA KEBUTUHAN

Dalam bab ini menjelaskan persyaratan minimal yang harus dipenuhi untuk perancangan hingga implementasi. Dengan harapan perancangan dan implementasi program ini bisa berjalan dengan baik.

4.1 Analisa Kebutuhan

Pada analisa kebutuhan menjelaskan tentang kebutuhan *user* yang diperlukan dalam mengendalikan alat elektronik di rumah. Kebutuhan tersebut dapat dijelaskan pada tabel berikut.

Tabel 4.1 Kebutuhan

No	Kebutuhan
1	Mengendalikan lampu
2	Mengunci pagar dan pintu depan
3	Mengendalikan perangkat elektronik
4	Mengendalikan penyiram tanaman

Dengan masalah tersebut didapatkan system yang menampilkan kebutuhan tersebut dalam *remote* yang di jelaskan pada tabel spesifikasi kebutuhan

Tabel 4.2 Spesifikasi Kebutuhan

No	Nama kebutuhan	Deskripsi kebutuhan	Aktor
1	<i>remote</i> menampilkan perangkat	dapat menu Pada menu terdapat kunci, lampu masing-masing ruangan, elektronik, dan penyiram tanaman	User
2	<i>remote</i> dapat mengendalikan perangkat	Tiap perangkat dapat dikendalikan menyala atau mati	User

4.2 System requirement

Berdasarkan persyaratan yang dikemukakan sebelumnya, didapatkan bahwa *remote Control* ini memiliki kebutuhan sebagai berikut

1. *remote control* dibuat menggunakan prinsip desain interaksi

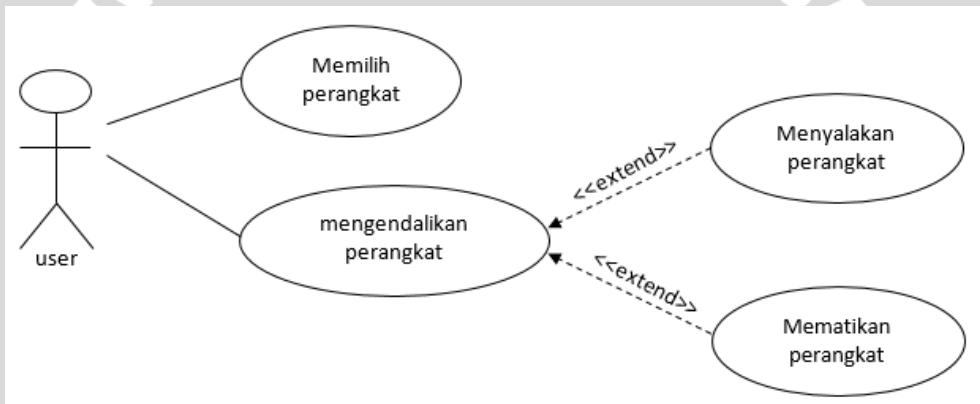


2. *remote control* berfungsi sebagai pengontrol maket rumah antar user dan maket
3. *remote control* yang digunakan user dapat mengakses fitur-fitur yang telah di sediakan
4. *remote control* menggunakan tombol yang sedikit

4.3 Desain Model

4.3.1 Use Case

Use case digunakan sebagai proses analisis dan memahami interaksi antara *user* dan *remote*. Dalam hal ini user dapat melakukan memilih perangkat dan mengendalikan perangkat.



Gambar 4.1 *Use Case Remote*

4.3.2 Identifikasi Use Case

Memilih perangkat merupakan kegunaan dimana user dapat memilih perangkat yang ingin dipilih.

Mengendalikan perangkat merupakan pengendalian dengan cara menyalakan dan mematikan perangkat.

4.3.3 Identifikasi Aktor

Actor dari use case ini adalah user, yang mana user dapat mengatur perangkat dengan menggunakan *remote*.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan

5.1.1 Desain Interaksi Pada *Remote Smart Home*

Tahapan perancangan *remote smart home* dimulai dengan menentukan langkah awal yang harus di kerjakan. Berdasarkan bagan desain interaksi yang ada pada sub-bab 2.2.1 desain interaksi, langkah pertama yang harus dilakukan adalah menentukan “*what is wanted*” atau bisa di artikan sebagai seperti apa, untuk siapa, dan dimana penggunaannya. Selanjutnya ialah melakukan analisis berupa skenario dari *remote smart home*. Setelah itu menentukan desain yang beracuan dengan 3 prinsip desain interaksi, kemudian dilanjutkan dengan pembuatan *prototype*. Setelah prototype jadi kemudian dilakukan pengujian dan di analisa untuk mengetahui kekurangan atau kesalahan dari prototype yang akan disampaikan di bab 7 sebagai saran dari *remote* sekaligus perbaikan yang bisa dilakukan untuk menambah usabilitasnya. Perancangan system yang berdasarkan prinsip desain interaksi diharapkan bisa meningkatkan usabilitas *remote smart home*.

5.1.1.1 *What Is Wanted*

Untuk memenuhi kriteria dari prinsip desain interaksi maka langkah pertama yang dilakukan adalah menentukan siapa yang akan menggunakan dan dimana penggunaannya. Berikut ini penjelasan what is wanted dari *remote*:

- Penggunaan : *remote* ini digunakan untuk mengontrol perangkat smart home.
- Tempat penggunaan: penggunaannya di dalam rumah.
- Kriteria pengguna: *remote* ini ditujukan pada pengguna umum yang mana pengguna yang memiliki pengetahuan dasar ilmu Teknologi Informasi (TI) maupun tidak.

5.1.1.2 Penggunaan Alat (*skenario*)

Cara penggunaan alat dapat ditinjau dari 2 sisi, yaitu pada sisi user dan alat(*remote*). Sisi user hanya mengetahui tampilan luar tanpa harus mengetahui bagaimana proses jalannya system di dalamnya. Skenario pada sisi user menunjukkan perancangan alur cerita dari bagaimana langkah demi langkah user akan menggunakan *remote* tersebut. Sedangkan pada sisi alat merupakan penjelasan skenario langkah demi langkah bagaimana kerja alat ketika di operasikan.

Sisi user

1. memilih perangkat pada menu layar *remote*



2. menyalakan dan mematikan perangkat dengan memilih pilihan “nyala” atau “mati”
3. melihat perubahan indikator pada gambar perangkat di *remote* sebagai feedback pilihan yang dipilih

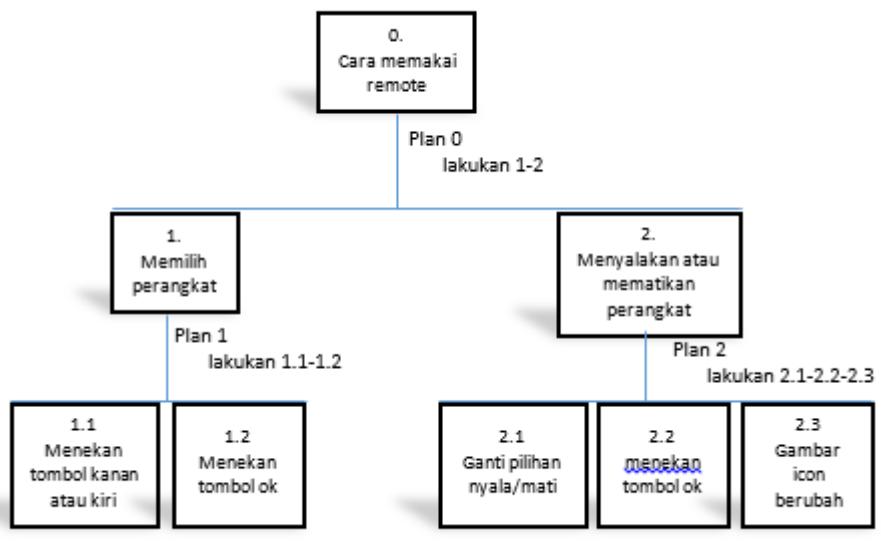
Sisi alat

1. menampilkan ragam perangkat smart home pada layar LCD *remote control*
2. menampilkan pilihan perangkat yang dipilih user melalui push-button
3. ketika user mematikan atau menyalakan perangkat melalui push-button, maka layar LCD menampilkan perubahan gambar/icon berdasarkan inputan
4. arduino di *remote control* akan mengirimkan data perintah pengendalian ke arduino di maket
5. kondisi perangkat dari maket akan berubah sesuai dengan input user

5.1.1.3 *Hierarchical Task Analysis*

Fungsi dari *task analysis* ini untuk menyediakan informasi dalam pengambilan keputusan desain serta mengevaluasi desain dari system. HTA untuk masing-masing perangkat adalah sama.

Pada HTA ini terdapat 2 plan utama, yaitu plan 1 yang memilih perangkat dan plan 2 yang menyala-matikan perangkat. Berikut adalah HTA dari *remote control smart home* yang dapat dilihat pada gambar 5.1



Gambar 5.1 HTA *Remote Control Smart Home*

5.1.1.4 Penerapan Prinsip Desain Interaksi

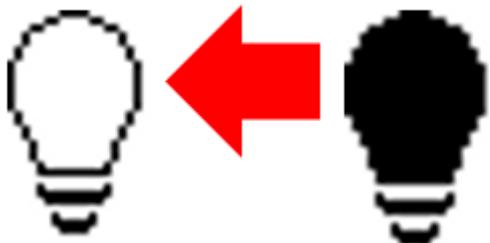
Tujuan penerapan Prinsip Desain Interaksi adalah untuk memaksimalkan usabilitas sistem yang nantinya akan digunakan oleh user. Sistem dikatakan bekerja dengan baik jika user dapat menggunakan system tersebut secara optimal, karena semua kemampuan dalam system dapat dimanfaatkan dengan baik(dix alan 2004). Sedangkan untuk mencapai sebuah tingkat usabilitas yang maksimal, dibutuhkan 3 prinsip dalam perancangannya seperti yang telah di jelaskan pada sub-bab 2.2.1.3 desain. Prinsip-prinsip tersebut kemudian diterapkan dalam proses untuk mendesain *remote smart home* yaitu:

- learnability*: user pemula mampu mempelajari sistem dan memanfaatkan sistem secara optimal. Prinsip ini terbagi menjadi 5 bagian, yaitu:
 - *predictability*: penerapan prinsip ini pada gambar icon yang dipakai, sehingga user dapat langsung mengetahui gambar icon tersebut (penerapan prinsip ini adalah struktur dari *remote* yang memiliki tombol, sehingga user dapat melihat secara langsung dan mengerti bahwa itu sebuah *remote*)



Gambar 5.2 Icon Perangkat

- *synthesizability*: user dapat melihat feedback langsung dari *remote* seperti lampu berwarna hitam ketika memilih perintah mati dan lampu berwarna putih ketika memilih perintah nyala



Gambar 5.3 Penerapan *Synthesizability*

- *familiarity*: memakai icon-icon tombol yang mudah dipahami dan sering dilihat oleh user seperti back, arrow kiri kanan, rumah untuk halaman awal atau home, tulisan OK untuk menyetujui pilihan



Gambar 5.4 Penerapan *Familiarity*

- *generalizability*: membuat bentuk desain *remote* ini sama halnya bentuk desain *remote* pada umumnya. Posisi dan tata letak kiri kanan, tombol dikumpulkan berupa lingkaran, ok di tengah itu juga sesuai dengan button2 umumnya.



Gambar 5.5 Penerapan *Generalizability*

- *Cosistensy*: menggunakan perubahan icon gambar perangkat pada layar dimana tanpa ikon kilat untuk kondisi mati, dan dengan ikon kilat untuk kondisi hidup.



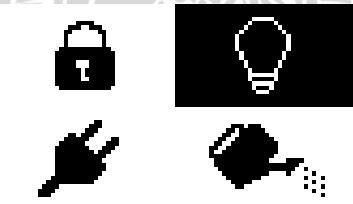
Gambar 5.6 Penerapan *Cosistensy* Icon

lalu semua pilihan bergeser dari kiri ke kanan.



Gambar 5.7 Penerapan *Cosistensy* Geser

- b. *flexibility*: system yang dianggap memenuhi usability diharapkan dioperasikan dengan prosedur tidak kaku.
- *Dialogue initiative*: user dapat memilih perangkat dan menyalakan/ mematikan perangkat dengan bebas. Dengan ini pengguna mempunyai kendali penuh terhadap perangkat pada rumah.
- c. *Robustness* : Prinsip ini diartikan sebagai kehandalan sebuah sistem dalam mencapai tujuan, khususnya dari sudut pandang pengguna. Berikut kriteria pada prinsip robustness:
- *Observability*: pada prinsip ini terletak pada menu utama yang ditampilkan pada LCD *remote*. Pada menu tersebut user dapat mencoba dan melihat secara langsung dari perangkat yang dapat dikendalikan oleh user.



Gambar 5.8 Penerapan *Observability*

Kondisi nyala dan mati, icon di layar juga berubah dan nama pilihan yang dipilih muncul di bagian atas, seperti konsep windows explorer

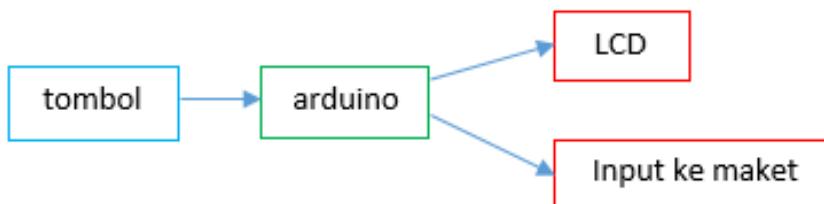
- *Responsiveness*: prinsip ini digunakan pada system *remote* sehingga pengguna memilih perangkat maupun mematikan atau menyalakan perangkat tanpa adanya kesalahan.
- *Task Conformance*: prinsip ini digunakan pada system *remote* sehingga perintah yang akan diberikan pengguna sesuai dengan yang diharapkan pengguna yakni mengendalikan perangkat smart home

5.1.2 Perancangan Perangkat Keras

Terdapat 2 bagian dalam perancangan perangkat keras ini, yaitu dari sisi *remote* smart home dan maket.

5.1.2.1 Perancangan *Remote Smart Home*

1. Diagram blok



Gambar 5.9 *Diagram Blok*

Perancangan *remote smart home* ini terdiri dari 3 blok, yaitu blok input, mikrokontroller dan output yang dibedakan dengan warna, seperti gambar diagram blok yang ditunjukan pada gambar 5.

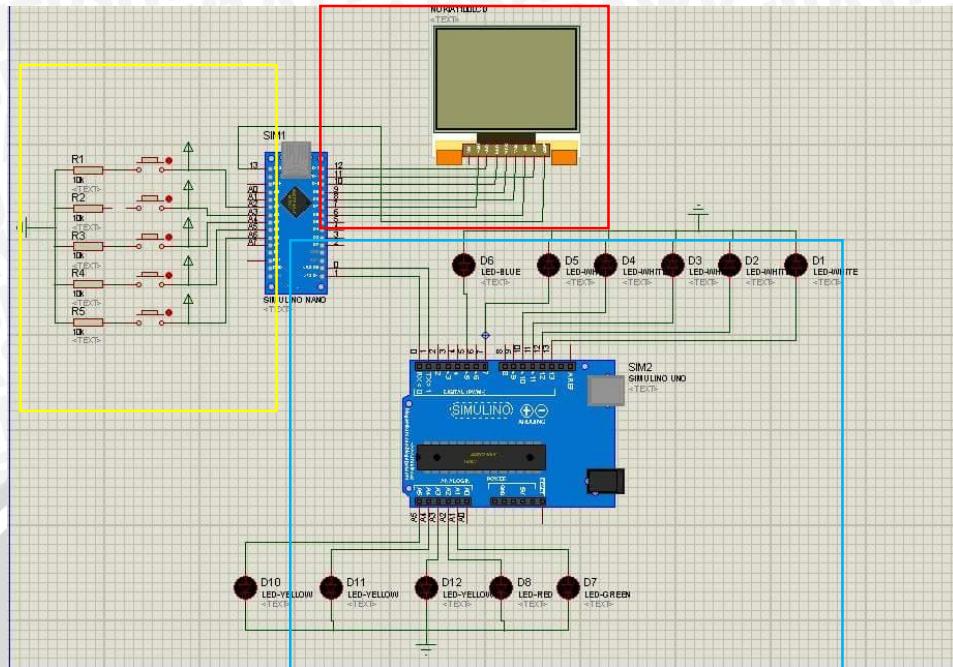
Pada blok berwarna biru adalah blok input yang terdiri atas tombol dimana tombol ini berfungsi sebagai input sekaligus pilihan untuk sistem dalam *remote* tersebut. Seperti layaknya tombol pada umumnya, tombol akan mengirimkan input sesuai dengan tombol yang ditekan.

Pada blok warna hijau adalah blok mikrokontroller yang terdiri atas arduino. Pada arduino ini berisikan dengan program yang mana di program tersebut terdapat deklarasi variable input dan output serta coding sistem.

Pada blok warna merah adalah blok output yang terdiri atas LCD dan input ke maket. Pada LCD akan menampilkan tampilan yang telah di program, dan menampilkan tampilan sesuai dengan input. Input ke maket adalah data yang akan dikirimkan oleh arduino, yang mana data tersebut dikirim melalui pin transceiver ke maket.

2. Rangkaian keseluruhan

Pada gambar 5.10 Merupakan rangkaian keseluruhan dari rangkaian *remote*. Pada blok warna kuning adalah bentuk rangkaian tombol ke arduino, blok warna merah adalah bentuk dari rangkaian LCD ke arduino. blok warna biru adalah bentuk dari rangkaian maket.



Gambar 5.10 Rangkaian Keseluruhan

3. Perancangan Rangkaian tombol ke arduino

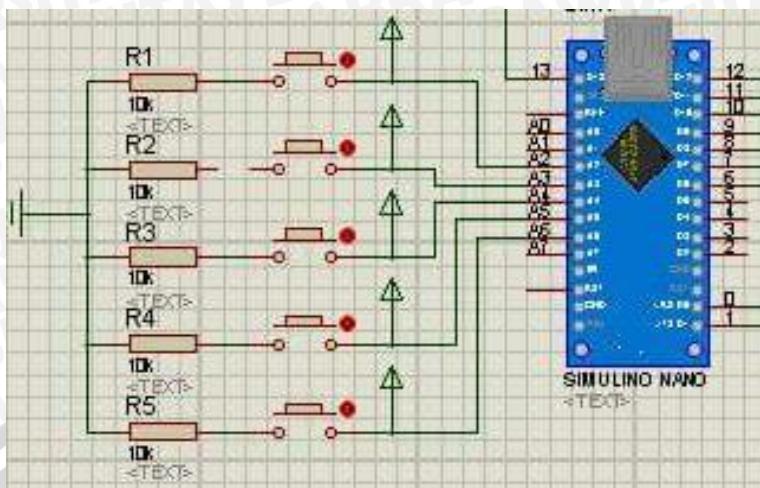
Lima tombol yang diperlukan dalam *remote control* ini adalah sebagai berikut:

- ➡ = Tombol Kembali
- OK = Tombol OK
- ➡ = Untuk Geser Pilihan Ke Kanan
- ➡ = Untuk Geser Pilihan Ke Kiri
- ➡ = Tombol Kembali Ke Menu Awal

Gambar 5.11 Icon Tombol

Dari 5 tombol tersebut dirangkai menjadi rangkaian 5 buah push button sesuai desain *remote* sehingga didapatkan bentuk rangkaian yang ditunjukkan pada gambar 5.12





Gambar 5.12 Rangkaian Tombol

Pada rangkaian tersebut masing-masing push button di sambungkan ke arduino, dimana salah satu kaki push button disambungkan ke pin ground pada arduino, dan kaki lainnya disambungkan ke pin input arduino yang mana di tengah sambungannya terdapat rangkaian resistor yang terhubung dengan vcc. Pada rangkaian tersebut dengan maksud bila ditekan maka bernilai 0, jika tidak ditekan maka bernilai 1.

4. Perancangan Rangkaian dan tampilan LCD ke arduino

Pada tampilan LCD telah menggunakan icon-icon sesuai dengan prinsip *predictability* yang mana user dapat mengenali langsung gambar Icon-icon yang digunakan sebagai berikut

Tabel 5.1 Perancangan Tampilan Icon

gambar	keterangan
	Icon gembok kunci
	Icon gembok buka
	Icon elektronik
	Icon lampu nyala

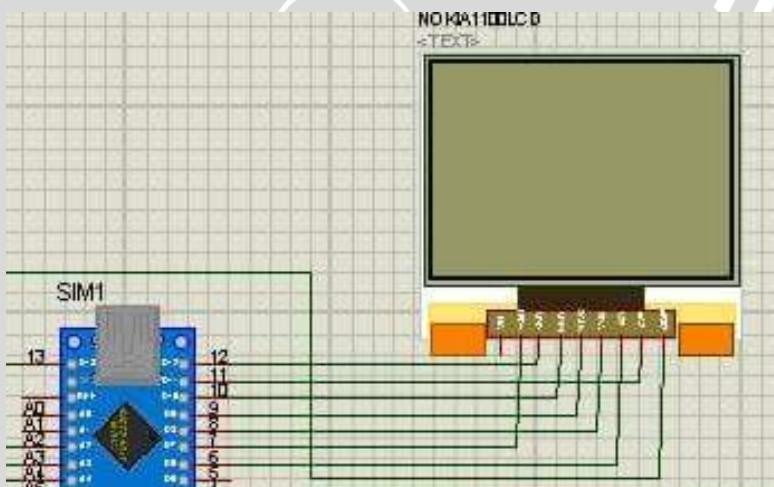
	Icon lampu mati
	Icon penyiram tanaman
	Icon TV mati
	Icon TV menyala
	Icon AC mati
	Icon AC menyala
	Icon penanak nasi mati
	Icon penanak nasi menyala

Pada perancangan LCD, pin pada LCD disambungkan langsung sesuai dengan pin pada arduino yang disesuaikan. pin tersebut sebagai berikut:

Tabel 5.2 Perancangan Pin LCD

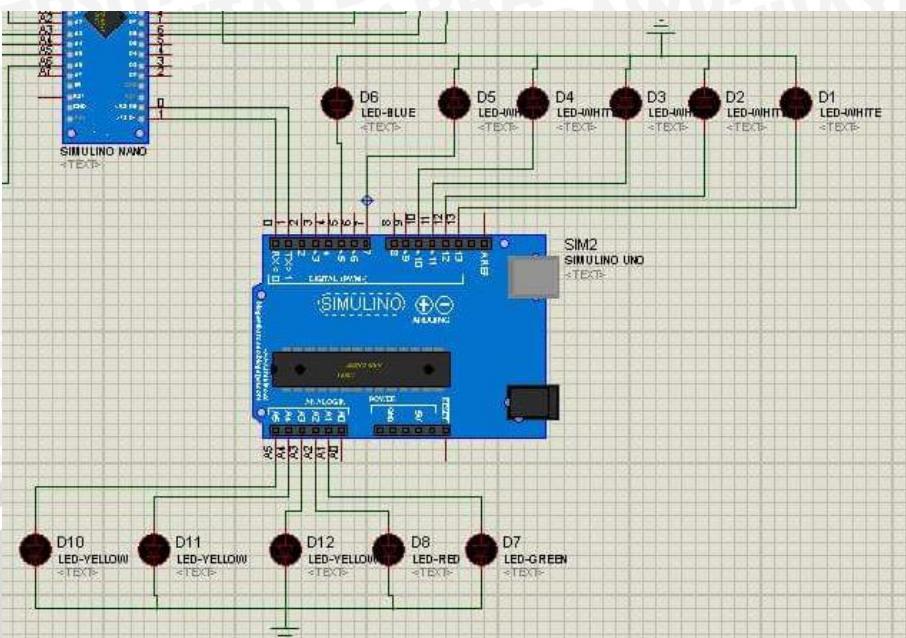
Pin LCD	Pin arduino
RST	D10
CE	D7
DC	D12
DIN	D11
CLK	D13
VCC	V3.3
BL	5V
GND	GND

Sehingga di dapatkan rangkaian seperti pada gambar 5.13

**Gambar 5.13 Rangkaian LCD**

5.1.2.2 Perancangan Maket

Untuk perancangan maket menggunakan arduino uno yang berfungsi sebagai pengatur indikator lampu pada maket sesuai dengan masukkan pada *remote*. Input dari *remote* akan diterima oleh pin receiver oleh arduino uno yang akan diproses sesuai dengan lampu LED mana yang dinyalakan. Berikut rangkaian maket pada Gambar 5.14.



Gambar 5.14 Rangkaian Maket

Untuk memenuhi implementasi perancangan maket di dapatkan gambar maket yang dipakai seperti pada gambar 5.15



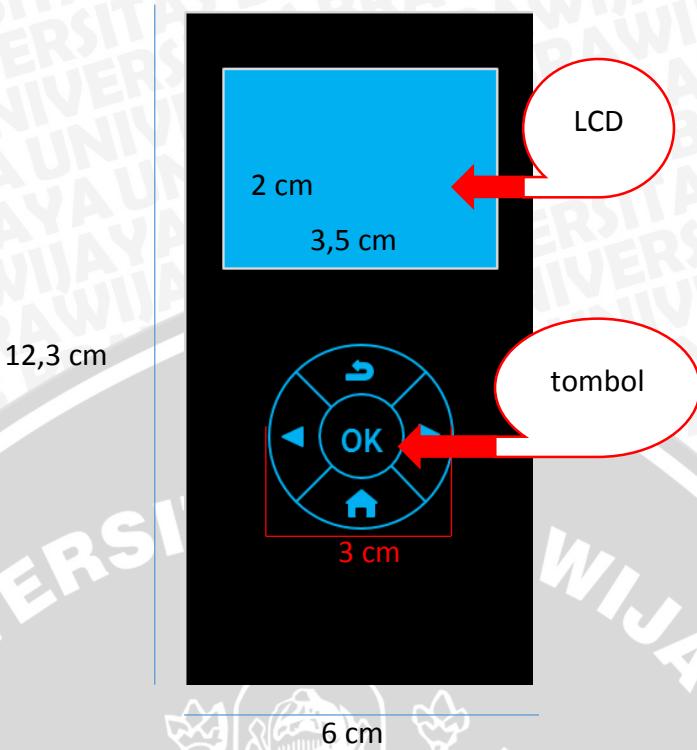


Gambar 5.15 Rancangan Maket

5.1.2.3 Perancangan Case Remote

Pada perancangan case ini di desain dengan bentuk umum seperti bentuk desain *remote* pada dasarnya. Untuk memudahkan dalam perancangan di dapatkan desain, case di ukur berdasarkan komponen *remote* yaitu LCD, arduino nano, dan tombol. Sehingga di dapatkan ukuran ringkas sebagai berikut:

- a. panjang x lebar x tinggi= 12,3cm x 6cm x 1,4cm
- b. ukuran lubang diameter tombol = 3cm
- c. ukuran lubang LCD = 3,5cm x 2cm



Gambar 5.16 Rancangan Remote

Konsep peletakan setiap bagian case *remote* ini mempunyai beberapa tujuan yaitu:

- a. bentuk *remote* di desain dengan bentuk umum seperti bentuk desain *remote* pada dasarnya sesuai dengan prinsip generalizability.
- b. Letak tombol dibawah LCD supaya mudah digunakan sama seperti desain yang memiliki komponen LCD seperti HP, *remote AC*, dan lainnya
- c. karena tombol yang digunakan 5 buah, bentuk tombol di desain bulat seperti tombol pada IPOD agar mudah untuk digunakan
- d. icon pada tombol diletakkan berdasarkan fungsi umum. Tombol “kanan” diletakkan di sebelah kanan, tombol “kiri” diletakkan di sebelah kiri, tombol “ok” diletakkan ditengah, tombol “home” diletakkan di bawah, dan yang terakhir tombol “kembali” di letakkan diatas.
- e. Sisa space di bawah untuk genggaman

5.1.3 Perancangan Perangkat Lunak

perancangan lunak pada *remote* di gunakan untuk program yang dimasukkan ke arduino berjalan dengan baik. Proses pembuatan program ini menggunakan bahasa C pada aplikasi arduinolDE. Dengan menjalankan perintah upload pada aplikasi tersebut dan akan muncul notifikasi “done uploading” yang berarti telah

sukses memasukkan program ke arduino. Langkah yang harus di lakukan sebagai berikut:

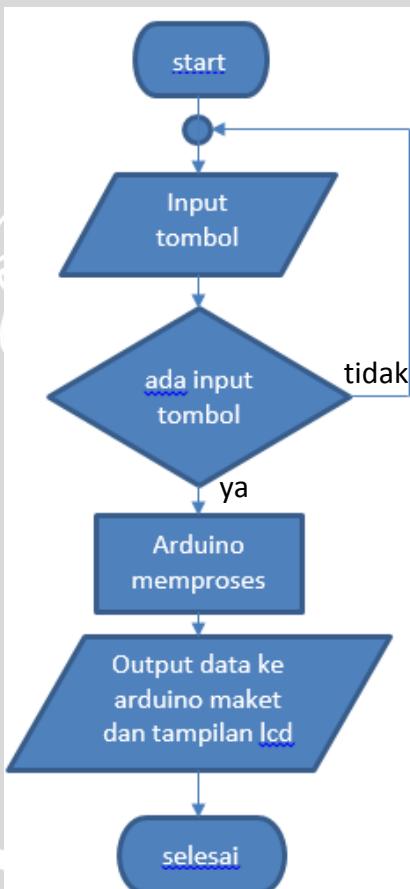
- a. mendefinisikan variable
- b. mendefinisikan library
- c. membuat program

5.1.4 perancangan buku manual

perancangan buku manual di tujuhan untuk membuat suatu bantuan dalam bentuk fisik yang berguna untuk acuan bagi user untuk menggunakan remote ini.

5.1.5 Flowchart System

flowchart system yang ditanamkan pada *remote* dapat dilihat pada gambar 5.17 sesuai dengan diagram alir, maka system ini dimulai dengan memasukan input tombol,



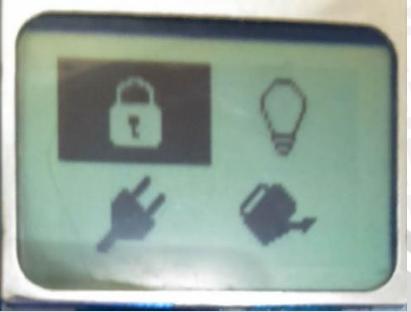
Gambar 5.17 Flowchart

5.2 Implementasi

5.2.1 Implementasi Perancangan Desain Interaksi

1. layar home

terdiri dari icon lampu, kunci, elektronik, dan penyiram,



Gambar 5.18 Implementasi Menu

2. layar lampu

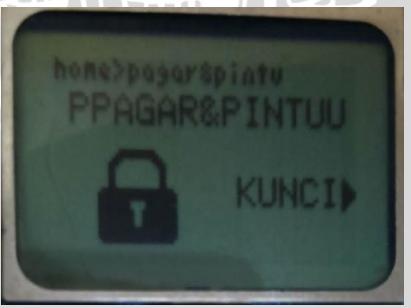
berisi tampilan icon lampu yang diatas layar terdapat pilihan lampu



Gambar 5.19 Implementasi Menu Lampu

3. layar kunci

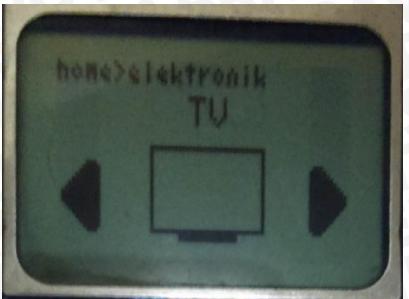
berisi tampilan layar menu kunci



Gambar 5.20 Implementasi Menu Kunci

4. layar elektronik

berisi tampilan layar pilihan elektronik



Gambar 5.21 Implementasi Menu Elektronik

5. layar penyiram
- berisi tampilan layar pilihan penyiram tanaman



Gambar 5.22 Implementasi Menu Penyiram Tanaman

5.2.2 Implementasi Case

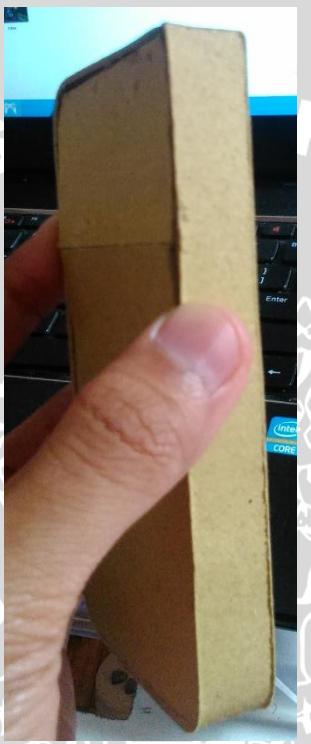
Setelah implementasi perangkat keras telah selesai, pada implementasi case dibuat case yang sesuai dengan perancangan case *remote* dengan menggunakan karton untuk memudahkan proses pembuatan. Berikut ini adalah gambar case *remote*.



Gambar 5.23 Remote Tampak Depan



Gambar 5.24 Remote Tampak Samping



Gambar 5.25 Remote Tampak Belakang

5.2.3 Implementasi Perangkat Keras

5.2.3.1 Implementasi Perancangan Remote

Untuk rangkaian tombol, proses implementasi yang dapat dilakukan adalah menyambungkan tombol tersebut ke arduino nano. Dengan masing-masing kabel pada rangkaian tombol di pasang ke pin 2, 3, 4, 5 dan 6. Pemasangan kabel rangkaian tombol ke pin arduino dapat dilihat pada gambar 5. Beri foto asli yang tidak ada case nya, beri keterangan bagian2nya mana arduino, mana button dll



Gambar 5.26 Implementasi *Remote Control*

5.2.3.2 Implementasi Perancangan Maket

Untuk rangkaian maket, proses implementasi yang dapat dilakukan adalah menyambungkan LED ke arduino dan di implementasikan ke maket. Implementasi perancangan maket dapat dilihat pada Gambar 5.27

UNIVE
RSIT
R
S
TAS
RSITAS
VERSIT
UNIVE
IJAYA UNI
BRAWI
SITAS BRA
VERSITAS BRA
AUNIVERSIT
AYA UNIVE
WIJAYA UNI
RAWIJAYA
S BRA
ITAS BRA
ERSITAS BRA
NIVERSITAS BRA
AUNIVERSITAS BRA
AYA UNIVE
WIJAYA UNI
RAWIJAYA
S BRA
ITAS BRA
ERSITAS BRA
NIVERSITAS BRA
AUNIVERSITAS BRA
AYA UNIVE
WIJAYA UNI
RAWIJAYA

Gambar 5.27 Implementasi Maket

5.2.4 Implementasi Perangkat Lunak

1. Implementasi Perangkat Lunak *Remote*

Setelah pemasangan tombol, pada arduino nano dibuat program untuk merealisasikan fungsi tombol. Yang pertama deklarasi semua tombol sesuai pemasangan kabel tombol ke arduino nano. Lalu buat kondisi dimana:

Tabel 5.3 Implementasi Pin Tombol

Pin	Fungsi
2	Sebagai tombol “kanan”
3	Sebagai tombol “kembali”
4	Sebagai tombol “kiri”
5	Sebagai tombol “OK”
6	Sebagai tombol “home”

Proses pembuatan program tombol ditulis di aplikasi arduino IDE sebagai berikut

```
int a=2; // tombol kanan  
int b=5; //tombol ok  
int c=4; // tombol kiri  
int d=3; // tombol kembali  
int e=6; // tombol home
```

Pseudo Code menu

2. LCD

Untuk rangkaian LCD, proses implementasinya adalah dengan menyambungkan pin LCD ke arduino nano yang telah di jelaskan pada perancangan LCD ke *remote* pada sub-bab 5.1.2.1 poin ke 4. Pin tersebut disambung dengan menggunakan kabel. Pemasangan LCD pada arduino nano dapat di lihat pada gambar 5.

Program untuk system LCD sebagai berikut:

a. gambar menu

- inisialisasi icon pada menu

tiap gambar akan di inisialisasikan menjadi variable yang berguna memudahkan pemanggilan dalam proses pembuatan program

```
extern uint8_t menugerbang[];  
extern uint8_t menulampu[];  
extern uint8_t menuelektro[];  
extern uint8_t menuair[];  
extern uint8_t menugerbangbuka[];  
extern uint8_t menuairmati[];  
  
extern uint8_t menugerbanginvert [];  
extern uint8_t menulampuinvert [];  
extern uint8_t menuelektroinvert [];  
extern uint8_t menuairinvert [];  
extern uint8_t menugerbangbukainvert[];  
extern uint8_t menuairmatiinvert[];
```

Pseudo Code menu

- menu gembok




```
const uint8_t menuelektronik [] PROGMEM= {0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x80, 0xC0, 0xC0, 0x80, 0xC0, 0xE0, 0x70, 0x38, 0x18,
0x00, 0x00, 0x80, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x0FC, 0xF8, 0x0FC, 0x6E, 0x07,
0x03, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x18, 0x1C, 0x1E, 0x0F, 0x07, 0x03, 0x03,
0x03, 0x03, 0x03, 0x03, 0x03, 0x03, 0x01, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
```

Pseudo Code menu

- menu penyiram tanaman

ketika gambar penyiram tanaman telah di gambar melalui aplikasi Paint, file tersebut di convert menjadi code biner dengan aplikasi lcd_asistant sehingga di dapatkan code biner di bawah ini

```
const uint8_t menuair [] PROGMEM= {0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0xC0, 0x60, 0x30, 0x90,
0xF0, 0x60, 0x30, 0x10, 0x30, 0x60, 0xC0, 0x80, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
```

Pseudo Code menu

- program utama menu

setelah mendapatkan code biner tiap icon menu dan telah di definisikan menjadi variable, di bawah ini adalah pseudo code untuk menampilkan tampilan menu

```
myGLCD.clrScr();
if(nilaiinpmbuka==0){
    myGLCD.drawBitmap(0, 0, menugerbanginvert, 42, 24);
}
```

```
if(nilaipinbuka==1){  
    myGLCD.drawBitmap(0, 0, menugerbangbukainvert, 42, 24);  
}  
myGLCD.drawBitmap(42, 0, menulampu, 42, 24);  
myGLCD.drawBitmap(0, 24, menuelektro, 42, 24);  
myGLCD.drawBitmap(42, 24, menuairmati, 42, 24);  
myGLCD.update();
```

Pseudo Code menu

b. tampilan pagar dan pintu

- inisialisasi icon gembok

mendeklarasikan sebuah variable untuk icon gembok yang berguna untuk mempermudah pemanggilan

```
extern uint8_t GERBANG7buka[];
```

Pseudo Code menu

- gambar icon gembok

ketika gambar penyiram tanaman telah di gambar melalui aplikasi Paint, file tersebut di convert menjadi code biner dengan aplikasi lcd_asistant sehingga di dapatkan code biner di bawah ini

```
const uint8_t menugerbangbuka [] PROGMEM= {0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0xE0, 0x30, 0x10, 0x18, 0x08, 0x08, 0x18, 0x10,  
0x30, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xF0,  
0xF8, 0xFF, 0xF8, 0xF8, 0x38, 0x38, 0xF8, 0xF8, 0xF8, 0xF9,  
0xF8, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0x1F, 0x1F,  
0x1F, 0x1F, 0x1F, 0x18, 0x1B, 0x1F, 0x1F, 0x1F, 0x1F, 0x1F, 0x0F,  
0x00, 0x00,
```

Pseudo Code menu

- gambar panah

ketika gambar panah telah di gambar melalui aplikasi Paint, file tersebut di convert menjadi code biner dengan aplikasi lcd_asistant sehingga di dapatkan code biner di bawah ini

```
const uint8_t PANAHKCTa1[] PROGMEM={0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,  
0x7E, 0x3C, 0x18, };
```

Pseudo Code menu

- program tampilan pagar dan pintu

setelah mendapatkan code biner icon gembok dan panah, selain itu telah di definisikan menjadi variable, di bawah ini adalah pseudo code untuk menampilkan tampilan pagar dan pintu

```
myLCD.clrScr();  
  
if(nilaiinpuk==1){  
    myLCD.drawBitmap(20, 18, GERBANG7buka, 48, 32);  
}  
  
if(nilaiinpuk==0){  
    myLCD.drawBitmap(20, 18, GERBANG7, 48, 32);  
}  
  
myLCD.drawBitmap(45, 27, PANAHKCTa2, 41, 8);  
myLCD.setFont(SmallFont);  
myLCD.print("PAGAR&PINTU", CENTER, 8);  
myLCD.setFont(TinyFont);  
myLCD.print("home>pagar&pintu", 0, 0);  
myLCD.update();
```

Pseudo Code menu

- c. tampilan lampu

- inisialisasi icon lampu

mendeklarasikan sebuah variable untuk icon lampu yang berguna untuk mempermudah pemanggilan

```
extern uint8_t LAMPUPT[];
```

Pseudo Code menu

- gambar icon lampu

ketika gambar lampu telah di gambar melalui aplikasi Paint, file tersebut di convert menjadi code biner dengan aplikasi lcd_asistant sehingga di dapatkan code biner di bawah ini

```
const uint8_t LAMPUPTT[] PROGMEM={0x80, 0x60, 0x18, 0x04,  
0x04, 0x02, 0x02, 0x01, 0x01, 0x01, 0x01, 0x02, 0x02, 0x04, 0x04,  
0x18, 0x60, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3F, 0xC0,  
0x00,  
0x00, 0x00, 0x00, 0xC0, 0x3F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x03, 0x1C, 0x60, 0x40, 0x40, 0x40, 0x40, 0x40, 0x40,  
0x40, 0x40, 0x60, 0x1C, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x06, 0x36, 0x66, 0x66,  
0x66, 0x66, 0x36, 0x06, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00 };
```

Pseudo Code menu

- gambar panah

ketika gambar panah telah di gambar melalui aplikasi Paint, file tersebut di convert menjadi code biner dengan aplikasi lcd_asisstant sehingga di dapatkan code biner di bawah ini

```
const uint8_t PANAHKCT[] PROGMEM={ 0x18, 0x3C, 0x7E, 0xFF,  
0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x18, };
```

Pseudo Code menu

- program tampilan lampu

setelah mendapatkan code biner icon lampu dan panah, selain itu telah di definisikan menjadi variable, di bawah ini adalah pseudo code untuk menampilkan tampilan lampu

```
myGLCD.clrScr();  
  
myGLCD.drawBitmap(13, 16, LAMPUPTT, 24, 32);  
  
myGLCD.drawBitmap(43, 27, PANAHKCT, 41, 8);  
  
myGLCD.setFont(SmallFont);  
  
myGLCD.print("NYALA", 48, 27);  
  
myGLCD.print("KAMAR TIDUR 2", CENTER, 8);  
  
myGLCD.setFont(TinyFont);  
  
myGLCD.print("home>lamp..>kama..", 0, 0);  
  
myGLCD.update();
```

Pseudo Code menu

- d. tampilan elektronik

- inisialisasi icon elektronik

mendeklarasikan sebuah variable untuk icon-icon elektronik yang berguna untuk mempermudah pemanggilan

```
extern uint8_t TV[];  
extern uint8_t TVI[];  
extern uint8_t RICE[];  
extern uint8_t RICE1[];  
extern uint8_t ACT[];  
extern uint8_t AC1[];
```

Pseudo Code menu

- gambar icon elektronik

ketika gambar elektronik telah di gambar melalui aplikasi Paint, file tersebut di convert menjadi code biner dengan aplikasi lcd_asistant sehingga di dapatkan code biner di bawah ini

```
const uint8_t RICE[] PROGMEM={ 0x00, 0x40, 0xA0, 0xA0, 0xB0,  
0xB8, 0xAC, 0xA6, 0xA2, 0xA2, 0xA2, 0xA2, 0xA2, 0xA2,  
0xA2, 0xA2, 0xA2, 0xA2, 0xA2, 0xA2, 0xA2, 0xA2, 0xA2,  
0xA2, 0xA2, 0xA2, 0xA2, 0xA2, 0xA6, 0xAC, 0xB8, 0xB0, 0xA0,  
0xA0, 0x40, 0x00, 0x00, 0x0C, 0x1E, 0x12, 0xF3, 0x1F, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20,  
0x20, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0xDC,  
0x76, 0x30, 0x10, 0x00, 0x1F, 0xF3, 0x12, 0x1E, 0x0C, 0x00, 0x00,  
0x00, 0x00, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0xE1, 0xED, 0xED, 0xED, 0xED, 0xED, 0xED, 0xED, 0xED,  
0xE1, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x03,  
0x06, 0x0C, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08,  
0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08,  
0x08, 0x08, 0x08, 0x08, 0x0C, 0x06, 0x03, 0x01, 0x00, 0x00, 0x00,  
0x00, };  
  
const uint8_t RICE1[] PROGMEM={ 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```



```
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0x7E, 0x3C,
0x18, };
```

Pseudo Code menu

- program tampilan elektronik

setelah mendapatkan code biner icon elektronik dan panah, selain itu telah di definisikan menjadi variable, di bawah ini adalah pseudo code untuk menampilkan tampilan elektronik

```
myGLCD.clrScr();
if(nilainasihidup==0){
    myGLCD.drawBitmap(0, 8, ricecookermati, 84, 48);
}
if(nilainasihidup==1){
    myGLCD.drawBitmap(0, 8, RICE1, 84, 48);
}
myGLCD.setFont(SmallFont);
myGLCD.print("PENANAK NASI", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>elektronik", 0, 0);
myGLCD.update();
```

Pseudo Code menu

- e. tampilan penyiram tanaman

- inisialisasi icon penyiram tanaman

mendeklarasikan sebuah variable untuk icon penyiram tanaman yang berguna untuk mempermudah pemanggilan

```
extern uint8_t air[];
```

Pseudo Code menu

- gambar icon penyiram tanaman

ketika gambar penyiram tanaman telah di gambar melalui aplikasi Paint, file tersebut di convert menjadi code biner dengan aplikasi lcd_asistant sehingga di dapatkan code biner di bawah ini

```
const uint8_t air[] PROGMEM= {0x00, 0x80, 0xC0, 0xE0, 0x70,
0x38, 0x9C, 0xCE, 0x66, 0x3E, 0x0C, 0x0C, 0x0C, 0x08, 0x18, 0x10,
0x30, 0xE0, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x7B, 0xF9,
0xFC, 0xFE, 0xFF, 0xFF, 0xFE, 0xFE, 0xFE, 0xFE, 0xFE, 0xFE,
0xFC, 0xF8, 0xE0, 0xF8, 0xEF, 0xE6, 0xE0, 0xE0, 0xE0, 0xF0,
0x70, 0x30, 0x10, 0x40, 0x00, 0x80, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x01, 0x03, 0x07, 0x0F, 0x1F, 0x1F, 0x0F, 0x0F,
0x07, 0x07, 0x03, 0x03, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00,
0x01, 0x00, 0x00, 0x00, 0x05, 0x00, 0x0A, 0x00, 0x15, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

Pseudo Code menu

- gambar panah

ketika gambar panah telah di gambar melalui aplikasi Paint, file tersebut di convert menjadi code biner dengan aplikasi lcd_assistant sehingga di dapatkan code biner di bawah ini

```
const uint8_t PANAHKCTa2[] PROGMEM={ 0x18, 0x3C, 0x7E,
0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

Pseudo Code menu

- program tampilan lampu

setelah mendapatkan code biner icon lampu dan panah, selain itu telah di definisikan menjadi variable, di bawah ini adalah pseudo code untuk menampilkan tampilan lampu

```
myGLCD.clrScr();
if(nilaiairnyala==1){
    myGLCD.drawBitmap(20, 18, air, 35, 32);
}
if(nilaiairnyala==0){
    myGLCD.drawBitmap(20, 18, air1, 35, 32);
}
myGLCD.setFont(SmallFont);
myGLCD.print("PENYIRAM", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>penyiram", 0, 0);
```



```
myGLCD.update();
```

Pseudo Code menu

3. Coding Transfer Data

Pada proses pengiriman data hal yang pertama dilakukan adalah inisialisasi library dan pin 9 yang digunakan untuk pengiriman data. Pseudo code inisialisasi library dan pin sebagai berikut

```
#include <SoftwareSerial.h>  
  
SoftwareSerial mySerial(8,9); // RX, TX
```

Pseudo Code menu

Setelah dapat digunakan sebagai pengiriman data, berikut adalah code yang digunakan sebagai pengiriman dan penerima data

```
mySerial.println(); //untuk pengiriman data  
  
n=mySerial.read(); // sebagai penerima data
```

Pseudo Code menu

5.2.4.2 Implementasi Perancangan Perangkat Lunak Pada Maket

Setelah perancangan maket selesai, pada implementasi perancangan perangkat lunak di buat program untuk menerima data dari *remote* dan program menyalakan indicator lampu sesuai dengan data yang diterima. Berikut code program maket.

a. Code penerima data *remote*

```
#include <SoftwareSerial.h>  
  
SoftwareSerial mySerial(8,9);  
  
Voidloop{  
n=mySerial.read();  
}
```

Pseudo Code menu

b. Code menyalakan indicator lampu sesuai data yang diterima

```
n=mySerial.read();  
  
if(n=='0'){  
digitalWrite(a, LOW);}  
  
if (n=='1'){  
digitalWrite(a, HIGH);}
```

Pseudo Code menu

BAB 6 PENGUJIAN DAN ANALISIS

6.1 Pengujian Dan Analisis

Pengujian *remote control smart home* ini dilakukan dalam 2 tahap. Tahap yang pertama yaitu pengujian verifikasi alat, selanjutnya pengujian usabilitas alat yang menggunakan keikutsertaan user. Untuk verifikasi alat, proses untuk mengamati fungsionalitas pada tiap-tiap bagian pada alat apakah sudah sesuai dengan perancangan. Selanjutnya pengujian usabilitas, yaitu melakukan pengujian kepada user.

6.2 Pengujian Verifikasi Alat

Pada verifikasi alat ada beberapa bagian yang akan di uji, bagian tersebut dapat di gambarkan pada grafik

Tabel 6.1 Verifikasi Alat

No	Kebutuhan	Tujuan	Hasil yang diharapkan
1	Uji kinerja arduino	Memastikan program telah masuk ke arduino	Pada software arduino IDE memberi informasi “done uploading” dan pada arduino lampu indicator menyala
2	Uji kinerja tombol ke arduino	Memastikan input dari tombol ke arduino terhubung	Input dari tombol telah sesuai
3	Uji kinerja tampilan LCD	Memastikan tampilan LCD sesuai dengan input tombol	LCD menampilkan sesuai dengan input dari tombol
4	Uji input dari <i>remote</i> ke maket	transfer data dari arduino <i>remote</i> kontrol via kabel tx rx dapat terkirim ke arduino pada maket.	Transfer data dari <i>remote</i> ke maket terkirim
5	Uji Nyala-Mati LED pada maket	kesesuaian Nyala-Mati LED pada maket berdasarkan Input dari <i>Remote</i>	LED pada maket menyala dan mati sesuai dengan input pengiriman data dari <i>remote</i> .

6.2.1 Pengujian Dan Analisis *Remote*

Pengujian analisis *remote* dilakukan untuk mengamati fungsionalitas *remote control* baik berupa entry input dari tombol dan tampilan LCD. Pengujian



remote ini meliputi 3 bagian yaitu pengujian perangkat lunak arduino, pengujian tombol ke arduino dan pengujian tampilan LCD pada *remote*. Pengujian *remote* ini dapat dilihat pada tabel

Pengujian dan Analisis Kinerja Arduino

a. Tujuan

untuk memastikan apakah program dapat diterima / masuk ke arduino

b. Peralatan

1. arduino

2. kabel usb arduino

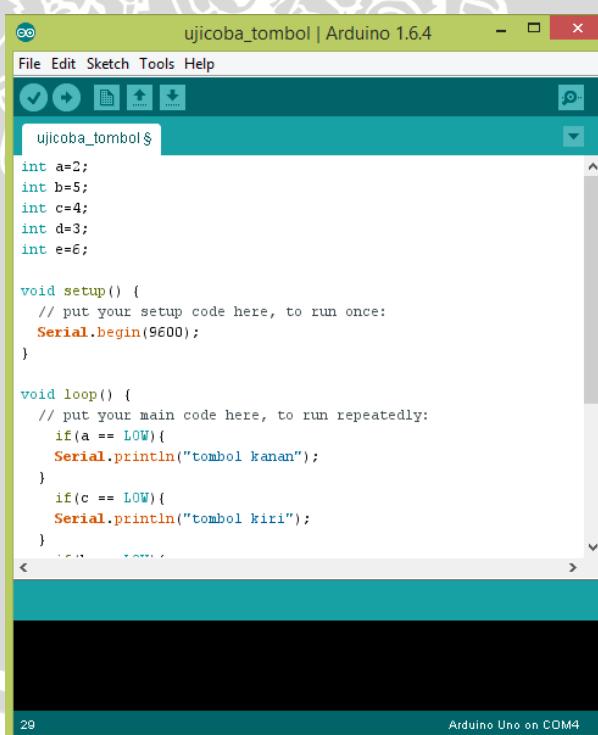
3. arduino IDE

c. Langkah Pengujian

1. Menghubungkan arduino nano dengan laptop

2. Membuka aplikasi arduino IDE

3. Meng-upload program pada arduino dengan memilih menu upload seperti pada Gambar 6.1



The screenshot shows the Arduino IDE interface with the title bar "ujicoba_tombol | Arduino 1.6.4". The code editor contains the following sketch:

```
ujicoba_tombol
int a=2;
int b=5;
int c=4;
int d=3;
int e=6;

void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
}

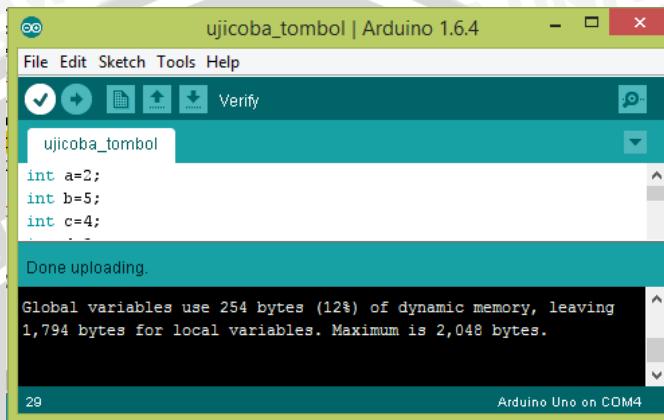
void loop() {
    // put your main code here, to run repeatedly:
    if(a == LOW){
        Serial.println("tombol kanan");
    }
    if(c == LOW){
        Serial.println("tombol kiri");
    }
}
```

The status bar at the bottom indicates "Arduino Uno on COM4".

Gambar 6.1 Program Pengujian Arduino

4. Mengamati pada lampu indikator arduino apakah menyala atau tidak, karena indikator LED akan menyala bila program di-upload.
- d. Hasil

Hasil dari langkah pengujian di atas dapat dilihat pada gambar 6.3 dan LED indikator yang menyala dapat dilihat pada gambar 6.2 yang memberi info bahwa program telah dimasukkan dengan sukses



Gambar 6.2 Hasil Pada Arduino IDE



Gambar 6.3 Hasil Pada Arduino

- e. Analisis

Program telah sukses diupload pada arduino sehingga arduino dapat digunakan untuk mengendalikan perangkat smart home.

Pengujian Kinerja Tombol Ke Arduino

- a. Tujuan

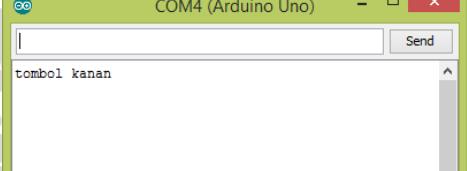
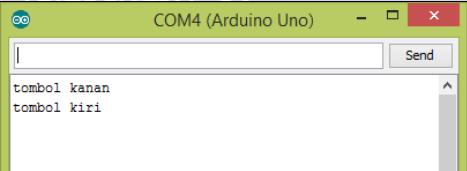
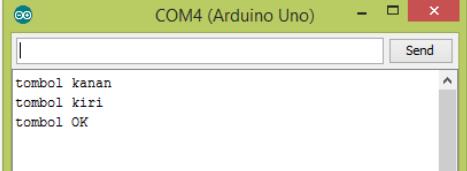
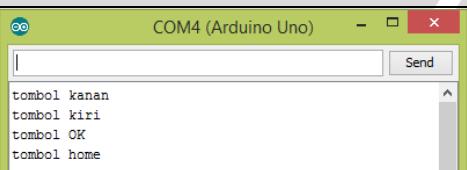
untuk mengetahui fungsionalitas tombol/ push button apakah sudah terhubung ke arduino

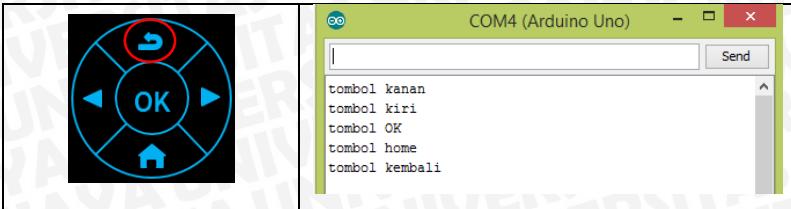
- b. peralatan

1. tombol
 2. arduino
 3. arduino IDE
- c. langkah pengujian
1. menghubungkan kabel pada tombol ke arduino
 2. membuka aplikasi arduino IDE
 3. maupload program uji coba tombol
 4. mencoba tekan tombol panah kiri, panah kanan, ok, home, back
 5. mengamati hasil keluaran pada serial monitor pada aplikasi arduino IDE apakah inputan sudah diterima dengan sesuai
- d. hasil

hasil dari langkah diatas dapat dilihat pada tabel di bawah ini:

Tabel 6.2 Hasil Pengujian Tombol

Tombol yang di tekan	Hasil pengamatan dari serial monitor
	
	
	
	



e. analisis

Input dari semua tombol yang ditekan telah diterima di arduino dengan sesuai dan arduino dapat memproses data yang masuk dari input tombol.

Pengujian Tampilan LCD

a. Tujuan

untuk mengetahui apakah tampilan LCD yang sesuai dengan input dari tombol

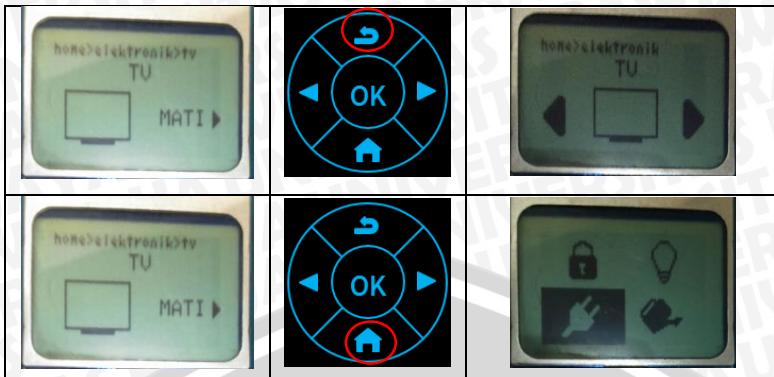
b. Peralatan

1. tombol
2. arduino nano
3. arduino IDE

Tampilan LCD sebelum menekan tombol	Tombol yang di tekan	Tampilan LCD sesudah menekan tombol

c. Langkah Pengujian

1. menghubungkan tombol dan LCD menggunakan kabel ke arduino



2. membuka aplikasi arduino IDE
3. mangupload program menu ke dalam arduino
4. menekan tombol panah kiri, panah kanan, ok, back dan home
5. mengamati apakah perubahan tampilan pada LCD sesuai

d. hasil

Tabel 6.3 Hasil Pengujian LCD

e. analisis

tampilan pada LCD sudah sesuai dengan perancangan inputan dari tombol sudah sesuai.

6.2.2 Pengujian Dan Analisis Fungsionalitas Pengiriman Data Dari *Remote Control* Ke Maket

a. tujuan

untuk mengetahui apakah transfer data dari arduino *remote control* via kabel dapat terkirim ke arduino pada maket.

b. peralatan

1. *remote*
2. arduino uno sebagai mikromkomputer maket
3. arduino IDE sebagai mikrokomputer *remote control*

c. langkah pengujian

1. menghubungkan koneksi data *remote TX* dan *RX* menggunakan kabel ke pin TX RX pada arduino
2. membuka aplikasi arduino IDE

3. mangupload program "ujicobamaket" ke dalam arduino *remote*
4. mangupload program ke dalam arduino uno
5. menekan tombol panah kiri, panah kanan, ok, back dan home pada *remote*
6. membuka serial monitor pada aplikasi arduino IDE
7. mengamati tampilan pada serial monitor pada aplikasi arduino IDE

d. hasil

Tabel 6.4 Hasil Pengujian Pengiriman Data

Tombol	Serial Monitor Pada Arduino Maket
	 COM4 (Arduino Uno)

e. analisa

Hasil pengujian data menunjukkan bahwa pengiriman data dari button *remote* yang diumpulkan ke arduino IDE dapat diterima oleh arduino uno pada maket dengan benar dan sesuai.

6.2.3 Pengujian Dan Analisis Kesesuaian Nyala-Mati Led Pada Maket Berdasarkan Input Dari *Remote*

- a. Tujuan

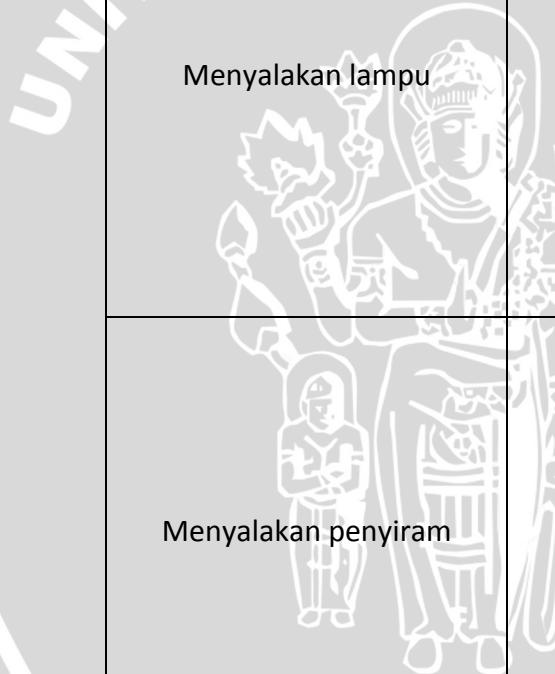
untuk mengetahui apakah kondisi nyala-mati lampu LED sebagai indikator on/off perangkat smart home bekerja dengan sesuai.
- b. peralatan
 1. tombol *remote*
 2. arduino uno
 3. arduino IDE
 4. LED
- c. langkah pengujian
 1. menghubungkan koneksi data *remote* TX-RX menggunakan kabel ke pin TX-RX pada arduino
 2. membuka aplikasi arduino IDE
 3. mengupload program “menu” ke dalam arduino *remote*
 4. masukkan program “tes” ke dalam arduino uno
 5. memberikan perintah buka pintu, mengunci pintu, menyalakan lampu, mematikan lampu, menyalakan perangkat elektronik, mematikan perangkat elektronik, ada lagi? Penyiram?
 6. mengamati kondisi LED pada maket
- d. hasil

Tabel 6.5 Hasil Pengujian Maket

Perintah	Lampu yang menyal
Membuka pintu	



Mengunci pintu



Menyalakan lampu



Menyalakan penyiram





e. analisis

Dari hasil yang didapatkan LED pada maket menyal dan mati sesuai dengan input pengiriman data dari *remote*.

Dari pengujian setiap bagian verifikasi alat didapatkan kesimpulan yang di jelaskan pada grafik di bawah ini

Tabel 6.6 Hasil Verifikasi Alat

No	Kebutuhan	Tujuan	Hasil yang diharapkan	Hasil yang didapatkan	status
1	Uji kinerja arduino	Memastikan program telah masuk ke arduino	Pada software arduino IDE memberi informasi “done uploading” dan pada arduino lampu indicator menyal	Pada software arduino IDE memberi informasi “done uploading” dan pada arduino lampu indicator menyal	sesuai
2	Uji kinerja tombol ke arduino	Memastikan input dari tombol ke arduino terhubung	Input dari tombol telah sesuai	Input dari tombol telah sesuai	Sesuai
3	Uji kinerja tampilan LCD	Memastikan tampilan LCD sesuai dengan	LCD menampilkan sesuai dengan input dari tombol	LCD menampilkan sesuai dengan input dari tombol	Sesuai

		input tombol	input dari tombol		
4	Uji input dari <i>remote</i> ke maket	transfer data dari arduino <i>remote</i> kontrol via kabel tx rx dapat terkirim ke arduino pada maket.	Transfer data dari <i>remote</i> ke maket terkirim	Transfer data dari <i>remote</i> ke maket terkirim	Sesuai
5	Uji Nyala-Mati LED pada maket	kesesuaian Nyala-Mati LED pada maket berdasarkan Input dari <i>Remote</i>	LED pada maket menyala dan mati sesuai dengan input pengiriman data dari <i>remote</i> .	LED pada maket menyala dan mati sesuai dengan input pengiriman data dari <i>remote</i> .	sesuai

6.3 Pengujian dan Analisis Usabilitas

Tujuan perngujian ini adalah untuk menguji USE, "U" untuk Usefullness atau kegunaan yang mewakili seberapa besar kegunaan alat ini. "S" untuk satisfaction atau kepuasan user terhadap alat yang dibuat. Dan "E" untuk easy atau kemudahan mempelajari dan mengoperasikan sebuah alat. Pengujian ini dilakukan dengan 2 cara yaitu dengan pengujian menggunakan teknik observasi dan query berupa kuisinoeruesioner.

Pengujian ini dilakukan dengan metode sampling yang mana alat ini di ujikan ke 30 responden yang terdiri dari 8 responden mahasiswa dengan jurusan T.I (Teknologi Informasi), 7 responden dosen dengan jurusan T.I, 8 responden mahasiswa dengan jurusan non-T.I, 7 responden dosen dengan jurusan non-T.I. telah di jelaskan pada sub-bab 3.1.5 pegujian analisis system.

6.3.1 Pengujian Menggunakan Observasi

Pengujian menggunakan observasi ini dilakukan dengan meminta responden untuk mencoba *remote* dan menjalankan sesuai dengan sekenario yang telah di siapkan. Indikator yang dinilai dari pengujian ini adalah waktu yang dibutuhkan responden untuk melaksanakan sekenario dan mengakses atau tidak mengakses buku manual yang telah disediakan.

- a. Tujuan



Untuk mengetahui rata-rata waktu penggeraan dan tingkat keberhasilan responden ketika melaksanakan skenario

b. Peralatan

1. *Remote control smart home*
2. Maket
3. Stopwatch
4. Lembar observasi untuk isian waktu
5. bolpoint

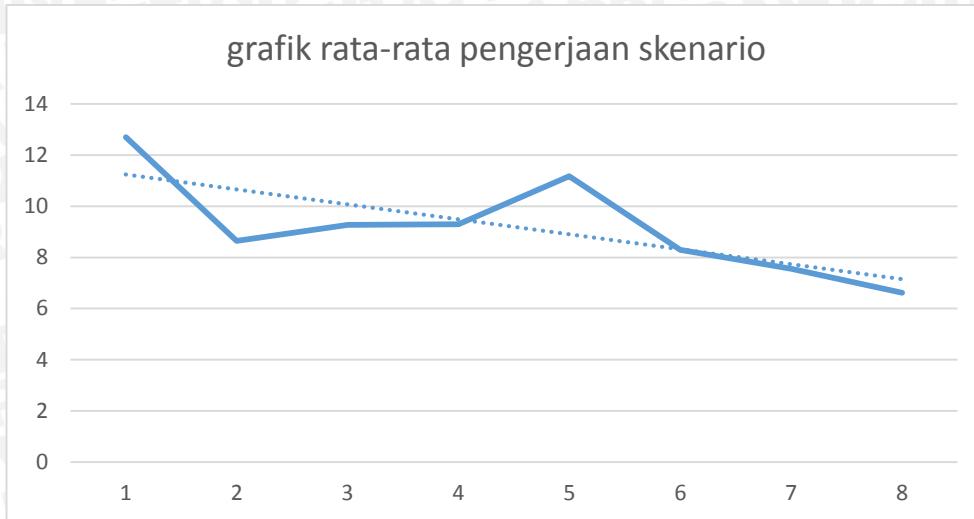
c. Langkah

1. Menghubungkan *remote* dan maket dengan sumber daya
2. Menghubungkan kabel tx-rx *remote* dengan maket
3. Meminta user melakukan satu persatu skenario
4. Mencatat waktu yang dibutuhkan responden ketika selesai menjalankan skenario
5. Menandai apabila responden membaca buku manual sebelum atau saat menjalankan skenario

d. Hasil pengujian

IT/non-IT	kejaya/remaja	No. responden	lama penggeraan (s)								rata-rata (s)	mengakses buku manual				
			lampu		pintu dan pagar		elektronik		penyiram tanaman							
			nyala	mati	buka	kunci	nyala	mati	nyala	mati						
IT	remaja	1	8	7	12	6	7	8	11	7	8,25	tidak				
		2	7	4	4	3	4	7	5	6	5	tidak				
		3	8	15	13	9	13	10	30	7	13,125	tidak				
		4	8	19	9	10	12	13	12	9	11,5	tidak				
		5	13	7	4	4	6	7	2	3	5,75	tidak				
		6	10	9	8	5	5	4	3	4	6	tidak				
		7	9	11	6	11	10	6	8	9	8,75	tidak				
		8	19	5	14	9	10	7	7	7	9,75	tidak				
	rata-rata IT remaja										8,5156	-				
	kejaya	9	30	13	9	18	7	6	5	5	11,625	tidak				
		10	4	7	13	8	9	9	10	6	8,25	tidak				
		11	7	5	8	9	36	10	4	7	10,75	tidak				
		12	6	6	3	7	9	8	4	4	5,875	tidak				
		13	10	5	6	7	13	7	6	6	7,5	tidak				
		14	5	5	7	6	6	4	4	7	5,5	ya				
		15	12	9	8	10	9	8	12	8	9,5	ya				
	rata-rata IT kerja										8,4286	-				
non-IT	remaja	16	5	6	4	4	7	4	3	3	4,5	tidak				
		17	7	5	6	6	8	5	6	6	6,125	tidak				
		18	17	9	11	19	14	15	9	7	12,625	tidak				
		19	11	6	10	6	5	6	4	5	6,625	ya				
		20	15	6	7	9	7	6	4	6	7,5	tidak				
		21	8	6	7	9	7	6	4	6	6,625	ya				
		22	8	7	10	7	9	9	7	6	7,875	ya				
		23	19	9	9	6	9	11	8	4	9,375	ya				
	rata-rata non-IT remaja										8,1071	-				
	kejaya	24	6	6	7	7	11	5	5	4	6,375	tidak				
		25	27	10	17	6	11	6	4	12	11,625	tidak				
		26	10	10	11	11	15	9	9	8	10,375	tidak				
		27	14	11,5	14	14	23	11	6	7	12,563	tidak				
		28	36	11	9	21	14	8	6,5	8,5	14,25	tidak				
		29	12	10	12	12	16	9	13	7	11,375	ya				
		30	30	20	20	20	23	25	15	14	20,875	tidak				
rata-rata non-IT kerja											12,491	-				
rata-rata (s)			12,7	8,65	9,267	9,3	11,17	8,3	7,55	6,6167	9,1938					

Gambar 6.4 Hasil Keseluruhan Pengujian Observasi



Gambar 6.5 Grafik Rata-rata Penggerjaan Skenario

grafik rata-rata waktu penggerjaan per-skenario dari 30 responden dengan sumbu x adalah skenario awal sampai akhir, sumbu y adalah waktu

e. Analisis

Pada tabel di atas didapatkan :

1. Rata-rata waktu yang dibutuhkan responden dalam menjalankan skenario adalah 9,2 detik.
2. Dari perbandingan rata-rata antara IT dan non-IT, maka IT lebih cepat 1,3 detik
3. Dari perbandingan rata-rata antara remaja dan pekerja, maka remaja lebih cepat 2,4 detik
4. Semua penggerjaan setiap skenario kurang dari 13 detik
5. Dari 30 responden, yang membutuhkan/membaca buku manual terlebih dahulu sebelum atau saat menjalankan skenario ada 7 responden, sedangkan yang langsung mencoba menjalankan skenario ada 23 responden. Sehingga pembuatan buku manual membantu pengguna untuk mengoperasikan alat.
6. Tingkat keberhasilan responden melaksanakan semua skenario adalah sebesar 100%
7. Pada grafik menggambarkan rata-rata waktu penggerjaan skenario 1-4 berkurang dikarenakan user belajar cepat dalam mencoba *remote*, lihat garis putus-putus.
8. ketika masuk pada skenario 5 perubahan rata-rata waktu naik dikarenakan saat percobaan user melihat menu elektronik terletak di bawah baris menu lampu, sehingga user mengharapkan aksi tombol ke bawah dari menu kunci ke menu elektronik. Namun pada tombol yang disediakan di *remote*, hanya terdapat tombol kiri-

kanan. Sehingga beberapa user kebingungan untuk menjalankan scenario 5. Namun dengan penggunaan berulang-ulang, pilihan tombol kiri-kanan tersebut akan lebih mudah dipahami.

6.3.2 Pengujian Menggunakan Questioner

Pengujian menggunakan questioner ini dilakukan dengan meminta responden untuk mengisi sebuah questioner. Questioner yang digunakan adalah USE quistionaire yang mencakup pertanyaan tentang usefulness, satisfaction dan easy to use dan easy to learn dari alat yang dicobakan. Total terdapat 30 pertanyaan dengan 8 soal untuk usefulness, 11 soal untuk satisfaction, dan easy to use 4 soal, easy to learn 7 soal. Questioner diisi setelah responden mencoba dan mempraktekkan alat ini. Responden dapat mengisi questioner berdasarkan pengalaman user dalam proses pengujian. Pertanyaan yang diajukan dalam questioner ini adalah:

1. Usefulness/kegunaan

Tabel 6.7 Tabel Pertanyaan Usefulness

1	<i>Remote</i> ini membantu saya menjadi lebih efektif dan efisien dalam mengontrol rumah
2	<i>Remote</i> ini membantu saya menjadi lebih produktif
3	<i>Remote</i> ini berguna
4	<i>Remote</i> ini memberi saya kontrol lebih besar untuk kegiatan kontrol rumah
5	<i>Remote</i> ini membuat kontrol rumah lebih mudah untuk dilakukan
6	<i>Remote</i> ini menghemat waktu saya
7	<i>Remote</i> ini memenuhi kebutuhan saya saat mengontrol rumah
8	<i>Remote</i> ini melakukan kontrol rumah sesuai dengan yang saya harapkan

2. satisfaction/kepuasan

Tabel 6.8 Tabel Pertanyaan Satisfaction

9	<i>Remote</i> ini mudah digunakan
10	<i>Remote</i> ini sederhana
11	<i>Remote</i> ini user friendly
12	<i>Remote</i> ini membutuhkan langkah-langkah yang mungkin paling sedikit dalam mengontrol rumah

13	<i>Remote</i> ini fleksibel
14	<i>Remote</i> ini dapat digunakan tanpa kesulitan
15	Saya dapat menggunakannya tanpa instruksi tertulis
16	Saya tidak melihat adanya inkonsistensi saat saya menggunakannya
17	Pengguna yang sesekali dan teratur memakai <i>remote</i> ini akan menyukainya
18	Jika melakukan kesalahan dalam penggunaannya, saya akan dengan cepat dan mudah memperbaikinya
19	saya dapat menggunakannya dengan sukses setiap kali

3. easy to use/kemudahan kegunaan

Tabel 6.9 Tabel Pertanyaan Easy To Use

20	Saya belajar dengan cepat untuk menggunakannya
21	Saya mudah mengingat bagaimana menggunakannya
22	Sangat mudah untuk belajar menggunakannya
23	Saya cepat terampil dengan alat ini

4. easy to learn/ lemudahan dipelajari

Tabel 6.10 Tabel Pertanyaan Easy To Learn

24	Saya puas dengan alat ini
25	Saya akan merekomendasikan <i>remote</i> ini kepada teman
26	<i>Remote</i> ini menyenangkan untuk digunakan
27	<i>Remote</i> ini bekerja dengan cara seperti yang saya inginkan
28	<i>Remote</i> ini indah/hebat
29	Saya merasa harus memiliki <i>remote</i> ini
30	alat ini menyenangkan untuk digunakan

Penilaian masing-masing pertanyaan yang diujikan menggunakan likert scale. Skala ini sering dijumpai pada questioner untuk memudahkan responden memberikan jawaban secara terstruktur dan terbatas. Likert scale adalah suatu skala psikometrik yang umum digunakan dalam questioner , dan merupakan skala yang paling banyak digunakan dalam riset berupa survei. Nama skala ini di ambil dari nama rencana likert, yang menerbitkan suatu laporan yang menjelaskan penggunaannya(). Skala yang digunakan dalam pengujian ini adalah skala 1 sampai 5 yang memiliki arti sebagai berikut:

- Nilai 1 yang menyatakan pilihan sangat tidak setuju
- Nilai 2 yang menyatakan pilihan tidak setuju

3. Nilai 3 yang menyatakan pilihan ragu-ragu
4. Nilai 4 yang menyatakan pilihan setuju
5. Nilai 5 yang menyatakan pilihan sangat setuju

1. Tujuan

Untuk mengukur usabilitas dari keempat kategori penilaian yaitu tingkat kegunaan (usefulness), kepuasan (satisfaction), kemudahan penggunaan (easy to use), dan kemudahan untuk dimengerti (easy to learn) berdasarkan pendapat dari responden.

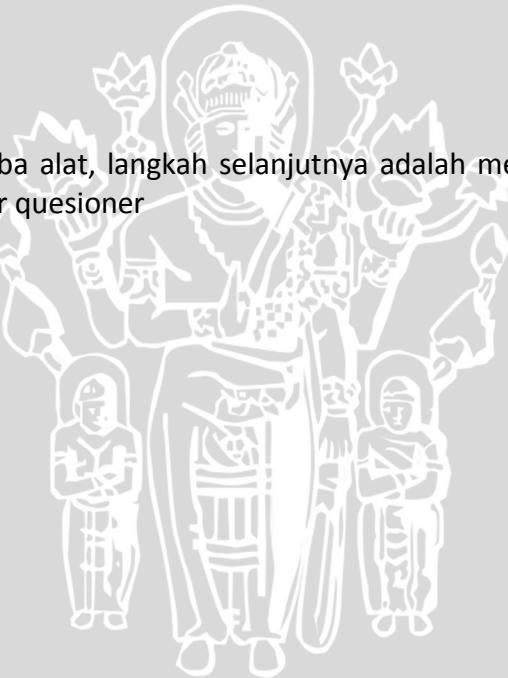
2. Peralatan

1. *Remote control smart home*
2. Maket
3. lampiran questioner
4. bolpoint

3. Langkah pengujian

Setelah mencoba alat, langkah selanjutnya adalah meminta responden mengisi lembar questioner

4. Hasil pengujian



Kegunaan	Kemudahan Penggunaan					Mudah dimengerti					Kejuasan					modus				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
Penilaian 1 (sangat tidak setuju) 5 (sangat setuju)	1	5	3	3	4	4	4	5	4	4	5	3	3	4	3	3	2	5	3	4
Nomor	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	3
kelebihan/rema	2	3	3	5	5	3	5	5	3	5	5	5	5	5	5	5	3	3	3	3
kelebihan/rema	3	4	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
kelebihan/rema	4	4	4	4	4	4	4	5	5	4	4	4	4	4	4	4	4	4	4	4
kelebihan/rema	5	4	4	4	5	4	5	4	5	5	5	4	4	4	4	4	5	4	4	5
kelebihan/rema	6	4	3	4	5	5	4	4	5	4	4	4	4	5	5	5	4	5	4	4
kelebihan/rema	7	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
kelebihan/rema	8	5	4	5	5	5	4	4	5	4	4	4	5	5	3	4	5	5	5	4
kelebihan/rema	9	3	2	4	3	5	3	4	3	3	3	3	2	3	2	4	4	4	4	4
kelebihan/rema	10	4	3	4	3	4	4	3	3	4	5	4	4	4	4	4	4	4	4	4
kelebihan/rema	11	4	3	4	3	4	4	3	3	4	2	3	4	4	4	4	4	4	3	4
kelebihan/rema	12	4	4	3	4	4	3	3	3	4	2	3	4	3	3	4	4	4	3	4
kelebihan/rema	13	5	4	5	5	5	4	4	4	5	5	4	5	5	4	5	5	4	4	4
kelebihan/rema	14	4	4	4	4	4	4	3	4	4	4	2	3	3	5	4	4	4	4	4
kelebihan/rema	15	5	4	5	5	5	5	5	5	5	4	4	5	4	5	5	5	5	5	5
kelebihan/rema	16	4	4	3	4	4	4	3	5	5	5	3	4	4	4	4	4	4	4	4
kelebihan/rema	17	4	3	4	4	5	4	3	4	4	5	3	4	5	4	4	4	4	4	4
kelebihan/rema	18	4	4	5	4	5	4	4	4	4	4	4	4	5	5	4	4	4	4	5
kelebihan/rema	19	4	3	4	4	4	4	3	4	4	4	3	4	4	4	3	2	4	4	4
kelebihan/rema	20	4	3	4	3	4	5	4	3	4	4	4	5	3	4	5	5	4	4	3
kelebihan/rema	21	5	3	4	5	5	4	4	4	4	4	4	3	4	4	5	4	5	5	5
kelebihan/rema	22	5	4	5	5	5	4	5	4	5	5	4	5	5	5	5	5	5	5	5
kelebihan/rema	23	3	5	4	4	4	3	2	2	3	3	3	4	3	3	4	3	3	3	2
kelebihan/rema	24	4	4	4	3	3	5	4	4	4	5	5	4	4	4	4	5	5	4	3
kelebihan/rema	25	5	4	5	5	5	5	5	5	5	5	4	4	5	5	5	4	5	5	4
kelebihan/rema	26	3	5	5	5	5	3	4	5	4	3	3	4	3	3	4	4	4	3	3
kelebihan/rema	27	4	4	4	2	3	4	4	3	5	4	3	4	3	3	2	3	3	3	3
kelebihan/rema	28	4	3	3	3	5	3	3	4	4	4	3	4	4	4	4	4	4	4	4
kelebihan/rema	29	4	3	4	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	2
kelebihan/rema	30	3	4	5	4	3	4	2	3	4	2	3	4	2	3	4	5	3	4	3
modus	4	4	4	4	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3

Gambar 6.6 Hasil Keseluruhan Pengujian Quesioner

Analisis

Secara keseluruhan dari 30 pertanyaan pilihan indikator penilaian dan total 900 jawaban yang diberikan, dengan mencari nilai modus dari keseluruhan di dapat nilai 4 yang berarti “puas”. Modus ini merupakan parameter statistik yang sederhana dan umum digunakan untuk menganalisa data-data kategorikal dari likert scale seperti yang digunakan pada USE questioner ini. Analisa hasil pengujian secara rinci dapat di lihat sebagai berikut:

6.3.2.2 Analisis Tingkat Kegunaan

Analisa tingkat “kegunaan” berdasarkan pada gambar 6.5. Dari hasil questioner 30 responden yang melakukan pengujian dan dihitung berdasarkan nilai modus, angka 4 mendapatkan jumlah yang paling banyak yaitu 115 dari total 240 jawaban. Nilai 4 atau setuju yang memiliki arti bahwa untuk penilaian usabilitas alat pada kategori tingkat kegunaan, user setuju terhadap tingkat kegunaan atau alat ini berguna untuk dijadikan pengontrol smart home (tabel *)

Tabel 6.11 Penilaian Jumlah Pada Tingkat Kegunaan

No	Indikator Penilaian	Jumlah Angka
1	Sangat Tidak Setuju	0
2	Tidak Setuju	4
3	Ragu-Ragu	57
4	Setuju	115
5	Sangat Setuju	64

Berdasarkan pada gambar 6.5 kategori kegunaan dapat dianalisa sebagai berikut:

1. Nilai modus jawaban kuisioner bagian kegunaan antara IT dan non-IT adalah 4, maka user IT dan non-IT setuju jika alat ini berguna untuk mengontrol rumah
2. Nilai modus jawaban kuisioner bagian kegunaan antara remaja dan pekerja adalah 4, maka user remaja dan pekerja setuju jika alat ini berguna untuk mengontrol rumah

6.3.2.3 Analisis Tingkat kemudahan penggunaan

Analisa tingkat “kemudahan penggunaan” berdasarkan pada gambar 6.5. Dari hasil questioner 30 responden yang melakukan pengujian dan dihitung berdasarkan nilai modus, angka 4 mendapatkan jumlah yang paling banyak yaitu 155 dari 330 jawaban. Nilai 4 atau setuju yang memiliki arti bahwa untuk penilaian usabilitas alat pada kategori tingkat kemudahan penggunaan, user setuju terhadap tingkat kegunaan atau alat ini berguna untuk dijadikan pengontrol smart home (tabel *)

Tabel 6.12 Penilaian Jumlah Pada Tingkat Kemudahan Penggunaan

No	Indikator Penilaian	Jumlah Angka
1	Sangat tidak setuju	0
2	Tidak setuju	14
3	Ragu-ragu	77
4	Setuju	155
5	Sangat setuju	84

Berdasarkan pada gambar 6.5 kategori kegunaan dapat dianalisa sebagai berikut:

1. Nilai modus jawaban kuisioner bagian kemudahan penggunaan antara IT dan non-IT adalah 4, maka user IT dan non-IT setuju jika alat ini mudah untuk di gunakan sebagai pengontrol perangkat rumah
2. Nilai modus jawaban kuisioner bagian kemudahan penggunaan antara remaja dan pekerja adalah 4, maka user remaja dan pekerja setuju jika alat ini mudah untuk di gunakan sebagai pengontrol perangkat rumah

Dari analisa berdasarkan keilmuan dan umur, didapat bahwa responden setuju bahwa *remote control smart home* ini mempunyai tingkat kemudahan penggunaan tinggi dalam mengontrol rumah.

6.3.2.4 Analisis Tingkat kemudahan untuk dimengerti

Analisa tingkat kemudahan untuk dimengerti berdasarkan pada gambar 6.5. Dari hasil questioner 30 responden yang melakukan pengujian dan dihitung berdasarkan nilai modus, angka 5 mendapatkan jumlah yang paling banyak yaitu 54 dari 120 jawaban. Nilai 5 atau sangat setuju yang memiliki arti bahwa untuk penilaian usabilitas alat pada kategori ini, user setuju terhadap tingkat kemudahan untuk dimengerti ,dan sangat mudah untuk digunakan bagi pengguna.

Tabel 6.13 Penilaian Jumlah Pada Tingkat Kemudahan Untuk Dimengerti

No	Indikator Penilaian	Jumlah Angka
1	Sangat tidak setuju	0
2	Tidak setuju	2
3	Ragu-ragu	14
4	Setuju	50



5	Sangat setuju	54
---	---------------	----

Berdasarkan pada gambar 6.5 kategori kegunaan dapat dianalisa sebagai berikut:

1. Dari perbandingan nilai modus jawaban kuisioner bagian kemudahan untuk dipelajari antara user IT dan non-IT adalah 4 dan 5, maka alat ini menurut user IT setuju jika alat ini mudah untuk digunakan sedangkan non-IT sangat setuju jika alat ini mudah untuk digunakan
2. Dari perbandingan nilai modus jawaban kuisioner bagian kemudahan untuk dipelajari antara user remaja dan pekerja adalah 5 dan 4, maka alat ini menurut user remaja sangat setuju jika alat ini mudah untuk digunakan sedangkan pekerja setuju jika alat ini mudah untuk digunakan

6.3.2.5 Analisis Tingkat Kepuasan

Analisa tingkat kepuasan berdasarkan pada gambar 6.5. Dari hasil questioner 30 responden yang melakukan pengujian dan dihitung berdasarkan nilai modus, angka 5 mendapatkan jumlah yang paling banyak yaitu 101. Nilai 4 atau setuju yang memiliki arti bahwa untuk penilaian usabilitas alat pada kategori ini, user setuju terhadap tingkat kepuasan, atau setelah melakukan percobaan maka user puas dengan kinerja *remote control smart home* ini. Over all nilai modus dari semua kategori

Tabel 6.14 Penilaian Jumlah Pada Tingkat Kepuasan

No	Indikator Penilaian	Jumlah Angka
1	Sangat tidak setuju	0
2	Tidak setuju	11
3	Ragu-ragu	62
4	Setuju	101
5	Sangat setuju	36

Berdasarkan pada gambar 6.5 kategori kegunaan dapat dianalisa sebagai berikut:

1. Dari perbandingan nilai modus jawaban kuisioner bagian kepuasan antara user IT dan non-IT adalah 4, maka alat ini menurut user IT dan non-IT setuju jika user puas dengan alat ini

2. Dari perbandingan nilai modus jawaban kuisioner bagian kepuasan antara user remaja dan pekerja adalah 4, maka alat ini menurut user remaja dan pekerja setuju jika jika user puas dengan alat ini



UNIVERSITAS BRAWIJAYA



BAB 7 PENUTUP

Bagian ini memuat kesimpulan dan saran terhadap skripsi. Kesimpulan dan saran disajikan secara terpisah, dengan penjelasan sebagai berikut:

7.1 Kesimpulan

1. Pada penelitian ini telah dibuat *remote control smart home* dengan menggunakan prinsip desain interaksi.
2. Untuk memaksimalkan usabilitas maka desain *remote control* ini menggunakan alur perancangan proses desain interaksi yaitu pembuatan definisi sistem (what is wanted), analisis perancangan interaksi sistem melalui skenario dan task analysis, dan penerapan design rule (generalizability, learnability, flexibility dan robustness) pada saat mendesain untuk memaksimalkan usabilitas.
3. Dari hasil pengujian verifikasi alat didapatkan bahwa seluruh fungsionalitas *remote* yang terdiri dari tombol, arduino nano sebagai mikrokontroller dan display, serta pengendali rumah menggunakan arduino UNO sebagai mikrokontroler berjalan sesuai dengan perancangan
4. Dari hasil pengujian usabilitas yang melibatkan 30 responden dengan pengkategorian usia (remaja dan pekerja) dan dasar keilmuan (IT dan non-IT) menunjukkan bahwa tingkat keberhasilan dalam menjalankan seluruh skenario sebesar 100% dengan rata-rata user membutuhkan waktu 9,35 detik dalam menjalankan suatu skenario pada pengujian observasi user dan didapatkan nilai 4 (skala likert 1-5) atau puas dalam kategori learnability, flexibility dan robustness. Dari hasil pengujian tersebut juga menunjukkan bahwa *remote control smart home* mudah dipelajari terlihat dari trenline lama penggerjaan setiap skenario yang semakin lama semakin singkat.

7.2 Saran

Untuk pengembangan lebih lanjut pada *remote control smart home* terdapat saran sebagai berikut

1. Pengembangan *remote* ini menggunakan wireless
2. Alat ini masih berupa prototype 1, sehingga masih di lakukan pengembangan lebih lanjut untuk meningkatkan usabilitas alat.
3. Navigasi alat ditingkatkan dari segi tampilan, material dan ergonomis
4. Tampilan layar ditingkatkan supaya lebih menarik menggunakan LCD dengan kualitas lebih bagus
5. Ditambahkan fitur timer misal untuk penyiram tanaman
6. Ditambahkan fitur penambahan alat

DAFTAR PUSTAKA

- Arduino Nano, 2015. Arduinoboardnano. Tersedia di: <<https://www.arduino.cc/en/Main/ArduinoBoardNano>> [diakses 10 desember 2015]
- Arduino Uno, 2015. Arduinoboarduno. Tersedia di: <<https://www.arduino.cc/en/Main/ArduinoBoardUno>> [diakses 10 desember 2015]
- Boone Harry, 2012. Analyzing Likert Data, *Journal Of extention* vol. 50 number 2
- Christoffer Björkskog, 2010 . “ Human computer interaction in ubiquitous smart homes”
- Dix, Alan, 2004. Human–Computer Interaction. 3rd ed. s.l.:Prentice Hall. London.
- Hill, Robin, 1998. Interpersonal Computing and Technology: An Electronic Journal for The 21st Century vol.6
- Lcd_asisstant, 2011. Bitmap converter tersedia di: <http://en.radzio.dxp.pl/bitmap_converter/> [diakses 20 september 2015]
- Santosh.M.Nejakar. 2014. Wireless Infrared Remote Controller for Multiple Home Appliances, *International Journal of Electrical and Electronics Research* vol. 2
- Schiefer Michael, 2015. Smart Home Definition and Security Threats, Ninth International Conference on IT Security Incident Management & IT Forensics.

Lampiran

A.1 Source Code

A.1.1 Uji coba tombol

```
int a=2;
int b=5;
int c=4;
int d=3;
int e=6;

void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
}

void loop() {
    // put your main code here, to run repeatedly:
    if(a == LOW){
        Serial.println("tombol kanan");
    }
    if(c == LOW){
        Serial.println("tombol kiri");
    }
    if(b == LOW){
        Serial.println("tombol OK");
    }
    if(e == LOW){
        Serial.println("tombol home");
    }
    if(d == LOW){
        Serial.println("tombol kembali");
    }
}
```

A.1.2 Uji coba maket

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(8,9); // RX, TX

int a=2;
int b=5;
int c=4;
int d=3;
int e=6;

void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
}
```



```
void loop() {
// put your main code here, to run repeatedly:
if(a == LOW){
mySerial.println("1");
}
if(c == LOW){
mySerial.println("2");
}
if(b == LOW){
mySerial.println("3");
}
if(e == LOW){
mySerial.println("4");
}
if(d == LOW){
mySerial.println("5");
}
}
```

A.1.3 Uji coba tombol2

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(8,9);
char n;

void setup()
{
// Open serial communications and wait for port to open:

Serial.begin(9600);
while (!Serial) {
; // wait for serial port to connect. Needed for Leonardo only
}

//Serial.println("Goodnight moon!");

// set the data rate for the SoftwareSerial port
mySerial.begin(9600);
//mySerial.println("Hello, world?");
}

void loop() // run over and over
{
if (mySerial.available()){
n=mySerial.read();
delay(100);
Serial.write(n);
}

//gerbang
if (n=='1'){

}
```



```
Serial.println("tombol kanan");
}
if(n== '2'){
Serial.println("tombol kiri");
}
if(n== '3'){
Serial.println("tombol OK");
}
if(n== '4'){
Serial.println("tombol home");
}
if(n== '5'){
Serial.println("tombol kembali");
}
```

A.1.4 Menu

```
#include <LCD5110_Graph.h>
#include <SoftwareSerial.h>

SoftwareSerial mySerial(8,9); // RX, TX
LCD5110 myGLCD(13,11,12,10,7);

const int a=2;
const int b=5;
const int c=4;
const int d=3;
const int e=6;

int nilaair;
int nilaipintu;
int nilaipinbuka;
int tombol;
int tombolelex;

int lamr;
int lamv;

int pinr;
int pinv;

int elexr;
int elevx;

int tvr;
int ttv;

int nasir;
int nasiv;

int acr;
int acv;
```



```
int airr;
int airv;
int nilaiairnyala;

int pl;
int pil;

int niltamu;
int nilkamar1;
int nilkamar2;
int nilmandi;
int nilteras;

int niltamunyala;
int nilkamar1nyala;
int nilkamar2nyala;
int nilmandinyala;
int nilterasnyala;

int nilaitv;
int nilaiac;
int nilainasi;
int nilaitvhidup;
int nilaiachidup;
int nilainasihidup;

extern uint8_t menugerbang[];
extern uint8_t menulampu[];
extern uint8_t menuelektro[];
extern uint8_t menuair[];
extern uint8_t menugerbangbuka[];
extern uint8_t menuairmati[];

extern uint8_t menugerbanginvert [];
extern uint8_t menulampuinvert [];
extern uint8_t menuelektroinvert [];
extern uint8_t menuairinvert [];
extern uint8_t menugerbangbukainvert[];
extern uint8_t menuairmatiinvert[];

extern uint8_t GERBANG7buka[];
extern uint8_t GERBANG7[];

extern uint8_t LAMPUPT[];
extern uint8_t LAMPUTT[];
extern uint8_t LAMPUHT[];
extern uint8_t LAMPUTHT[];
extern uint8_t PANAHKCT[];
extern uint8_t PANAHKCT11[];
extern uint8_t PANAHKCT21[];
extern uint8_t PANAHKCT12[];
extern uint8_t PANAHKCT22[];
```

```
extern uint8_t TV[];  
extern uint8_t TVI[];  
extern uint8_t RICE[];  
extern uint8_t RICE1[];  
extern uint8_t ACT[];  
extern uint8_t AC1[];  
  
extern uint8_t tvmmati[];  
extern uint8_t tvmati[];  
extern uint8_t ricecoermati[];  
extern uint8_t ricecooker1mati[];  
extern uint8_t ac1mati[];  
extern uint8_t ACTmati[];  
  
extern uint8_t PANAHKCTa1[] ;  
extern uint8_t PANAHKCTa2[] ;  
extern uint8_t air[];  
extern uint8_t air1[];  
extern uint8_t SmallFont[];  
extern uint8_t TinyFont[];  
  
int buttonState = 0;  
  
void setup()  
{  
  
    while (!Serial) {  
        ; // wait for serial port to connect. Needed for Leonardo only  
    }  
    mySerial.begin(9600);  
    myGLCD.InitLCD();  
  
}  
  
void loop()  
{  
    if (digitalRead(a) == LOW && airr==0 && lamr==0 && pinr==0 && elexr==0){  
        if(tombol==4){  
            tombol=1;  
        }  
        else{tombol=tombol+1;}  
        delay(300);  
    }  
    if (digitalRead(c) == LOW && airr==0 && lamr==0 && pinr==0 && elexr==0){  
        if(tombol==1 || tombol==0){  
            tombol=4;  
        }  
        else{tombol=tombol-1;}  
        delay(300);  
    }  
    if(digitalRead(e) == LOW){  
        pinr=0; pinv=0; lamr=0; lamv=0; pil=0; elexr=0; elexv=0; tombolelex=0;  
        tsvr=0; tvv=0; acr=0; acv=0; nasir=0; nasiv=0; airr=0; airv=0;  
    }  
}
```

```
if(digitalRead(b) == LOW && airr==0 && lamr==0 && pinr==0 && elexr==0){  
    if(tombol==1){  
        pinr=1;  
        pinv=1;  
        delay(300);  
    }  
  
    if(tombol==2){  
        lamr=1;  
        lamv=1;  
        pl=1;  
        delay(300);  
    }  
  
    if(tombol==3){  
        elexr=1;  
        elevx=1;  
        delay(300);  
    }  
  
    if(tombol==4){  
        airr=1;  
        airv=1;  
        delay(300);  
    }  
  
}  
  
if(tombol==0 && airr==0 && lamr==0 && pinr==0 && elexr==0){  
    myGLCD.clrScr();  
    if(nilainipbuka==0){  
        myGLCD.drawBitmap(0, 0, menugerbanginvert, 42, 24);  
    }  
    if(nilainipbuka==1){  
        myGLCD.drawBitmap(0, 0, menugerbangbukainvert, 42, 24);  
    }  
    myGLCD.drawBitmap(42, 0, menulampu, 42, 24);  
    myGLCD.drawBitmap(0, 24, menuelektro, 42, 24);  
    myGLCD.drawBitmap(42, 24, menuairmati, 42, 24);  
    myGLCD.update();}  
  
if(tombol==1 && airr==0 && lamr==0 && pinr==0 && elexr==0){  
    myGLCD.clrScr();  
    if(nilainipbuka==0){  
        myGLCD.drawBitmap(0, 0, menugerbanginvert, 42, 24);  
    }  
    if(nilainipbuka==1){  
        myGLCD.drawBitmap(0, 0, menugerbangbukainvert, 42, 24);  
    }  
  
    myGLCD.drawBitmap(42, 0, menulampu, 42, 24);  
    myGLCD.drawBitmap(0, 24, menuelektro, 42, 24);
```

```
if(nilaiairnyala==1){
    myGLCD.drawBitmap(42, 24, menuair, 42, 24);
}
if(nilaiairnyala==0){
    myGLCD.drawBitmap(42, 24, menuairmati, 42, 24);
}
myGLCD.update();

}

if(tombol==2 && airr==0 && lamr==0 && pinr==0 && elexr==0){
    myGLCD.clrScr();
    if(nilaipinbuka==0){
        myGLCD.drawBitmap(0, 0, menugerbang, 42, 24);
    }
    if(nilaipinbuka==1){
        myGLCD.drawBitmap(0, 0, menugerbangbuka, 42, 24);
    }

    myGLCD.drawBitmap(42, 0, menulampuinvert, 42, 24);
    myGLCD.drawBitmap(0, 24, menuelektro, 42, 24);
    if(nilaiairnyala==1{
        myGLCD.drawBitmap(42, 24, menuair, 42, 24);
    }
    if(nilaiairnyala==0){
        myGLCD.drawBitmap(42, 24, menuairmati, 42, 24);
    }
    myGLCD.update();

}
if(tombol==3 && airr==0 && lamr==0 && pinr==0 && elexr==0){
    myGLCD.clrScr();
    if(nilaipinbuka==0){
        myGLCD.drawBitmap(0, 0, menugerbang, 42, 24);
    }
    if(nilaipinbuka==1){
        myGLCD.drawBitmap(0, 0, menugerbangbuka, 42, 24);
    }

    myGLCD.drawBitmap(42, 0, menulampu, 42, 24);
    myGLCD.drawBitmap(0, 24, menuelektroinvert, 42, 24);
    if(nilaiairnyala==1{
        myGLCD.drawBitmap(42, 24, menuair, 42, 24);
    }
    if(nilaiairnyala==0){
        myGLCD.drawBitmap(42, 24, menuairmati, 42, 24);
    }
    myGLCD.update();

}

if(tombol==4 && airr==0 && lamr==0 && pinr==0 && elexr==0){
    myGLCD.clrScr();
    if(nilaipinbuka==0){
        myGLCD.drawBitmap(0, 0, menugerbang, 42, 24);
    }
```

```
        }

        if(nilainpbuka==1){
            myGLCD.drawBitmap(0, 0, menugerbangbuka, 42, 24);
        }

        myGLCD.drawBitmap(42, 0, menulampu, 42, 24);
        myGLCD.drawBitmap(0, 24, menuelektro, 42, 24);
        if(nilairnyala==1){
            myGLCD.drawBitmap(42, 24, menuairinvert, 42, 24);
        }

        if(nilairnyala==0){
            myGLCD.drawBitmap(42, 24, menuairmatiinvert, 42, 24);
        }

        myGLCD.update();

    }

    //coding pintu
    if(pinr==1 && pinv==1){

        myGLCD.clrScr();
        if(nilainpbuka==1){
            myGLCD.drawBitmap(20, 18, GERBANG7buka, 48, 32);
        }
        if(nilainpbuka==0){
            myGLCD.drawBitmap(20, 18, GERBANG7, 48, 32);
        }

        myGLCD.setFont(SmallFont);
        myGLCD.print("PAGAR&PINTU", CENTER, 8);
        myGLCD.setFont(TinyFont);
        myGLCD.print("home>pagar&pintu", 0, 0);
        myGLCD.update();

        for (int i=20; i>=0; i--)
        {
            if(nilainpbuka==1){
                myGLCD.drawBitmap(i, 18, GERBANG7buka, 48, 32);
            }
            if(nilainpbuka==0){
                myGLCD.drawBitmap(i, 18, GERBANG7, 48, 32);
            }

            myGLCD.setFont(SmallFont);
            myGLCD.print("PAGAR & PINTU", CENTER, 8);
            myGLCD.setFont(TinyFont);
            myGLCD.print("home>pagar&pintu", 0, 0);
            myGLCD.update();
            delay(10);
        }

        pinv=2;
        if(nilainpbuka==1){
            myGLCD.drawBitmap(45, 27, PANAHKCTa2, 41, 8);
            myGLCD.setFont(SmallFont);
            myGLCD.print("BUKA", 52, 27);
        }

    }
```

```
        }
        if(nilainpinbuka==0){
            myGLCD.drawBitmap(45, 27, PANAHKCTa1, 41, 8);
            myGLCD.setFont(SmallFont);
            myGLCD.print("KUNCI", 49, 27);
        }
        myGLCD.update();
    }

    if(digitalRead(d) == LOW && pinr==1 && airr==0 && lamr==0 && elexr==0){
        pinr=0;
        pinv=0;

    }

    if(digitalRead(b) == LOW && pinv==2 && nilaipintu==1){

        myGLCD.drawBitmap(0, 18, GERBANG7, 48, 32);
        myGLCD.drawBitmap(45, 27, PANAHKCTa1, 41, 8);
        myGLCD.setFont(SmallFont);
        myGLCD.print("KUNCI", 49, 27);
        myGLCD.update();
        nilaipinbuka=0;
        mySerial.println("10");
        delay(300);
    }
    if(digitalRead(b) == LOW && pinv==2 && nilaipintu==2){

        myGLCD.drawBitmap(0, 18, GERBANG7buka, 48, 32);
        myGLCD.drawBitmap(45, 27, PANAHKCTa2, 41, 8);
        myGLCD.setFont(SmallFont);
        myGLCD.print("BUKA", 52, 27);
        myGLCD.update();
        nilaipinbuka=1;
        mySerial.println("11");
        delay(300);
    }

    if (digitalRead(c) == LOW && pinr==1 && pinv==2){
        myGLCD.drawBitmap(45, 27, PANAHKCTa1, 41, 8);
        myGLCD.setFont(SmallFont);
        myGLCD.print("PAGAR&PINTU", CENTER, 8);
        myGLCD.print("KUNCI", 49, 27);
        myGLCD.setFont(TinyFont);
        myGLCD.print("home>pagar&pintu", 0, 0);
        myGLCD.update();
        nilaipintu=1;

        delay(300);
    }
    if (digitalRead(a) == LOW && pinr==1 && pinv==2){
        myGLCD.drawBitmap(45, 27, PANAHKCTa2, 41, 8);
        myGLCD.setFont(SmallFont);
```

```
myGLCD.print("PAGAR&PINTU", CENTER, 8);
myGLCD.print("BUKA", 52, 27);
myGLCD.setFont(TinyFont);
myGLCD.print("home>pagar&pintu", 0, 0);
myGLCD.update();
nilaipintu=2;

delay(300);
}

//coding lampu
if(lamr==1 && lamv==1){
myGLCD.clrScr();
myGLCD.drawBitmap(33, 16, LAMPUPTT, 24, 32);
myGLCD.setFont(SmallFont);
myGLCD.print("LAMPU", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lampu", 0, 0);
myGLCD.update();

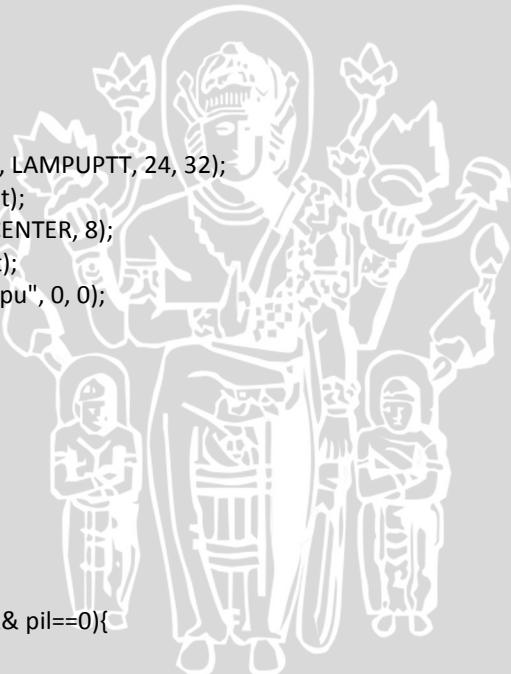
myGLCD.clrScr();
for (int i=33; i>=13; i--)
{
myGLCD.drawBitmap(i, 16, LAMPUPTT, 24, 32);
myGLCD.setFont(SmallFont);
myGLCD.print("LAMPU", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lampu", 0, 0);
myGLCD.update();
delay(10);

}
lamv=2;
}

if(lamr==1 && lamv==2){

if(digitalRead(a) == LOW && pil==0){
if(pl==5){
pl=1;
}
else{pl=pl+1;}
delay(300);
}
if(digitalRead(c) == LOW && pil==0){
if(pl==1 || pl==0){
pl=5;
}
else{pl=pl-1;}
delay(300);
}

if(digitalRead(d) == LOW && pil==0){
```



```
lamr=0;
lamv=0;
delay(300);
}

if(digitalRead(b) == LOW && pil==0){
if(pl==1){
if(niltamunyala==1){
myGLCD.clrScr();
myGLCD.drawBitmap(13, 16, LAMPUPTT, 24, 32);
myGLCD.drawBitmap(43, 27, PANAHKCT12, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("NYALA", 48, 27);
myGLCD.print("RUANG TAMU", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lamp..>ruan..", 0, 0);
myGLCD.update();
pil=1;
}
if(niltamunyala==0){
myGLCD.clrScr();
myGLCD.drawBitmap(13, 16, LAMPUHTT, 24, 32);
myGLCD.drawBitmap(43, 27, PANAHKCT11, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("MATI", 52, 27);
myGLCD.print("RUANG TAMU", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lamp..>ruan..", 0, 0);
myGLCD.update();
pil=1;
}

delay(300);
}

if(pl==2){
if(nilkamar1nyala==1){
myGLCD.clrScr();
myGLCD.drawBitmap(13, 16, LAMPUPTT, 24, 32);
myGLCD.drawBitmap(43, 27, PANAHKCT12, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("NYALA", 48, 27);
myGLCD.print("KAMAR TIDUR 1", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lamp..>kama..", 0, 0);
myGLCD.update();
pil=2;
}
if(nilkamar1nyala==0){
myGLCD.clrScr();
myGLCD.drawBitmap(13, 16, LAMPUHTT, 24, 32);
myGLCD.drawBitmap(43, 27, PANAHKCT11, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("MATI", 52, 27);
```

```
myGLCD.print("KAMAR TIDUR 1", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lamp..>kama..", 0, 0);
myGLCD.update();
pil=2;
}
delay(300);
}
if(pl==3{
if(nilkamar2nyala==1){
myGLCD.clrScr();
myGLCD.drawBitmap(13, 16, LAMPUPTT, 24, 32);
myGLCD.drawBitmap(43, 27, PANAHKCT12, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("NYALA", 48, 27);
myGLCD.print("KAMAR TIDUR 2", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lamp..>kama..", 0, 0);
myGLCD.update();
pil=3;
}
if(nilkamar2nyala==0){
myGLCD.clrScr();
myGLCD.drawBitmap(13, 16, LAMPUHTT, 24, 32);
myGLCD.drawBitmap(43, 27, PANAHKCT11, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("MATI", 52, 27);
myGLCD.print("KAMAR TIDUR 2", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lamp..>kama..", 0, 0);
myGLCD.update();
pil=3;
}
delay(300);
}

if(pl==4){
if(nilmandinyala==1){
myGLCD.clrScr();
myGLCD.drawBitmap(13, 16, LAMPUPTT, 24, 32);
myGLCD.drawBitmap(43, 27, PANAHKCT12, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("NYALA", 48, 27);
myGLCD.print("KAMAR MANDI", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lamp..>kama..", 0, 0);
myGLCD.update();
pil=4;
}
if(nilmandinyala==0){
myGLCD.clrScr();
myGLCD.drawBitmap(13, 16, LAMPUHTT, 24, 32);
myGLCD.drawBitmap(43, 27, PANAHKCT11, 41, 8);
myGLCD.setFont(SmallFont);}
```

```
myGLCD.print("MATI", 52, 27);
myGLCD.print("KAMAR MANDI", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lamp..>kama..", 0, 0);
myGLCD.update();
pil=4;
}
delay(300);
}

if(pl==5){
if(nilterasnyala==1){
myGLCD.clrScr();
myGLCD.drawBitmap(13, 16, LAMPUPTT, 24, 32);
myGLCD.drawBitmap(43, 27, PANAHKCT12, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("NYALA", 48, 27);
myGLCD.print("TERAS", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lampu>teras", 0, 0);
myGLCD.update();
pil=5;
}
if(nilterasnyala==0){
myGLCD.clrScr();
myGLCD.drawBitmap(13, 16, LAMPUHTT, 24, 32);
myGLCD.drawBitmap(43, 27, PANAHKCT11, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("MATI", 52, 27);
myGLCD.print("TERAS", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lampu>teras", 0, 0);
myGLCD.update();
pil=5;
}
delay(300);
}

if(pl==1 && pil==0){
//PILIHAN RUANG TAMU
myGLCD.clrScr();
if(niltamunyala==0){
myGLCD.drawBitmap(13, 16, LAMPUHTT, 24, 32);
}
if(niltamunyala==1){
myGLCD.drawBitmap(13, 16, LAMPUPTT, 24, 32);
}
myGLCD.drawBitmap(34, 27, PANAHKCT, 49, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("RUANG", 43, 23);
myGLCD.print("TAMU", 45, 31);
myGLCD.print("LAMPU", CENTER, 8);
myGLCD.setFont(TinyFont);
```

```
myGLCD.print("home>lampu", 0, 0);
myGLCD.update();

}

if(pl==1 && pil==1 && digitalRead(a) == LOW){
//PILIHAN
myGLCD.drawBitmap(43, 27, PANAHKCT12, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("NYALA", 48, 27);
myGLCD.print("RUANG TAMU", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lamp..>ruan..", 0, 0);
myGLCD.update();
//delay(3000);
niltamu=1;
delay(300);
}

if(pl==1 && pil==1 && digitalRead(c) == LOW){

myGLCD.drawBitmap(43, 27, PANAHKCT11, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("MATI", 52, 27);
myGLCD.print("RUANG TAMU", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lamp..>ruan..", 0, 0);
myGLCD.update();
//delay(3000);
niltamu=2;
delay(300);
}

if(pl==1 && pil==1 && digitalRead(d) == LOW{
pil=0;
delay(300);
}

if(pl==2 && pil==0){
//PILIHAN KAMAR TIDUR 1
myGLCD.clrScr();
if(nilkamar1nyala==0){
myGLCD.drawBitmap(13, 16, LAMPUHTT, 24, 32);
}
if(nilkamar1nyala==1){
myGLCD.drawBitmap(13, 16, LAMPUPTT, 24, 32);
}
myGLCD.drawBitmap(34, 27, PANAHKCT, 49, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("KAMAR", 43, 23);
myGLCD.print("TIDUR", 40, 31);
myGLCD.print("1", 72, 31);
```

```
myGLCD.print("LAMPU", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lampu", 0, 0);
myGLCD.update();

}

if(pl==2 && pil==2 && digitalRead(a) == LOW){
//PILIHAN

myGLCD.drawBitmap(43, 27, PANAHKCT12, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("NYALA", 48, 27);
myGLCD.print("KAMAR TIDUR 1", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lamp..>kama..", 0, 0);
myGLCD.update();

nilkamar1=1;
delay(300);
}
if(pl==2 && pil==2 && digitalRead(c) == LOW){

myGLCD.drawBitmap(43, 27, PANAHKCT11, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("MATI", 52, 27);
myGLCD.print("KAMAR TIDUR 1", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lamp..>kama..", 0, 0);
myGLCD.update();
//delay(3000);
nilkamar1=2;
delay(300);
}

if(pl==2 && pil==2 && digitalRead(d) == LOW){
pil=0;
delay(300);
}

if(pl==3 && pil==0){
//PILIHAN KAMAR TIDUR 2
myGLCD.clrScr();
if(nilkamar2nyala==0){
myGLCD.drawBitmap(13, 16, LAMPUHTT, 24, 32);
}
if(nilkamar2nyala==1){
myGLCD.drawBitmap(13, 16, LAMPUPTT, 24, 32);
}
myGLCD.drawBitmap(34, 27, PANAHKCT, 49, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("KAMAR", 43, 23);
myGLCD.print("TIDUR", 39, 31);
```



```
myGLCD.print("2", 71, 31);
myGLCD.print("LAMPU", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lampu", 0, 0);
myGLCD.update();

}

if(pl==3 && pil==3 && digitalRead(a) == LOW){
//PILIHAN

myGLCD.drawBitmap(43, 27, PANAHKCT12, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("NYALA", 48, 27);
myGLCD.print("KAMAR TIDUR 2", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lamp..>kama..", 0, 0);
myGLCD.update();

nilkamar2=1;
delay(300);
}
if(pl==3 && pil==3 && digitalRead(c) == LOW){

myGLCD.drawBitmap(43, 27, PANAHKCT11, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("MATI", 52, 27);
myGLCD.print("KAMAR TIDUR 2", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lamp..>kama..", 0, 0);
myGLCD.update();

nilkamar2=2;
delay(300);
}

if(pl==3 && pil==3 && digitalRead(d) == LOW){
pil=0;
delay(300);
}

if(pl==4 && pil==0){
//PILIHAN KAMAR MANDI
myGLCD.clrScr();
if(nilmandinyala==0){
myGLCD.drawBitmap(13, 16, LAMPUHTT, 24, 32);
}
if(nilmandinyala==1){
myGLCD.drawBitmap(13, 16, LAMPUPTT, 24, 32);
}
myGLCD.drawBitmap(34, 27, PANAHKCT, 49, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("KAMAR", 43, 23);
myGLCD.print("MANDI", 44, 31);
```



```
myGLCD.print("LAMPU", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lampu", 0, 0);
myGLCD.update();

}

if(pl==4 && pil==4 && digitalRead(a) == LOW){
//PILIHAN

myGLCD.drawBitmap(43, 27, PANAHKCT12, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("NYALA", 48, 27);
myGLCD.print("KAMAR MANDI", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lamp..>kama..", 0, 0);
myGLCD.update();

nilmandi=1;
delay(300);
}

if(pl==4 && pil==4 && digitalRead(c) == LOW){

myGLCD.drawBitmap(43, 27, PANAHKCT11, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("MATI", 52, 27);
myGLCD.print("KAMAR MANDI", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lamp..>kama..", 0, 0);
myGLCD.update();

nilmandi=2;
delay(300);
}

if(pl==4 && pil==4 && digitalRead(d) == LOW){
pil=0;
delay(300);
}

if(pl==5 && pil==0){
//PILIHAN TERAS
myGLCD.clrScr();
if(nilterasnyala==0){
myGLCD.drawBitmap(13, 16, LAMPUHTT, 24, 32);
}
if(nilterasnyala==1){
myGLCD.drawBitmap(13, 16, LAMPUPTT, 24, 32);
}
myGLCD.drawBitmap(34, 27, PANAHKCT, 49, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("TERAS", 43, 27);
myGLCD.print("LAMPU", CENTER, 8);
myGLCD.setFont(TinyFont);
```

```
myGLCD.print("home>lampu", 0, 0);
myGLCD.update();
//delay(3000);
}

if(pl==5 && pil==5 && digitalRead(a) == LOW){
//PILIHAN

myGLCD.drawBitmap(43, 27, PANAHKCT12, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("NYALA", 48, 27);
myGLCD.print("TERAS", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lampu>teras", 0, 0);
myGLCD.update();
//delay(3000);
nilteras=1;
delay(300);
}

if(pl==5 && pil==5 && digitalRead(c) == LOW){
myGLCD.drawBitmap(43, 27, PANAHKCT11, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("MATI", 52, 27);
myGLCD.print("TERAS", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>lampu>teras", 0, 0);
myGLCD.update();
//delay(3000);
nilteras=2;
delay(300);
}

if(pl==5 && pil==5 && digitalRead(d) == LOW){
pil=0;
delay(300);
}

//nilai lampu
//1
if(digitalRead(b) == LOW && pil==1 && pl==1 && niltamu==1){
myGLCD.drawBitmap(13, 16, LAMPUPTT, 24, 32);
myGLCD.update();
niltamunyala=1;
mySerial.println("211");
delay(300);
}
if(digitalRead(b) == LOW && pil==1 && pl==1 && niltamu==2){
myGLCD.drawBitmap(13, 16, LAMPUHTT, 24, 32);
myGLCD.update();
niltamunyala=0;
mySerial.println("210");
delay(300);
}
```

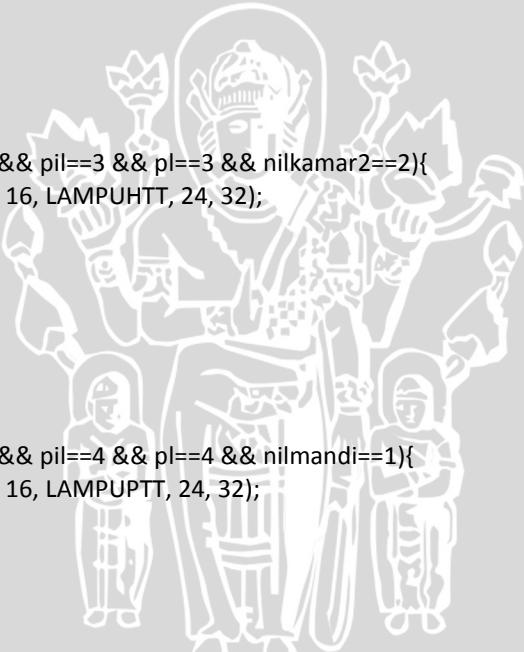


```
//2
if(digitalRead(b) == LOW && pil==2 && pl==2 && nilkamar1==1){
myGLCD.drawBitmap(13, 16, LAMPUPTT, 24, 32);
myGLCD.update();
nilkamar1nyala=1;
mySerial.println("221");
delay(300);
}
if(digitalRead(b) == LOW && pil==2 && pl==2 && nilkamar1==2){
myGLCD.drawBitmap(13, 16, LAMPUHTT, 24, 32);
myGLCD.update();
nilkamar1nyala=0;
mySerial.println("220");
delay(300);
}

//3
if(digitalRead(b) == LOW && pil==3 && pl==3 && nilkamar2==1){
myGLCD.drawBitmap(13, 16, LAMPUPTT, 24, 32);
myGLCD.update();
nilkamar2nyala=1;
mySerial.println("231");
delay(300);
}
if(digitalRead(b) == LOW && pil==3 && pl==3 && nilkamar2==2){
myGLCD.drawBitmap(13, 16, LAMPUHTT, 24, 32);
myGLCD.update();
nilkamar2nyala=0;
mySerial.println("230");
delay(300);
}

//4
if(digitalRead(b) == LOW && pil==4 && pl==4 && nilmandi==1){
myGLCD.drawBitmap(13, 16, LAMPUPTT, 24, 32);
myGLCD.update();
hilmandinyala=1;
mySerial.println("241");
delay(300);
}
if(digitalRead(b) == LOW && pil==4 && pl==4 && nilmandi==2){
myGLCD.drawBitmap(13, 16, LAMPUHTT, 24, 32);
myGLCD.update();
hilmandinyala=0;
mySerial.println("240");
delay(300);
}

//5
if(digitalRead(b) == LOW && pil==5 && pl==5 && nilteras==1){
myGLCD.drawBitmap(13, 16, LAMPUPTT, 24, 32);
myGLCD.update();
nilterasnyala=1;
mySerial.println("251");
```



```
delay(300);
}
if(digitalRead(b) == LOW && pil==5 && pl==5 && nilteras==2){
myGLCD.drawBitmap(13, 16, LAMPUHTT, 24, 32);
myGLCD.update();
nilterasnalya=0;
mySerial.println("250");
delay(300);
}
}

//elektronik

if(elexr==1 && elexv==1){

if(digitalRead(d)==LOW && tvr==0 && nasir==0 && acr==0){
elexr=0;
elexv=0;
tombolelex=0;

}

if (digitalRead(a) == LOW && tvr==0 && nasir==0 && acr==0){
if(tombolelex==2){
tombolelex=0;
}
else{tombolelex=tombolelex+1;}
delay(300);
}

if(digitalRead(c) == LOW && tvr==0 && nasir==0 && acr==0){
if(tombolelex==0){
tombolelex=2;
}
else{tombolelex=tombolelex-1;}
delay(300);
}

if(digitalRead(b) == LOW && tvr==0 && nasir==0 && acr==0){
if(tombolelex==0){
tvr=1;
tvv=1;
delay(300);
}
if(tombolelex==1){
nasir=1;
nasiv=1;
delay(300);
}
if(tombolelex==2){
acr=1;
acv=1;
delay(300);
}
```



```
}

if (tombolelex==0 && tvr==0 && nasir==0 && acr==0){
myLCD.clrScr();
if(nilaitvhidup==0){
myLCD.drawBitmap(0, 8, tvmmati, 84, 48);
}
if(nilaitvhidup==1){
myLCD.drawBitmap(0, 8, TVI, 84, 48);
}
myLCD.setFont(SmallFont);
myLCD.print("TV", CENTER, 8);
myLCD.setFont(TinyFont);
myLCD.print("home>elektronik", 0, 0);
myLCD.update();
}

if (tombolelex==1 && tvr==0 && nasir==0 && acr==0){
myLCD.clrScr();
if(nilainasihidup==0){
myLCD.drawBitmap(0, 8, ricecookermati, 84, 48);
}
if(nilainasihidup==1){
myLCD.drawBitmap(0, 8, RICE1, 84, 48);
}
myLCD.setFont(SmallFont);
myLCD.print("PENANAK NASI", CENTER, 8);
myLCD.setFont(TinyFont);
myLCD.print("home>elektronik", 0, 0);
myLCD.update();
}

if (tombolelex==2 && tvr==0 && nasir==0 && acr==0){
myLCD.clrScr();
if(nilaiachidup==0){
myLCD.drawBitmap(0, 8, ac1mati, 84, 48);
}
if(nilaiachidup==1){
myLCD.drawBitmap(0, 8, AC1, 84, 48);
}
myLCD.setFont(SmallFont);
myLCD.print("AC", CENTER, 8);
myLCD.setFont(TinyFont);
myLCD.print("home>elektronik", 0, 0);
myLCD.update();
}

//coding tv
if(tvr==1 && tvv==1){
myLCD.clrScr();
if(nilaitvhidup==0){
myLCD.drawBitmap(26, 20, tvmati, 34, 24);
}
```

```
if(nilaitvhidup==1){  
    myGLCD.drawBitmap(26, 20, TV, 34, 24);  
}  
  
myGLCD.setFont(SmallFont);  
myGLCD.print("TV", CENTER, 8);  
myGLCD.setFont(TinyFont);  
myGLCD.print("home>elektronik>tv", 0, 0);  
myGLCD.update();  
  
for (int i=26; i>=5; i--)  
{  
if(nilaitvhidup==0){  
    myGLCD.drawBitmap(i, 20, tvmati, 34, 24);  
}  
if(nilaitvhidup==1){  
    myGLCD.drawBitmap(i, 20, TV, 34, 24);  
}  
myGLCD.setFont(SmallFont);  
myGLCD.print("TV", CENTER, 8);  
myGLCD.setFont(TinyFont);  
myGLCD.print("home>elektronik>tv", 0, 0);  
myGLCD.update();  
delay(10);  
}  
tvv=2;  
if(nilaitvhidup==0){  
    myGLCD.drawBitmap(43, 27, PANAHKCT21, 41, 8);  
    myGLCD.setFont(SmallFont);  
    myGLCD.print("MATI", 52, 27);  
    myGLCD.update();  
}  
if(nilaitvhidup==1){  
    myGLCD.drawBitmap(43, 27, PANAHKCT22, 41, 8);  
    myGLCD.setFont(SmallFont);  
    myGLCD.print("NYALA", 48, 27);  
    myGLCD.update();  
}  
}  
  
if(digitalRead(a)== LOW && tvr==1 && tvv==2){  
    myGLCD.clrScr();  
    if(nilaitvhidup==0){  
        myGLCD.drawBitmap(5, 20, tvmati, 34, 24);  
    }  
    if(nilaitvhidup==1){  
        myGLCD.drawBitmap(5, 20, TV, 34, 24);  
    }  
    myGLCD.drawBitmap(43, 27, PANAHKCT22, 41, 8);  
    myGLCD.setFont(SmallFont);  
    myGLCD.print("NYALA", 48, 27);  
    myGLCD.print("TV", CENTER, 8);  
    myGLCD.setFont(TinyFont);  
    myGLCD.print("home>elektronik>tv", 0, 0);  
}
```

```
myGLCD.update();
nilaitv=1;

}

if(digitalRead(c)== LOW && tvr==1 && tvv==2){
myGLCD.clrScr();
if(nilaitvhidup==0){
myGLCD.drawBitmap(5, 20, tvmati, 34, 24);
}
if(nilaitvhidup==1){
myGLCD.drawBitmap(5, 20, TV, 34, 24);
}
myGLCD.drawBitmap(43, 27, PANAHKCT21, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("MATI", 52, 27);
myGLCD.print("TV", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>elektronik>tv", 0, 0);
myGLCD.update();
nilaitv=0;
}

//coding nasi
if(nasir==1 && nasiv==1){
myGLCD.clrScr();
if(nilainasihidup==0){
myGLCD.drawBitmap(24, 17, ricecooker1mati, 39, 32);
}
if(nilainasihidup==1){
myGLCD.drawBitmap(24, 17, RICE, 39, 32);
}

myGLCD.setFont(SmallFont);
myGLCD.print("PENANAK NASI", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>elek..>penanak..", 0, 0);
myGLCD.update();

for (int i=24; i>=0; i--){
if(nilainasihidup==0){
myGLCD.drawBitmap(i, 17, ricecooker1mati, 39, 32);
}
if(nilainasihidup==1){
myGLCD.drawBitmap(i, 17, RICE, 39, 32);
}
myGLCD.setFont(SmallFont);
myGLCD.print("PENANAK NASI", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>elek..>penanak..", 0, 0);
myGLCD.update();
delay(10);
}
```

```
    }
    nasiv=2;
    if(nilainasihidup==0){
        myGLCD.drawBitmap(43, 27, PANAHKCT21, 41, 8);
        myGLCD.setFont(SmallFont);
        myGLCD.print("MATI", 52, 27);
        myGLCD.update();
    }
    if(nilainasihidup==1){
        myGLCD.drawBitmap(43, 27, PANAHKCT22, 41, 8);
        myGLCD.setFont(SmallFont);
        myGLCD.print("NYALA", 48, 27);
        myGLCD.update();
    }
}

if(digitalRead(a)== LOW && nasir==1 && nasiv==2){
    myGLCD.clrScr();
    if(nilainasihidup==0){
        myGLCD.drawBitmap(0, 17, ricecooker1mati, 39, 32);
    }
    if(nilainasihidup==1){
        myGLCD.drawBitmap(0, 17, RICE, 39, 32);
    }
    myGLCD.drawBitmap(43, 27, PANAHKCT22, 41, 8);
    myGLCD.setFont(SmallFont);
    myGLCD.print("NYALA", 48, 27);
    myGLCD.print("PENANAK NASI", CENTER, 8);
    myGLCD.setFont(TinyFont);
    myGLCD.print("home>elek..>penanak..", 0, 0);
    myGLCD.update();
    nilainasi=1;
}

if(digitalRead(c)== LOW && nasir==1 && nasiv==2){
    myGLCD.clrScr();
    if(nilainasihidup==0){
        myGLCD.drawBitmap(0, 17, ricecooker1mati, 39, 32);
    }
    if(nilainasihidup==1){
        myGLCD.drawBitmap(0, 17, RICE, 39, 32);
    }
    myGLCD.drawBitmap(43, 27, PANAHKCT21, 41, 8);
    myGLCD.setFont(SmallFont);
    myGLCD.print("MATI", 52, 27);
    myGLCD.print("PENANAK NASI", CENTER, 8);
    myGLCD.setFont(TinyFont);
    myGLCD.print("home>elek..>penanak..", 0, 0);
    myGLCD.update();
    nilainasi=0;
}
//coding ac
```

```
if(acr==1 && acv==1){
    myGLCD.clrScr();
    if(nilaiachidup==0){
        myGLCD.drawBitmap(22, 22, ACTmati, 48, 24);
    }
    if(nilaiachidup==1){
        myGLCD.drawBitmap(22, 22, ACT, 48, 24);
    }
    myGLCD.setFont(SmallFont);
    myGLCD.print("AC", CENTER, 8);
    myGLCD.setFont(TinyFont);
    myGLCD.print("home>elektronik>ac", 0, 0);
    myGLCD.update();
    for (int i=22; i>=0; i--)
    {
        if(nilaiachidup==0){
            myGLCD.drawBitmap(i, 22, ACTmati, 48, 24);
        }
        if(nilaiachidup==1){
            myGLCD.drawBitmap(i, 22, ACT, 48, 24);
        }
        myGLCD.setFont(SmallFont);
        myGLCD.print("AC", CENTER, 8);
        myGLCD.setFont(TinyFont);
        myGLCD.print("home>elektronik>ac", 0, 0);
        myGLCD.update();
        delay(10);
    }
    acv=2;
    if(nilaiachidup==0){
        myGLCD.drawBitmap(43, 27, PANAHKCT21, 41, 8);
        myGLCD.setFont(SmallFont);
        myGLCD.print("MATI", 52, 27);
        myGLCD.update();
    }
    if(nilaiachidup==1){
        myGLCD.drawBitmap(43, 27, PANAHKCT22, 41, 8);
        myGLCD.setFont(SmallFont);
        myGLCD.print("NYALA", 48, 27);
        myGLCD.update();
    }
}

if(digitalRead(a)== LOW && acr==1 && acv==2){
    myGLCD.clrScr();
    if(nilaiachidup==0){
        myGLCD.drawBitmap(0, 22, ACTmati, 48, 24);
    }
    if(nilaiachidup==1){
        myGLCD.drawBitmap(0, 22, ACT, 48, 24);
    }
    myGLCD.drawBitmap(43, 27, PANAHKCT22, 41, 8);
    myGLCD.setFont(SmallFont);
    myGLCD.print("NYALA", 48, 27);
```



```
myGLCD.print("AC", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>elektronik>ac", 0, 0);
myGLCD.update();
nilaiac=1;

}

if(digitalRead(c)== LOW && acr==1 && acv==2){
myGLCD.clrScr();
if(nilaiachidup==0){
myGLCD.drawBitmap(0, 22, ACTmati, 48, 24);
}
if(nilaiachidup==1){
myGLCD.drawBitmap(0, 22, ACT, 48, 24);
}
myGLCD.drawBitmap(43, 27, PANAHKCT21, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("MATI", 52, 27);
myGLCD.print("AC", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>elektronik>ac", 0, 0);
myGLCD.update();
nilaiac=0;
}

if(digitalRead(b) == LOW && tvr==1 && tvv==2){
if (nilaitv==1){
myGLCD.clrScr();
myGLCD.drawBitmap(5, 20, TV, 34, 24);
myGLCD.drawBitmap(43, 27, PANAHKCT22, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("NYALA", 48, 27);
myGLCD.print("TV", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>elektronik>tv", 0, 0);
myGLCD.update();
nilaitvhidup=1;
mySerial.println("311");
delay(300);
}
if (nilaitv==0){
myGLCD.clrScr();
myGLCD.drawBitmap(5, 20, tvmati, 34, 24);
myGLCD.drawBitmap(43, 27, PANAHKCT21, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("MATI", 52, 27);
myGLCD.print("TV", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>elektronik>tv", 0, 0);
myGLCD.update();
nilaitvhidup=0;
mySerial.println("310");
delay(300);
}
```

```
    }}

if(digitalRead(b) == LOW && acr==1 && acv==2){
if (nilaiac==1){
    myGLCD.clrScr();
    myGLCD.drawBitmap(0, 22, ACT, 48, 24);
    myGLCD.drawBitmap(43, 27, PANAHKCT22, 41, 8);
    myGLCD.setFont(SmallFont);
    myGLCD.print("NYALA", 48, 27);
    myGLCD.print("AC", CENTER, 8);
    myGLCD.setFont(TinyFont);
    myGLCD.print("home>elektronik>ac", 0, 0);
    myGLCD.update();
    nilaiachidup=1;
    mySerial.println("321");
    delay(300);
}
if (nilaiac==0){
    myGLCD.clrScr();
    myGLCD.drawBitmap(0, 22, ACTmati, 48, 24);
    myGLCD.drawBitmap(43, 27, PANAHKCT21, 41, 8);
    myGLCD.setFont(SmallFont);
    myGLCD.print("MATI", 52, 27);
    myGLCD.print("AC", CENTER, 8);
    myGLCD.setFont(TinyFont);
    myGLCD.print("home>elektronik>ac", 0, 0);
    myGLCD.update();
    nilaiachidup=0;
    mySerial.println("320");
    delay(300);
}

if(digitalRead(b) == LOW && nasir==1 && nasiv==2){
if (nilainasi==1){
    myGLCD.clrScr();
    myGLCD.drawBitmap(0, 17, RICE, 39, 32);
    myGLCD.drawBitmap(43, 27, PANAHKCT22, 41, 8);
    myGLCD.setFont(SmallFont);
    myGLCD.print("NYALA", 48, 27);
    myGLCD.print("RICECOOKER", CENTER, 8);
    myGLCD.setFont(TinyFont);
    myGLCD.print("home>elek..>penanak..", 0, 0);
    myGLCD.update();
    nilainasihidup=1;
    mySerial.println("331");
    delay(300);
}
if (nilainasi==0){
    myGLCD.clrScr();
    myGLCD.drawBitmap(0, 17, ricecooker1mati, 39, 32);
    myGLCD.drawBitmap(43, 27, PANAHKCT21, 41, 8);
    myGLCD.setFont(SmallFont);
```



```
myGLCD.print("MATI", 52, 27);
myGLCD.print("RICECOOKER", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>elek..>penanak..", 0, 0);
myGLCD.update();
nilainasihidup=0;
mySerial.println("330");
delay(300);
}
}
if(digitalRead(d)==LOW && tvr==1 && tvv==2){
tvr=0;
tvv=0;
delay(300);
}
if(digitalRead(d)==LOW && acr==1 && acv==2){
acr=0;
acv=0;
delay(300);
}
if(digitalRead(d)==LOW && nasir==1 && nasiv==2){
nasir=0;
nasiv=0;
delay(300);
}
}

//coding air
if(airr==1 && airv==1){
myGLCD.clrScr();
if(nilaiairnyala==1){
myGLCD.drawBitmap(20, 18, air, 35, 32);
}
if(nilaiairnyala==0){
myGLCD.drawBitmap(20, 18, air1, 35, 32);
}
myGLCD.setFont(SmallFont);
myGLCD.print("PENYIRAM", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>penyiram", 0, 0);
myGLCD.update();
delay(300);
for (int i=20; i>=5; i--)
{
if(nilaiairnyala==1){
myGLCD.drawBitmap(i, 18, air, 35, 32);
}
if(nilaiairnyala==0){
myGLCD.drawBitmap(i, 18, air1, 35, 32);
}
myGLCD.setFont(SmallFont);
myGLCD.print("PENYIRAM", CENTER, 8);
myGLCD.setFont(TinyFont);
myGLCD.print("home>penyiram", 0, 0);
```



```
myGLCD.update();
delay(10);
airv=2;
}
if(nilaiairnyala==1){
myGLCD.drawBitmap(45, 27, PANAHKCTa2, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("NYALA", 49, 27);
myGLCD.update();
}
if(nilaiairnyala==0){
myGLCD.drawBitmap(45, 27, PANAHKCTa1, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("MATI", 52, 27);
myGLCD.update();
}
}

if (digitalRead(a) == LOW && airr==1 && airv==2){
myGLCD.drawBitmap(45, 27, PANAHKCTa2, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("PENYIRAM", CENTER, 8);
myGLCD.print("NYALA", 49, 27);
myGLCD.setFont(TinyFont);
myGLCD.print("home>penyiram", 0, 0);
myGLCD.update();
nilaiair=1;
delay(300);
}

if (digitalRead(c) == LOW && airr==1 && airv==2){
myGLCD.drawBitmap(45, 27, PANAHKCTa1, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("PENYIRAM", CENTER, 8);
myGLCD.print("MATI", 52, 27);
myGLCD.setFont(TinyFont);
myGLCD.print("home>penyiram", 0, 0);
myGLCD.update();
nilaiair=2;
delay(300);
}

if (digitalRead(b) == LOW && airr==1 && airv==2){
if(nilaiair==2){
myGLCD.drawBitmap(5, 18, air1, 35, 32);
myGLCD.drawBitmap(45, 27, PANAHKCTa1, 41, 8);
myGLCD.setFont(SmallFont);
myGLCD.print("MATI", 52, 27);
myGLCD.update();
nilaiairnyala=0;
mySerial.println("40");
delay(300);
}
```

```
if(nilaiair==1){  
    myLCD.drawBitmap(5, 18, air, 35, 32);  
    myLCD.drawBitmap(45, 27, PANAHKCTa2, 41, 8);  
    myLCD.setFont(SmallFont);  
    myLCD.print("NYALA", 49, 27);  
    myLCD.update();  
    nilaiairnyala=1;  
    mySerial.println("41");  
    delay(300);  
}  
if(digitalRead(d) == LOW && airr==1 && airv==2){  
    airr=0;  
    airv=0;  
    delay(300);  
}  
}
```

A.1.5 Icon

```
#include <avr/pgmspace.h>  
  
const uint8_t menuair [] PROGMEM= {  
0x00, 0x80, 0xC0, 0x60, 0x30, 0x90, 0xF0,  
0x60, 0x30, 0x10, 0x30, 0x60, 0xC0, 0x80, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x0E, 0x3B, 0x79, 0xFC, 0xFE, 0xFF, 0xFE, 0xFE, 0xFE, 0xFC, 0xF0, 0xF8,  
0x6D, 0x67, 0x60, 0x60, 0xF0, 0x70, 0x30, 0x10, 0x40, 0x00, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x01, 0x03, 0x07, 0x07, 0x03, 0x03, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x05, 0x00, 0xA, 0x00, 0x15, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};  
};  
  
const uint8_t menuelektr [] PROGMEM= {  
0x00,  
0x80, 0xC0, 0xC0, 0x80, 0xC0, 0xE0, 0x70, 0x38, 0x18, 0x00, 0x00, 0x80, 0x80, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};  
};  
  
const uint8_t menugerbang [] PROGMEM= {  
0x00,  
0xC0, 0x60, 0x20, 0x30, 0x10, 0x10, 0x30, 0x20, 0x60, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00},  
};
```



```
};

const uint8_t menulampu [] PROGMEM= {
0x00, 0x80, 0x40,
0x20, 0x10, 0x08, 0x08, 0x08, 0x08, 0x10, 0x20, 0x40, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0x18, 0x60, 0x80, 0x00, 0x00, 0x00,
0x00, 0x80, 0x60, 0x18, 0x07, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x0D, 0x15, 0x15, 0x15, 0x0D, 0x03, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
};

const uint8_t menuairinvert [] PROGMEM= {
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x7F, 0x3F, 0x9F, 0xCF, 0x6F, 0x0F,
0x1F, 0x0F, 0x0F, 0x1F, 0x3F, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xF1, 0xC4, 0x96, 0x3B, 0x7D, 0xFE, 0xFE, 0xFE, 0xFE, 0xFE, 0x7E, 0x7C, 0x00,
0x98, 0x98, 0x9F, 0x9F, 0x0F, 0x8F, 0xCF, 0xEF, 0xBF, 0xFF, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFC, 0xF9, 0xF9, 0xFD, 0xFC, 0xFE, 0xFE, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFA, 0xF5, 0xFF, 0xEA, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
};

const uint8_t menuelektroinvert [] PROGMEM= {
0xFF, 0xFF,
0x7F, 0x3F, 0x3F, 0x7F, 0x3F, 0x1F, 0x8F, 0xC7, 0xE7, 0xFF, 0xFF, 0x7F, 0x7F, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x03, 0x07, 0x03, 0x91, 0xF8, 0xFC, 0xFE, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xE7, 0xE3,
0xE1, 0xF0, 0xF8, 0xFC, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
};

const uint8_t menugerbanginvert [] PROGMEM= {
0xFF, 0xFF,
0x3F, 0x9F, 0xDF, 0xCF, 0xEF, 0xEF, 0xCF, 0xDF, 0x9F, 0x3F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x07, 0x03, 0x00, 0x03, 0x03, 0x63, 0x63,
0x03, 0x03, 0x03, 0x00, 0x03, 0x07, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
};

const uint8_t menulampuinvert [] PROGMEM= {
0xFF, 0xFF,
0xDF, 0xEF, 0xF7, 0xF7, 0xF7, 0xF7, 0xF7, 0xEF, 0xDF, 0xBF, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x7F, 0x9F, 0xE7, 0xF8, 0xE7, 0x9F, 0x7F, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
```

```

0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFC, 0xF2, 0xEA, 0xEA, 0xF2, 0xFC, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
};

const uint8_t menuairmatiinvert [] PROGMEM= {
0xFF, 0x7F, 0x3F, 0x9F, 0xCF, 0x6F, 0x0F,
0x1F, 0x0F, 0x0F, 0x0F, 0x1F, 0x3F, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xF1, 0xC4, 0x86, 0x03, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x90, 0x98, 0x9F, 0x9F, 0x0F, 0x8F, 0xCF, 0xEF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFE, 0xFC, 0xF8, 0xF8, 0xFC, 0xFE, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
};

const uint8_t menuairmati [] PROGMEM= {
0x00, 0x80, 0xC0, 0x60, 0x30, 0x90, 0xF0,
0xE0, 0xF0, 0xF0, 0xE0, 0xC0, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x0E, 0x3B, 0x79, 0xFC, 0xFE, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0x6F, 0x67, 0x60, 0x60, 0xF0, 0x70, 0x30, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x01, 0x03, 0x07, 0x07, 0x03, 0x03, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
};

const uint8_t menugerbangbukainvert [] PROGMEM= {
0xFF, 0xFF,
0x1F, 0xCF, 0xEF, 0xE7, 0xF7, 0xF7, 0xE7, 0xEF, 0xCF, 0x1F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xF0, 0xE0, 0xE0, 0xE0, 0xE0, 0xE7, 0xE4, 0xE0, 0xE0, 0xE0, 0xE0, 0xE0, 0xE0, 0xE0,
0xFF, 0xFF,
};

const uint8_t menugerbangbuka [] PROGMEM= {
0x00, 0x00,
0xE0, 0x30, 0x10, 0x18, 0x08, 0x08, 0x18, 0x10, 0x30, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xF0, 0xF8, 0xFF, 0xF8, 0xF8, 0xF8, 0x38, 0x38, 0x38,
0xF8, 0xF8, 0xF8, 0xF9, 0xF8, 0xF0, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x0F, 0x1F, 0x1F, 0x1F, 0x1F, 0x18, 0x1B, 0x1F, 0x1F, 0x1F, 0x1F, 0x1F, 0x0F,
0x00, 0x00,
};

const uint8_t PANAHKCTa1[] PROGMEM= {
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0xFF, 0x7E, 0x3C, 0x18,
}
```

```
};

const uint8_t PANAHKCTa2[] PROGMEM={
0x18, 0x3C, 0x7E, 0xFF, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
};

const uint8_t air[] PROGMEM= {
0x00, 0x80, 0xC0, 0xE0, 0x70, 0x38, 0x9C, 0xCE, 0x66, 0x3E, 0x0C, 0x0C, 0x0C, 0x08, 0x18, 0x10,
0x30, 0xE0, 0x80, 0x00, 0x00,
0x00, 0x00, 0x00, 0x3F, 0x7B, 0xF9, 0xFC, 0xFE, 0xFF, 0xFF, 0xFE, 0xFE, 0xFE, 0xFE, 0xFE,
0xFE, 0xFC, 0xF8, 0xE0, 0xF8, 0xEF, 0xE6, 0xE0, 0xE0, 0xFO, 0xF0, 0x70, 0x30, 0x10, 0x40,
0x00, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x03, 0x07, 0x0F, 0x1F, 0x1F, 0x0F,
0x0F, 0x07, 0x07, 0x03, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00,
0x00, 0x05, 0x00, 0xA, 0x00, 0x15, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
};

const uint8_t air1[] PROGMEM= {
0x00, 0x80, 0xC0, 0xE0, 0x70, 0x38, 0x9C, 0xCE, 0xE6, 0xFE, 0xFC, 0xFC, 0xF8, 0xF8, 0xF0,
0xF0, 0xE0, 0x80, 0x00, 0x00,
0x00, 0x00, 0x00, 0x3F, 0x7B, 0xF9, 0xFC, 0xFE, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xE6, 0xE0, 0xE0, 0xFO, 0xF0, 0x70, 0x30, 0x10, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x03, 0x07, 0x0F, 0x1F, 0x1F, 0x0F,
0x0F, 0x07, 0x07, 0x03, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
};

const uint8_t LAMPUPTT[] PROGMEM= {
0x80, 0x60, 0x18, 0x04, 0x04, 0x02, 0x02, 0x01, 0x01, 0x01, 0x01, 0x02, 0x02, 0x04, 0x04, 0x18,
0x60, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0x3F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x03, 0x1C, 0x60, 0x40, 0x40, 0x40, 0x40, 0x40, 0x40, 0x60, 0x1C, 0x03,
0x00, 0x03, 0x06, 0x36,
0x66, 0x66, 0x66, 0x36, 0x06, 0x03, 0x00, 0x00
};

const uint8_t LAMPUHTT[] PROGMEM= {
0x80, 0xE0, 0xF8, 0xFC, 0xFC, 0xFE, 0xFE, 0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0xFE, 0xFC, 0xFC, 0xF8,
0xE0, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x03, 0x1F, 0x7F, 0x1F, 0x03,
0x00, 0x03, 0x06, 0x36,
0x66, 0x66, 0x66, 0x36, 0x06, 0x03, 0x00, 0x00
};

const uint8_t LAMPUPUT[] PROGMEM= {
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};
```



A.1.6 Maket

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(8,9);
char n;
int pintu1=1;
int pintu2=2;

int lt=13;
int lk1=12;
int lk2=11;
int lm=10;
int lr=7;

int et=6;
int er=5;
int ea=4;

int a=3;

void setup()
{
    // Open serial communications and wait for port to open:

    pinMode(pintu1,OUTPUT);
    pinMode(pintu2,OUTPUT);

    pinMode(lt,OUTPUT);
    pinMode(lk1,OUTPUT);
    pinMode(lk2,OUTPUT);
    pinMode(lm,OUTPUT);
    pinMode(lr,OUTPUT);

    pinMode(et,OUTPUT);
    pinMode(er,OUTPUT);
    pinMode(ea,OUTPUT);

    pinMode(a,OUTPUT);

    //Serial.begin(9600);
    //while (!Serial) {
        // // wait for serial port to connect. Needed for Leonardo only
    //}

    //Serial.println("Goodnight moon!");

    // set the data rate for the SoftwareSerial port
    mySerial.begin(9600);
    //mySerial.println("Hello, world?");
}

void loop() // run over and over
{
```



```
if (mySerial.available()){\n    n=mySerial.read();\n    delay(100);\n    Serial.write(n);\n}\n\n//gerbang\nif (n=='1'){\n    n=mySerial.read();\n    if(n=='0'){\n        Serial.println("gerbangtutup");\n        digitalWrite(pintu1, LOW);\n        digitalWrite(pintu2, HIGH);\n    }if(n=='1'){\n        Serial.println("gerbangbuka");\n        digitalWrite(pintu1, HIGH);\n        digitalWrite(pintu2, LOW);\n    }\n}\n\n//lampa\nif (n=='2'){\n    n=mySerial.read();\n    if(n=='1'){\n        n=mySerial.read();\n        if(n=='0'){\n            Serial.println("tamuo");\n            digitalWrite(lt, LOW);\n        }\n        if(n=='1'){\n            Serial.println("tamu1");\n            digitalWrite(lt, HIGH);\n        }\n    if(n=='2'){\n        n=mySerial.read();\n        if(n=='0'){\n            Serial.println("kama10");\n            digitalWrite(lk1, LOW);\n        }\n        if(n=='1'){\n            Serial.println("kama11");\n            digitalWrite(lk1, HIGH);\n        }\n    if(n=='3'){\n        n=mySerial.read();\n        if(n=='0'){\n            Serial.println("kama20");\n            digitalWrite(lk2, LOW);\n        }\n        if(n=='1'){\n            Serial.println("kama21");\n            digitalWrite(lk2, HIGH);\n        }\n    if(n=='4'){\n        n=mySerial.read();\n    }\n}
```



```
if(n=='0'){
Serial.println("mandi0");
digitalWrite(lm, LOW);
}
if(n=='1'){
Serial.println("mandi1");
digitalWrite(lm, HIGH);
}
if(n=='5'){
n=mySerial.read();
if(n=='0'){
Serial.println("ter0");
digitalWrite(lr, LOW);
}
if(n=='1'){
Serial.println("ter1");
digitalWrite(lr, HIGH);
}
}

//elektronik
//lampa
if (n== '3'){
n=mySerial.read();
if(n=='1'){
n=mySerial.read();
if(n=='0'){
Serial.println("tv0");
digitalWrite(et, LOW);
}
if(n=='1'){
Serial.println("tv1");
digitalWrite(et, HIGH);
}
if(n=='2'){
n=mySerial.read();
if(n=='0'){
Serial.println("ac0");
digitalWrite(ea, LOW);
}
if(n=='1'){
Serial.println("ac1");
digitalWrite(ea, HIGH);
}
if(n=='3'){
n=mySerial.read();
if(n=='0'){
Serial.println("rice0");
digitalWrite(er, LOW);
}
if(n=='1'){
Serial.println("rice1");
digitalWrite(er, HIGH);
}
}
```



```
    }
    //air
    if (n== '4'){
        n=mySerial.read();
    if(n=='0'){
        Serial.println("air0");
        digitalWrite(a, LOW);
    }if(n=='1'){
        Serial.println("air1");
        digitalWrite(a, HIGH);
    }
    //if (mySerial.write(mySerial.read())== 'kama21'){
    //Serial.println("1");
    }
}
```

UNIVERSITAS BRAWIJAYA



A.2 Sampel Questioner

Nama: Hera
 Jenis kelamin: L/P
 Umur: 28

Pekerjaan: Dosen
 Alamat: Bantul
 E-mail:

Gunakan (✓) untuk mengisi kolom sesuai indikator yang ditentukan

NO	KEGUNAAN		1	2	3	4	5	
1	Remote ini membantu saya menjadi lebih efektif dan efisien dalam mengontrol rumah	Sangat tidak setuju				✓		Sangat setuju
2	Remote ini membantu saya menjadi lebih produktif	Sangat tidak setuju		✓				Sangat setuju
3	Remote ini berguna	Sangat tidak setuju			✓			Sangat setuju
4	Remote ini memberi saya kontrol lebih besar untuk kegiatan kontrol rumah	Sangat tidak setuju			✓			Sangat setuju
5	Remote ini membuat kontrol rumah lebih mudah untuk dilakukan	Sangat tidak setuju			✓			Sangat setuju
6	Remote ini menghemat waktu saya	Sangat tidak setuju					✓	Sangat setuju
7	Remote ini memenuhi kebutuhan saya saat mengontrol rumah	Sangat tidak setuju			✓			Sangat setuju
8	Remote ini melakukan kontrol rumah sesuai dengan yang saya harapkan	Sangat tidak setuju			✓			Sangat setuju

	KEMUDAHAN PENGGUNAAN		1	2	3	4	5	
9	Remote ini mudah digunakan	Sangat tidak setuju				✓		Sangat setuju
10	Remote ini sederhana	Sangat tidak setuju				✓		Sangat setuju
11	Remote ini user friendly	Sangat tidak setuju				✓		Sangat setuju
12	Remote ini membutuhkan langkah-langkah yang mungkin paling sedikit dalam mengontrol rumah	Sangat tidak setuju				✓		Sangat setuju
13	Remote ini fleksibel	Sangat tidak setuju				✓		Sangat setuju
14	Remote ini dapat digunakan tanpa kesulitan	Sangat tidak setuju					✓	Sangat setuju
15	Saya dapat menggunakan tanpa instruksi tertulis	Sangat tidak setuju					✓	Sangat setuju
16	Saya melihat adanya konsistensi saat saya menggunakan	Sangat tidak setuju				✓		Sangat setuju
17	Pengguna yang sesekali dan teratur memakai remote ini akan menyukainya	Sangat tidak setuju				✓		Sangat setuju



18	Jika melakukan kesalahan dalam penggunaannya, saya akan dengan cepat dan mudah memperbaikinya	Sangat tidak setuju				✓	Sangat setuju
19	saya dapat menggunakannya dengan sukses setiap kali	Sangat tidak setuju			✓		Sangat setuju

	MUDAH UNTUK DIMENGERTI		1	2	3	4	5	
20	Saya belajar dengan cepat untuk menggunakannya	Sangat tidak setuju				✓	.	Sangat setuju
21	Saya mudah mengingat bagaimana menggunakannya	Sangat tidak setuju					✓	Sangat setuju
22	Sangat mudah untuk belajar menggunakannya	Sangat tidak setuju					✓	Sangat setuju
23	Saya cepat terampil dengan alat ini	Sangat tidak setuju					✓	Sangat setuju

	KEPUASAN		1	2	3	4	5	
24	Saya puas dengan alat ini	Sangat tidak setuju				✓		Sangat setuju
25	Saya akan merekomendasikan remote ini kepada teman	Sangat tidak setuju				✓		Sangat setuju
26	Remote ini menyenangkan untuk digunakan	Sangat tidak setuju				✓		Sangat setuju
27	Remote ini bekerja dengan cara seperti yang saya inginkan	Sangat tidak setuju				✓		Sangat setuju
28	Remote ini indah/hebat	Sangat tidak setuju				✓		Sangat setuju
29	Saya merasa harus memiliki remote ini	Sangat tidak setuju				✓		Sangat setuju
30	alat ini menyenangkan untuk digunakan	Sangat tidak setuju				✓		Sangat setuju
			1	2	3	4	5	

Positif :

praktis

Negative:

Tidak tahu kondisi nyala atau hilangnya alat.

A.3 Sampel Observasi

No	Item	Perintah	Keberhasilan eksekusi task			
			Berhasil		Tidak Berhasil	
			Waktu (s)	Akses Buku Manual	Alasan	Akses Buku Manual
1	Lampu	Nyala	36	X		
2		Mati	11	X		
3	Pintu	Nyala	9	X		
4		Mati	21	X		
5	Peralatan elektronik	Nyala	19	X		
6		Mati	8	X		
7	penyiram tanaman	Nyala	6,5	X		
8		Mati	8,5	X		

nama: Heren



A.4 Dokumentasi

