

**SISTEM KENDALI BEL SEKOLAH UNTUK PROGRAM
PENGABDIAN MASYARAKAT**



KERJA PRAKTIK

Program Studi

S1 Teknik Komputer

UNIVERSITAS
Dinamika

Oleh :

ADAM FIQIH RISTANTO

17410200025

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2021

LAPORAN KERJA PRAKTIK
SISTEM KENDALI BEL SEKOLAH UNTUK PROGRAM
PENGABDIAN MASYARAKAT

Diajukan sebagai salah satu syarat untuk menempuh ujian Tahap Akhir

Program Strata Satu (S1)



Disusun Oleh:

Nama : Adam Fiqih Ristanto

NIM : 17.41020.0025

Program : S1 (Strata Satu)

Jurusan : Teknik Komputer

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2021

LEMBAR PENGESAHAN

**SISTEM KENDALI BEL SEKOLAH UNTUK PROGRAM
PENGABDIAN MASYARAKAT**

Laporan Kerja Praktik oleh

Adam Fiqih Ristanto

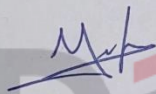
Nim : 17410200025

Telah diperiksa, diuji dan disetujui

Surabaya, 12 Januari 2021

Disetujui :

Dosen Pembimbing,



Yosefine Triwidyastuti, M.T.
NIDN. 0729038504



Penyelia,
Fakultas Teknologi dan Informatika
UNIVERSITAS



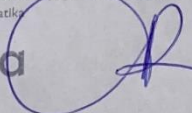
Pauladie Susanto, S.Kom., M.T.
NIDN. 0729047501

Mengetahui,

Ketua Program Studi S1 Teknik Komputer



Fakultas Teknologi dan Informatika
UNIVERSITAS
Dinamika



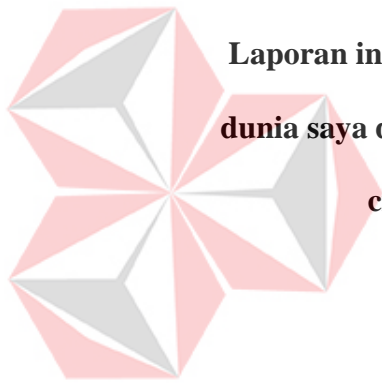
Pauladie Susanto, S.Kom., M.T.
NIDN. 0729047501



“Once You Stop Learning, You Start Dying”

ALBERT EINSTEIN

UNIVERSITAS
Dinamika



Laporan ini saya persembahkan untuk ayah dan ibu yang telah mengisi dunia saya dengan begitu banyak kebahagiaan. Terima kasih atas semua cinta yang telah ayah dan ibu berikan kepada saya.

UNIVERSITAS
Dinamika

SURAT PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya :

Nama : Adam Fiqih Ristanto

NIM : 17410200025

Program Studi : S1 Teknik Komputer

Fakultas : Fakultas Teknologi dan Informatika

Jenis Karya : Laporan Kerja Praktik

Judul Karya : **Sistem Kendali Bel Sekolah Untuk Program Pengabdian Masyarakat**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 18 Desember 2020

Yang menyatakan



Adam Fiqih Ristanto

NIM : 17.41020.0025

ABSTRAK

Universitas Dinamika mencoba menerapkan paradigma baru dalam kegiatan pengabdian kepada masyarakat yang bersifat memecahkan masalah, komprehensif, bermakna, dan tuntas. Hal-hal inilah yang menjadi alasan dikembangkannya program Program Kemitraan Masyarakat (PKM). Salah satu khalayak sasaran program PKM adalah masyarakat yang tidak produktif secara ekonomi (masyarakat umum/biasa).

Sasaran yang dituju pada PKM kali ini adalah TK Wijaya Surabaya yang masih terbilang baru didirikan dan belum memiliki sistem bel sekolah dalam kegiatan mengajar, sehingga para pengajar di TK Wijaya harus memperhatikan jam secara intensif untuk mengatur waktu pelaksanaan kegiatan di TK Wijaya. Tentu saja hal ini berdampak pada manajemen penjadwalan kegiatan yang ada di TK Wijaya mengingat para pengajar bisa saja lupa sehingga pelaksanaan kegiatan berjalan lebih lama atau lebih cepat, maka dibangun sistem kendali bel otomatis berbasis RTC DS3231 terintegrasi dengan mikrokontroler yang bekerja secara kabel, yaitu Arduino dengan inputan dan output berupa tombol *Push-On* , Potensiometer, LCD 16x2, dan Buzzer.

Dengan dibuatnya sistem ini, maka terciptalah sistem bel sekolah yang berjalan secara otomatis dalam hal pengecekan jadwal dan membunyikan bel serta pengaturan jadwal yang fleksibel, ditunjang dengan lagu yang sesuai dijadikan pengingat waktu kegiatan bagi anak-anak ditingkat TK (Taman Kanak-kanak).

Kata Kunci: RTC DS3231, Tombol *Push On*, Potensiometer, LCD 16x2, Buzzer.

KATA PENGANTAR

Puji Syukur penulis panjatkan kepada Allah SWT atas segala hikmat dan anugrah yang telah diberikan, maka Laporan Kerja Praktik ini dapat selesai dengan baik, sehingga terpenuhinya salah satu syarat dalam menempuh Tugas Akhir pada Program Studi S1 Teknik Komputer Universitas Dinamika Surabaya.

Pada kesempatan ini, penulis mengucapkan terima kasih kepada pihak-pihak yang telah memberikan bimbingan, serta kritik dan saran mendukung. Ucapan terima kasih ini penulis tujukan kepada:

1. Kedua orang tua penulis yang telah memberikan bimbingan dan dukungan moral dan materi, sehingga penulis dapat menyelesaikan Kerja Praktik beserta laporan ini dengan baik. Begitu pula bagi seluruh keluarga besar penulis yang telah memberikan kritik dan saran membangun.
2. Kelompok penulis semasa Kerja Praktik yang telah memberikan ide-ide serta motivasi dalam menyelesaikan tugas yang diberikan. Kami menyebut kelompok ini TRIO DARJO, terdiri atas penulis, Afri, dan Fikri.
3. Universitas Dinamika Surabaya, khususnya Penyelia kami Bapak Pauladie Susanto, S.Kom., M.T. yang telah memberikan kepercayaan dalam membantu pengerjaan projek Pengabdian Masyarakat.
4. Dosen Pembimbing penulis, Ibu Yosefine Triwidyastuti, M.T. yang telah memberikan masukan, saran dan kritik membangun, sehingga Kerja Praktik dan Laporan ini dapat selesai dengan baik.

5. Ketua Program Studi S1 Teknik Komputer, Bapak Pauladie Susanto, S.Kom., M.T., yang telah memberikan ijin kepada penulis untuk melaksanakan Kerja Praktik.
6. Keluarga Besar S1 Teknik Komputer yang telah memberikan semangat dan dorongan.

Surabaya, 18 Desember 2020

Penulis



UNIVERSITAS
Dinamika

DAFTAR ISI

LEMBAR PENGESAHAN	iii
SURAT PERNYATAAN.....	vi
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xvi
DAFTAR LAMPIRAN	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Manfaat	3
1.6 Kontribusi	3
BAB II GAMBARAN UMUM PERUSAHAAN.....	5
2.1 Sejarah dan Perkembangan	5
2.2 Visi, Misi, dan Strategi	8
2.2.1 Visi.....	8
2.2.2 Misi	8

2.2.3 Tujuan	8
2.2.4 Struktur Organisasi	9
2.3 Lokasi Perusahaan	11
2.4 Tri Dharma Perguruan Tinggi.....	12
2.5 PKM (Program Kemitraan Masyarakat)	14
BAB III LANDASAN TEORI.....	16
3.1 Arduino Uno	16
3.2 RTC (Reak Time Clock) DS3231	18
3.3 LCD 16x2.....	20
3.4 Buzzer	22
3.5 Tombol Push-On.....	23
3.6 Modul I2C LCD.....	24
3.7 Arduino IDE.....	25
3.7.1 Bagian – bagian pada Arduino IDE	25
3.7.2 <i>Sketch</i> Arduino IDE	28
3.8 EEPROM	28
BAB IV DESKRIPSI PEKERJAAN	30
4.1 Analisis Sistem.....	30
4.2 Diagram Alur Proses Sistem Kendali Bel Sekolah.....	31
4.3 Alur Kerja Sistem Kendali Bel Otomatis.....	32
4.4 Spesifikasi Alat	32

4.4.1 PERAPITU	33
4.5 Skematik Alat.....	33
4.6 Skematik Tampilan Menu.....	36
4.7 Penjelasan Program.....	37
4.7.1 Inisialisasi	37
4.7.2 void pinmode ().....	38
4.7.3 void setup()	39
4.7.4 void getwaktunormal ().....	39
4.7.5 void buzzerr ().....	40
4.7.6 void loop ().....	42
4.7.7 void daftarlagu ()	45
4.7.8 void updatemenu ()	47
4.7.9 void menualarm1 ().....	48
4.7.10 void digit ()	49
4.7.11 Penyimpanan data jadwal ke EEPROM	50
4.7.12 Menghapus data alarm pada EEPROM	52
4.7.13 Program lagu.....	55
4.8 Hasil Alat	58
4.8.1 Hasil Kemasan	58
4.8.2 Hasil Sistem	60
BAB V PENUTUP.....	69

5.1	Kesimpulan	69
5.2	Saran	69
DAFTAR PUSTAKA		71

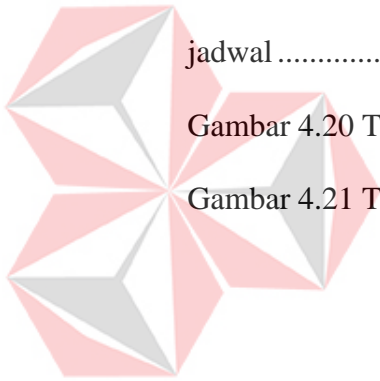


UNIVERSITAS
Dinamika

DAFTAR GAMBAR

Gambar 2.1 Struktur Organisasi.....	9
Gambar 2.2 Lokasi Universitas Dinamika	12
Gambar 3.1 Modul Arduino Uno	16
Gambar 3.2 Modul RTC DS3231	18
Gambar 3.3 LCD 16x2.....	20
Gambar 3.4 Pin-pin pada LCD 16x2.....	21
Gambar 3.5 Buzzer.....	22
Gambar 3.6 Tombol Push-On	23
Gambar 3.7 Pin-pin modul I2C	24
Gambar 3.8 Pin LCD Karakter 16 x 2.....	25
Gambar 3.9 Interface dan bagian-bagian Arduino IDE	26
Gambar 3.10 Struktur pada Sketch Arduino IDE	28
Gambar 4.1 Diagram alur Sistem Kendali Bel Sekolah.....	31
Gambar 4.2 Rangkaian listrik PERAPITU	34
Gambar 4.3 Skematik tampilan menu pada sistem PERAPITU	36
Gambar 4.4 Kemasan alat PERAPITU	59
Gambar 4.5 Tampilan awal PERAPITU	60
Gambar 4.6 Tampilan Perbaikan pada menu utama sistem	61
Gambar 4.7 Tampilan setelah masuk ke menu Perbaikan untuk set tanggal	61
Gambar 4.8 Tampilan setelah masuk ke menu Perbaikan untuk set bulan	62
Gambar 4.9 Tampilan setelah masuk ke menu Perbaikan untuk set tahun	62

Gambar 4.10 Tampilan setelah masuk ke menu Perbaikan untuk set hari.....	63
Gambar 4.11 Tampilan setelah masuk ke menu Perbaikan untuk set tanggal	63
Gambar 4.12 Tampilan setelah masuk ke menu Perbaikan untuk set menit.....	64
Gambar 4.13 Tampilan menu Alarm pada menu utama	64
Gambar 4.14 Tampilan setelah masuk ke menu Alarm	65
Gambar 4.15 Tampilan setelah masuk ke menu Tambah Jadwal untuk set jam...	65
Gambar 4.16 Tampilan setelah masuk ke menu Tambah Jadwal untuk set menit	66
Gambar 4.17 Tampilan setelah masuk ke menu Tambah Jadwal untuk set hari ..	66
Gambar 4.18 Tampilan setelah masuk ke menu Tambah Jadwal untuk set lagu..	67
Gambar 4.19 Tampilan setelah masuk ke menu utama Alarm untuk menghapus jadwal	67
Gambar 4.20 Tampilan jadwal yang telah ditambahkan oleh pengguna	68
Gambar 4.21 Tampilan jadwal yang telah dihapus oleh sistem.....	68



UNIVERSITAS
Dinamika

DAFTAR TABEL

Tabel 3.1 Cara mengakses EEPROM pada Arduino IDE.....	29
Tabel 4.1 Spesifikasi alat PERAPITU	33
Tabel 4.2 Urutan data pada sistem dan tampilan	54
Tabel 4.3 Sistem dan tampilan setelah terjadi penghapusan.....	54
Tabel 4.4 Penjelasan parameter function void PlayTone ()	56



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

Pada bab satu, menjelaskan mengenai Latar Belakang Masalah, Perumusan Masalah, Batasan Masalah, yang ada di TK Wijaya Surabaya, selanjutnya dijabarkan tujuan dan manfaat serta kontribusi.

1.1 Latar Belakang Masalah

TK Wijaya adalah TK Swasta yang bernaung dibawah yayasan PKK Kel. Penjaringan Sari. Pelaksanaan kegiatan belajar mengajar dilakukan didalam komplek balai RW kelurahan penjaringan. Kegiatan belajar mengajar di TK WIJAYA dibagi 2 sesi. Sesi pertama kegiatan belajar mengajar untuk TK A mulai pukul 07.00 - 09.00 WIB, kemudian di sesi kedua kegiatan belajar mengajar untuk TK B mulai pukul 09.15 - 11.15 WIB.

Untuk mengetahui jam pergantian kegiatan para pengajar di TK WIJAYA masih menggunakan media seperti jam tangan, smartphone, dan jam dinding. Untuk memberitahu murid bahwa kegiatan dimulai atau sudah selesai. Adapun pengajar masih harus sedikit berteriak untuk mengatur para murid ketika pergantian jam kegiatan.

Atas dasar tersebut, melalui PKM (Program Kemitraan Masyarakat) yang diselenggarakan oleh Universitas Dinamika Surabaya, maka dirancanglah sebuah sistem kendali bel sekolah yang bekerja secara otomatis sehingga para pengajar di TK WIJAYA Penjaringan Sari Surabaya terbantu dari permasalahan yang telah diuraikan di atas.

1.2 Perumusan Masalah

Bagaimana cara merancang dan membangun Sistem Bel Sekolah Otomatis berbasis RTC (*Real Time Clock*) dan Mikrokontroler Arduino dengan output alarm berupa Ringtone lagu anak – anak melalui Buzzer pada TK Wijaya Penjaringan Sari Surabaya.

1.3 Batasan Masalah

Berdasarkan permasalahan yang ada, adapun batasan masalah sesuai dengan kebutuhan di TK WIJAYA Penjaringan Sari Surabaya , yaitu:

1. Bunyi alarm melalui Buzzer.
2. Pengoperasian Sistem Bel Otomatis menggunakan tombol *Push On*.
3. Mikrokontroler menggunakan Arduino Uno untuk pengoperasian sistem Bel otomatis.
4. Penambahan atau penghapusan jadwal menggunakan EEPROM dari mikrokontroler Arduino Uno.
5. Pembuatan program Mikrokontroler menggunakan Arduino IDE.

1.4 Tujuan

Adapun berdasarkan rumusan masalah, tujuannya adalah membuat Sistem Bel Sekolah Otomatis berbasis RTC (*Real Time Clock*) dan Mikrokontroler Arduino dengan output alarm berupa Ringtone lagu anak – anak melalui Buzzer untuk menunjang sistem penjadwalan kegiatan di TK Wijaya Penjaringan Sari Surabaya.

1.5 Manfaat

1. Bagi Mahasiswa

- a. Menambah relasi dengan pengajar di TK Wijaya.
- b. Membekali diri dengan wawasan dan pengetahuan dalam dunia kerja, baik itu *hardskill* dan *softskill*.
- c. Dapat menerapkan ilmu pengetahuan yang didapatkan selama perkuliahan.

2. Bagi TK Wijaya

- a. Dapat mempererat hubungan antara TK Wijaya dengan Universitas Dinamika Surabaya.
- b. Permasalahan mengenai penjadwalan di TK Wijaya dapat terbantu, karena mendapatkan bantuan berupa alat Bel Otomatis dari mahasiswa.
- c. Dengan adanya mahasiswa Kerja Praktik, maka sistem jam pelajaran di TK Wijaya menjadi terstruktur tanpa harus melihat waktu secara manual.

3. Bagi Universitas Dinamika

- a. Sistem kendali bel sekolah sebagai produk program pengabdian diharapkan dapat menjadi sarana transfer ilmu dari program studi Teknik Komputer ke TK Wijaya.
- b. Pkm diharapkan bisa menjadi program yang berkelanjutan sebagai bentuk kerjasama antara Universitas Dinamika dengan TK Wijaya.

1.6 Kontribusi

Memberikan kontribusi ke TK Wijaya Penjaringan Sari Surabaya dengan merancang dan membangun Sistem Bel Sekolah Otomatis berbasis RTC (*Real Time*

Clock) dan Mikrokontroler Arduino dengan output alarm berupa Ringtone lagu anak – anak melalui Buzzer, sehingga menunjang sistem penjadwalan kegiatan di TK Wijaya Penjaringan Sari Surabaya.



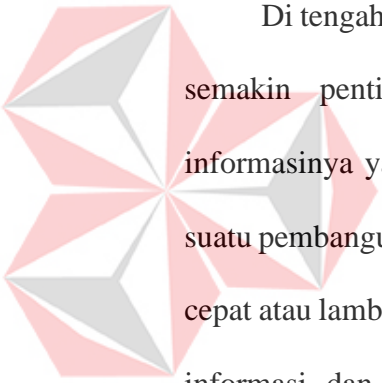
UNIVERSITAS
Dinamika

BAB II

GAMBARAN UMUM PERUSAHAAN

Pada bab dua, menjelaskan mengenai gambaran umum dari Universitas Dinamika Surabaya, yang terdiri atas sejarah dan perkembangan, visi dan misi, serta strategi, tugas pokok fungsi dan susunan organisasi, sumber daya, moto dan logo, serta lokasi perusahaan.

2.1 Sejarah dan Perkembangan



Di tengah langkah-langkah Pembangunan Nasional, posisi informasi menjadi semakin penting. Hasil perkembangan sangat ditentukan oleh substansi informasinya yang dimiliki oleh suatu negara. Kemajuan yang didambakan oleh suatu pembangunan akan mudah dicapai dengan kelengkapan informasi. Kecepatan cepat atau lambat suatu perkembangan juga ditentukan oleh kecepatan memperoleh informasi dan kecepatan untuk menginformasikannya kembali kepada pihak berwenang.

Kemajuan teknologi telah memberikan jawaban terhadap kebutuhan informasi, komputer yang canggih memungkinkan untuk memperoleh informasi dengan cepat, tepat dan akurat. Hasil dari informasi canggih telah mulai menyentuh kehidupan kita. Penggunaan dan pemanfaatan komputer yang optimal dapat memacu laju perkembangan. Kesadaran akan hal itu membutuhkan pengadaan tenaga ahli yang terampil dalam mengelola informasi, dan pendidikan adalah salah satu cara yang harus ditempuh untuk memenuhi kebutuhan tenaga kerja. Dalam hal

ini pendidikan adalah salah satu cara yang harus ditempuh untuk memenuhi kebutuhan tenaga kerja.

Berdasarkan pemikiran ini, maka untuk pertama kalinya di wilayah Jawa Timur, Yayasan Putra Bhakti membuka Komputer Pendidikan Tinggi, "Akademi Komputer & Informatika Surabaya" (Akis) (Akademi Komputer. & Teknologi Informasi Surabaya) pada 30 April 1983 dengan dekrit Yayasan Putra Bhakti nomor 01 / KPT / PB / III / 1983. Pendirinya adalah:

1. Laksda. TNI (Purn) Mardiono
2. Ir. Andrian A. T
3. Ir. Handoko Anindyo
4. Dra. Suzana Surojo
5. Dra. Rosy Merianti, Ak

Kemudian berdasarkan rapat BKLPTS tanggal 2-3 Maret 1984 kepanjangan AKIS dirubah menjadi Akademi Manajemen Informatika & Komputer Surabaya yang bertempat di jalan Ketintang Baru XIV/2. Tanggal 10 Maret 1984 memperoleh Ijin Operasional penyelenggaraan program Diploma III Manajemen Informatika dengan surat keputusan nomor: 061/Q/1984 dari Direktorat Jendral Pendidikan Tinggi (Dikti) melalui Koordinator Kopertis Wilayah VII. Kemudian pada tanggal 19 Juni 1984 AKIS memperoleh status TERDAFTAR berdasar surat keputusan Direktorat Jendral Pendidikan Tinggi (Dikti) nomor: 0274/O/1984 dan kepanjangan AKIS berubah lagi menjadi Akademi Manajemen Informatika & Teknik Komputer Surabaya. Berdasar SK Dirjen DIKTI nomor: 45/DIKTI/KEP/1992, status DIII Manajemen Informatika dapat ditingkatkan menjadi DIAKUI.

Waktu berlalu terus, kebutuhan akan informasi juga terus meningkat. Untuk menjawab kebutuhan tersebut AKIS ditingkatkan menjadi Sekolah Tinggi dengan membuka program studi Strata 1 dan Diploma III jurusan Manajemen Informatika. Dan pada tanggal **20 Maret 1986 nama AKIS berubah menjadi STIKOM SURABAYA**, singkatan dari Sekolah Tinggi Manajemen Informatika & Teknik

Komputer Surabaya berdasarkan SK Yayasan Putra Bhakti nomor: 07/KPT/PB/03/86 yang selanjutnya memperoleh STATUS TERDAFTAR pada tanggal 25 Nopember 1986 berdasarkan Keputusan Mendikbud nomor: 0824/O/1986 dengan menyelenggarakan pendidikan S1 dan D III Manajemen Informatika. Di samping itu STIKOM SURABAYA juga melakukan pembangunan gedung Kampus baru di jalan Kutisari 66 yang saat ini menjadi Kampus II STIKOM SURABAYA. Peresmian gedung tersebut dilakukan pada tanggal 11 Desember 1987 oleh Bapak Wahono Gubernur Jawa Timur pada saat itu.

Berdasarkan Keputusan Menteri Pendidikan dan Kebudayaan No 378/E/O/2014 tanggal 4 September 2014 maka STIKOM Surabaya resmi berubah bentuk menjadi Institut dengan nama Institut Bisnis dan Informatika Stikom Surabaya. Pada tanggal 29 Juli 2019, melalui surat keputusan Riset Dikti, Institut bisnis dan informatika STIKOM Surabaya resmi berubah bentuk menjadi **UNIVERSITAS DINAMIKA.**

Program studi yang diselenggarakan oleh UNIVERSITAS DINAMIKA adalah sebagai berikut:

A. Fakultas Ekonomi dan Bisnis:

1. Program Studi S1 Akuntansi
2. Program Studi S1 Manajemen
3. Program Studi DIII Administrasi Perkantoran

B. Fakultas Teknologi dan Informatika:

1. Program Studi S1 Sistem Informasi
2. Program Studi S1 Teknik Komputer
3. Program Studi S1 Desain dan Komunikasi Visual
4. Program Studi S1 Desain Produk
5. Program Studi DIV Produksi Film dan Televisi
6. Program Studi DIII Sistem Informasi

2.2 Visi, Misi, dan Strategi

2.2.1 Visi

Visi Universitas Dinamika Surabaya merupakan potret masa depan yang dicita-citakan, yaitu: “Menjadi Perguruan Tinggi yang Produktif dalam berinovasi.”

2.2.2 Misi

Universitas Dinamika Surabaya dalam usaha keras mencapai Visi yang telah ditetapkan diatas, mengemban Misi sebagai berikut:

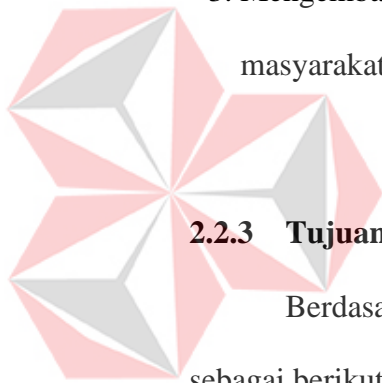
1. Menyelenggarakan Pendidikan yang berkualitas dan futuristis
2. Mengembangkan produktivitas berkreasi dan berinovasi
3. Mengembangkan layanan untuk meningkatkan kesejahteraan

masyarakat

2.2.3 Tujuan

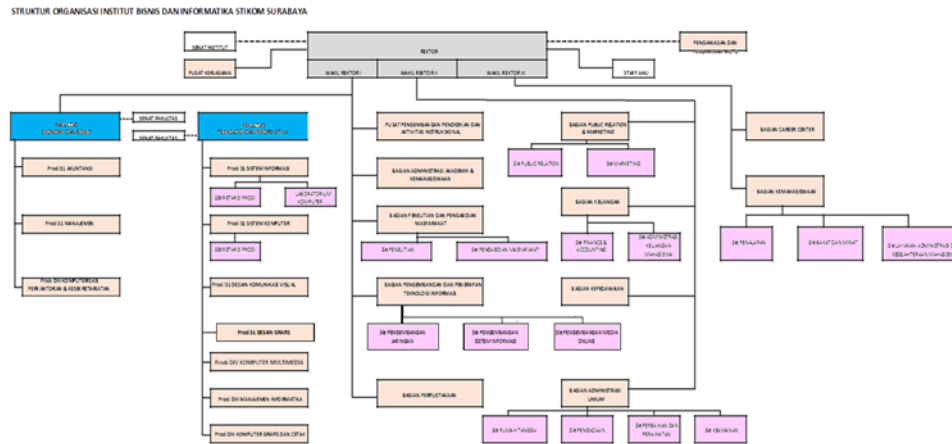
Berdasarkan visi dan misinya, univestras dinamika menetapkan tujuan sebagai berikut :

1. Menghasilkan SDM berbudipekerti luhur, kompetitif, dan adaptif terhadap perkembangan
2. Mengembangkan Pendidikan yang berkualitas dan inovatif
3. Menghasilkan produk kreatif dan inovatif yang tepat guna
4. Memperluas kolaborasi yang produktif
5. Mengembangkan lingkungan yang sehat dan produktif
6. Meningkatkan produktivitas layanan bagi masyarakat



UNIVERSITAS
Dinamika

2.2.4 Struktur Organisasi



Gambar 2.1 Struktur Organisasi

Universitas Dinamika, terdiri atas:

A. Rektor

B. Rektor, membawahi:

a. Wakil Rektor I

1. Fakultas Ekonomi Dan Bisnis

1.1 Senat Fakultas

1.2 Program Studi S1 Akutansi

1.3 Program Studi S1 Manajemen

1.4 Program Studi DIII Komputerisasi dan Kesekretariatan

2. Fakultas Teknologi dan Informatika

2.1 Senat Fakultas

2.2 Program Studi S1 Sistem Informasi

A. Sekretaris Program Studi

B. Laboratorium Komputer

2.3 Program Studi S1 Sistem Informasi

A. Sekretaris Program Studi

2.4 Program Studi S1 Desain Komunikasi Visual

2.5 Program Studi S1 Desain Grafis

2.6 Program Studi DIV Komputer Multimedia

2.7 Program Studi DIII Manajemen Informatika

2.8 Program Studi DIII Komputer Grafis dan Cetak

2.9 Pusat Pengembangan Pendidikan dan Aktivitas Instruksional

2.10 Bagian Administrasi dan Kemahasiswaan

2.11 Bagian Penelitian dan Pengabdian Masyarakat

A. Sie Penelitian

B. Sie Pengabdian Masyarakat

2.12 Bagian Pengembangan dan Penerapan Teknologi Informasi

A. Sie Pengembangan Jaringan

B. Sie Pengembangan Sistem informasi

C. Sie Pengembangan Media Online

2.13 Bagian Perpustakaan

b. Wakil Rektor II

1. Bagian Public Relation dan Marketing

A. Sie Public Relation

B. Sie Marketing

C. Bagian Keuangan

1.1 Sie Finance dan Accounting

1.2 Sie Administrasi Keuangan Mahasiswa



UNIVERSITAS
Dinamika

A. Bagian Kepegawaian

B. Bagian Administrasi Umum

1.3 Sie Rumah tangga

1.4 Sie Pengadaan

1.5 Sie Perbaikan dan Perawatan

1.6 Sie Keamanan

c. Wakil Rektor III

1. Bagian Career Center

2. Bagian Kemahasiswaan

A. Sie Penalaran

B. Sie Bakat dan Minat

C. Sie Layanan Administrasi dan Kesejahteraan

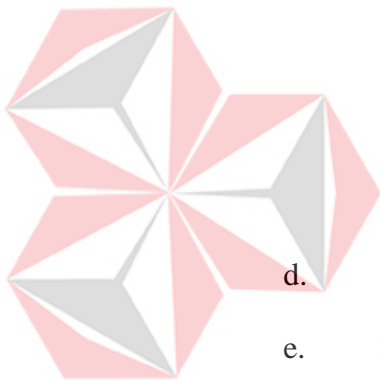
Mahasiswa

d. Senat Institut

e. Pusat Kerja Sama

f. Staff Ahli

g. Pengawasan dan Penjaminan Mutu



UNIVERSITAS
Dinamika

2.3 Lokasi Perusahaan

Lokasi Universitas Dinamika yaitu Raya Kedung Baruk No.98, Kedung Baruk, Kec. Rungkut, Kota SBY, Jawa Timur 60298. Berikut adalah peta dari lokasi Universitas Dinamika:



Gambar 2.2 Lokasi Universitas Dinamika

2.4 Tri Dharma Perguruan Tinggi

Tri Dharma Perguruan Tinggi adalah salah satu visi dari seluruh perguruan tinggi yang ada di Indonesia. Tri Dharma Perguruan Tinggi merupakan salah satu tujuan pencapaian yang harus dilakukan oleh perguruan tinggi tersebut.

Tri Dharma Perguruan Tinggi terdiri dari 3 poin, yaitu :

1. Pendidikan dan Pengajaran
2. Penelitian dan Pengembangan
3. Pengabdian kepada Masyarakat

Maka dari itu seluruh mahasiswa wajib tahu dan memahami apa yang dimaksud dengan Tri Dharma Perguruan Tinggi. Berikut ini isi dari Tri Dharma Perguruan Tinggi:

1. Pendidikan dan Pengajaran. Pendidikan pada hakikatnya merupakan ajang memberikan ilmu pengetahuan. Pendidikan dan Pengajaran disini diartikan sebagai sistem pendidikan yang berkelanjutan atau lebih dikenal dengan istilah *transfer of knowledge* , jadi mahasiswa akan menjalani pendidikan dan kemudian dikembangkan dengan penelitian. Kualitas penelitian dan pengembangan oleh mahasiswa saat ini telah ditunjang oleh mutu pendidikan yang bagus. Yaitu pendidikan mulai dari Diploma, Strata, hingga Doktor. Diharapkan akan adanya kualitas yang sudah memenuhi standar tersebut, maka Indonesia diharapkan bisa memenuhi tenaga di bidang penelitian untuk memulai memajukan kualitas negara dengan lebih mandiri.

2. Penelitian dan Pengembangan. Kegiatan penelitian dan pengembangan memiliki peranan yang sangat penting dalam mendukung kemajuan ilmu pengetahuan dan teknologi. Tanpa adanya penelitian dan pengembangan, maka laju perkembangan ilmu pengetahuan akan terhambat. Penelitian tidaklah selalu berdiri sendiri, karena dilatar belakangi oleh kebutuhan dalam proses pembangunan dalam arti yang luas. Penelitian juga sebagai faktor utama dalam menentukan keputusan terkait suatu masalah. Penelitian yang dilakukan ada dua jenis, yaitu penelitian terapan dan penelitian terhadap ilmu-ilmu dasar. Penelitian terapan digunakan untuk mengatasi masalah yang sedang terjadi

pada saat itu, sementara penelitian terhadap ilmu-ilmu dasar manfaatnya akan lebih penting di masa depan.

3. **Pengabdian pada Masyarakat.** Pendidikan yang kemudian dilanjutkan dengan penelitian sebenarnya merupakan persiapan untuk menghadapi kasus secara nyata. Penelitian juga merupakan kontribusi dari perguruan tinggi terutama mahasiswanya terhadap masyarakat. Adanya penelitian yang dilakukan secara individu maupun kelompok dari mahasiswa perguruan tinggi secara langsung merupakan contoh dari “pengabdian pada masyarakat”. Diharapkan masyarakat juga memberikan umpan balik yang positif terhadap penelitian-penelitian yang dilakukan untuk mengetahui dan mempelajari lebih lanjut terkait objek dan masalah yang dihadapi. Sehingga para mahasiswa dapat lebih memahami kebutuhan masyarakat sendiri.

2.5 PKM (Program Kemitraan Masyarakat)

Direktorat Riset dan Pengabdian Masyarakat (DRPM) Ditjen Penguatan Risbang Kemenristekdikti mencoba menerapkan paradigma baru dalam kegiatan pengabdian kepada masyarakat yang bersifat memecahkan masalah, komprehensif, bermakna, tuntas, dan berkelanjutan (*sustainable*) dengan sasaran yang tidak tunggal. Hal-hal inilah yang menjadi alasan dikembangkannya program Program Kemitraan Masyarakat (PKM).

Khalayak sasaran program PKM adalah: 1) masyarakat yang produktif secara ekonomi; 2) masyarakat yang belum produktif secara ekonomis, tetapi berhasrat kuat menjadi wirausahawan; dan 3) masyarakat yang tidak produktif secara ekonomi (masyarakat umum/biasa).

Khalayak sasaran (mitra) masyarakat yang produktif secara ekonomi seperti: kelompok perajin, kelompok nelayan, kelompok tani, kelompok ternak, yang setiap anggotanya memiliki karakter produktif secara ekonomis. Mitra sasaran industri rumah tangga (IRT) dengan kepemilikan usaha bersifat individu/perseorangan disyaratkan mempunyai karyawan minimal 4 orang di luar anggota keluarga. Mitra sasaran yang mengarah pada bidang ekonomi produktif disyaratkan merupakan kelompok dengan jumlah anggota minimal 5 orang, seperti kelompok dasawisma, pokdarwis, kelompok PKK, kelompok pengajian, kelompok ibu-ibu rumah tangga dan lain-lain.

Mitra sasaran masyarakat yang tidak produktif secara ekonomi misalnya sekolah (PAUD, SD, SMP, SMA/SMK), karang taruna, kelompok ibu-ibu rumah tangga, kelompok anak-anak jalanan, RT/RW, dusun, desa, Puskesmas/Posyandu, Pesantren dan lain sebagainya.

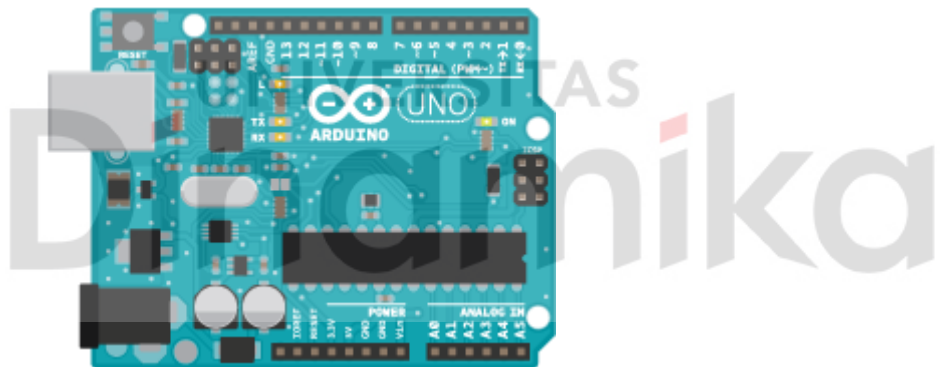
Jenis permasalahan yang wajib ditangani dalam program PKM, khususnya masyarakat produktif secara ekonomi atau calon wirausaha baru meliputi bidang produksi, manajemen usaha dan pemasaran. Untuk kegiatan yang tidak bermuara pada bidang ekonomi, wajib mengungkapkan rinci permasalahan yang diprioritaskan untuk diselesaikan seperti peningkatan pelayanan, peningkatan ketentraman masyarakat, memperbaiki/membantu fasilitas layanan dan lain-lain. Kegiatan yang dilaksanakan pada mitra PKM baik mitra produktif secara ekonomi, mengarah ke produktif ekonomi dan mitra tidak produktif/sosial harus terdiri dari 2 bidang kegiatan yang membutuhkan kepakaran yang berbeda.

BAB III

LANDASAN TEORI

Pada bab tiga, menjelaskan mengenai landasan teori dari Kerja Praktik di Universitas Dinamika Surabaya, yang memuat perangkat keras maupun perangkat lunak yang digunakan untuk memenuhi pekerjaan selama Kerja Praktik berlangsung, meliputi Arduino Uno, Modul RTC DS3231, LCD 16x2, Buzzer, Tombol *Push-On*, Modul I2C LCD dan Arduino IDE.

3.1 Arduino Uno



Gambar 3.1 Modul Arduino Uno

Arduino Uno adalah board mikrokontroler berbasis ATmega328 (datasheet). Memiliki 14 pin input dari output digital dimana 6 pin input tersebut dapat digunakan sebagai output PWM dan 6 pin input analog, 16 MHz osilator kristal, koneksi USB, jack power, ICSP header, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan Board Arduino Uno ke komputer dengan menggunakan kabel USB atau listrik dengan AC yang-ke adaptor-DC atau baterai untuk menjalankannya.

Uno berbeda dengan semua board sebelumnya dalam hal koneksi USB-to-serial yaitu menggunakan fitur Atmega8U2 yang diprogram sebagai konverter USB-to-serial berbeda dengan board sebelumnya yang menggunakan chip FTDI driver USB-to-serial.

Nama “Uno” berarti satu dalam bahasa Italia, untuk menandai peluncuran Arduino 1.0. Uno dan versi 1.0 akan menjadi versi referensi dari Arduino. Uno adalah yang terbaru dalam serangkaian board USB Arduino, dan sebagai model referensi untuk platform Arduino, untuk perbandingan dengan versi sebelumnya, lihat indeks board Arduino.

Spesifikasi dari Arduino Uno secara singkat :

- Mikrokontroler : ATmega32P
- Tegangan operasional pada 5 Vdc
- Tegangan masukan (rekomendasi) pada 7 – 12 Vdc
- Jumlah Digital I/O > 14 pin
- Jumlah analog Input > 6 pin
- Flash Memory 32 KB
- SRAM 2 KB
- eeprom 1 KB
- Clocking speed > 16 MHz
- Panjang papan elektronik > 68.6 mm
- Lebar papan elektronik > 53.4 mm
- Berat modul : 25 gr

Untuk saat ini Arduino Uno yang dijual dipasaran yaitu Arduino R3 (Revisi 3).

Dari Arduino revisi terbaru itu memiliki fitur-fitur baru antara lain :

- **Pinout** : ditambah dengan pin khusus SDA dan SCL yang dekat dengan pin AREF dan 2 pin baru lainnya yang diletakkan dekat dengan pin RESET, IOREF. Untuk ke depannya, module shield kompatibel dengan board yang beroperasi dengan tegangan 5V.
- Rangkaian **RESET** yang lebih efektif
- Penggunaan **Atmega 16U2** menggantikan **Atmega 8U2**

3.2 RTC (Real Time Clock) DS3231



Gambar 3.2 Modul RTC DS3231

Module RTC DS3231 adalah salah satu jenis module yang dimana berfungsi sebagai RTC (Real Time Clock) atau pewaktuan digital serta penambahan fitur pengukur suhu yang dikemas kedalam 1 module.

Pada RTC DS3231 terdapat IC EEPROM tipe AT24C32 yang dapat dimanfaatkan juga. Interface atau antarmuka untuk mengakses modul ini yaitu menggunakan i2c atau two wire (SDA dan SCL). Sehingga apabila diakses menggunakan mikrontroler misal Arduino Uno pin yang dibutuhkan 2 pin saja dan 2 pin power.

Module DS3231 RTC ini pada umumnya sudah tersedia dengan battery CR2032 3V yang berfungsi sebagai back up RTC apabila catudaya utama mati. RTC DS3231 memiliki banyak kelebihan, sebagai contoh untuk range VCC input dapat disupply menggunakan tegangan antara 2.3V sampai 5.5V dan memiliki cadangan baterai.

Berbeda dengan DS1307, pada DS3231 juga memiliki kristal terintegrasi (sehingga tidak diperlukan kristal eksternal), sensor suhu, 2 alarm waktu terprogram, pin output 32.768 kHz untuk memastikan akurasi yang lebih tinggi. Selain itu, terdapat juga EEPROM AT24C32 yang bisa memberi Anda 32K EEPROM untuk menyimpan data, ini adalah pilihan terbaik untuk aplikasi yang memerlukan untuk fitur data logging, dengan presisi waktu yang lebih tinggi.

Spesifikasi dan fitur :

- RTC yang Sangat Akurat Mengelola Semua Fungsi Pengatur Waktu
- Jam Real Time Menghitung Detik, Menit, Jam, Tanggal Bulan, Bulan, Hari dalam Seminggu, dan tahun, dengan Kompensasi Tahun Lawan Berlaku Hingga 2100
- Akurasi $\pm 2\text{ppm}$ dari 0°C sampai $+40^\circ\text{C}$
- Akurasi $\pm 3.5\text{ppm}$ dari -40°C sampai $+85^\circ\text{C}$
- Digital Temp Sensor Output: $\pm 3^\circ\text{C}$ Akurasi
- Mendaftar untuk Aging Trim
- Active-Low RST Output / Pushbutton Reset Debounce Input
- Two Time-of-Day Alarms
- Output Programmable Square-Wave Output
- Antarmuka Serial Sederhana Menghubungkan ke Kebanyakan Microcontrollers
- Kecepatan data transfer I2C Interface (400kHz)
- Masukan Cadangan Baterai untuk Pencatatan Waktu Terus-menerus

- Low Power Operation Memperpanjang Waktu Jalankan Baterai-Cadangan
- Rentang Suhu Operasional: Komersial (0° C sampai + 70° C) dan Industri (-40° C sampai +85° C)
- Tegangan operasi: 3,3-5,55 V
- Chip jam: chip clock presisi tinggi DS3231
- Ketepatan Jam: Kisaran 0-40, akurasi 2ppm, kesalahannya sekitar 1 menit
- Output gelombang persegi yang dapat diprogram
- Sensor suhu chip hadir dengan akurasi 3
- Chip memori: AT24C32 (kapasitas penyimpanan 32K)
- Antarmuka bus IIC, kecepatan transmisi maksimal 400KHz (tegangan kerja 5V)
- Dapat mengalir dengan perangkat IIC lainnya, alamat 24C32 dapat disingkat A0 / A1 / A2
- memodifikasi alamat defaultnya adalah 0x57
- Dengan baterai isi ulang CR2032, untuk memastikan sistem setelah power
- Ukuran: 38mm (panjang) * 22mm (W) * 14mm (tinggi)
- Berat: 8g

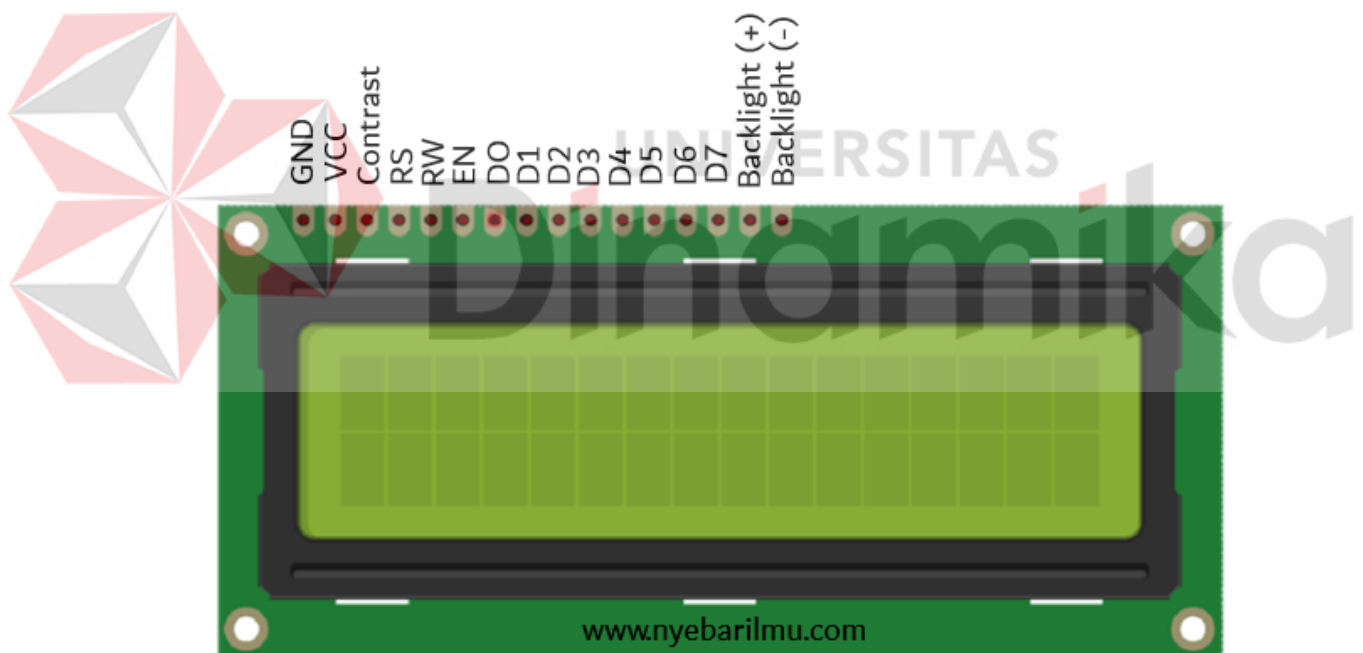
3.3 LCD 16x2



Gambar 3.3 LCD 16x2

LCD (*Liquid Crystal Display*) 16x2 adalah jenis media tampilan atau Display dari bahan cairan kristal sebagai penampil utama. LCD 16x2 dapat menampilkan sebanyak 32 karakter yang terdiri dari 2 baris dengan tiap baris menampilkan 16 karakter. Pada Arduino IDE untuk memprogram LCD 16x2 dibutuhkan library bernama `LiquidCrystal.h`. Adapun fitur – fitur yang tersedia antara lain :

- Terdiri dari 16 kolom dan 2 baris
- Dilengkapi dengan back light
- Mempunyai 192 karakter tersimpan
- Dapat dialamati dengan mode 4-bit dan 8-bit
- Terdapat karakter generator terprogram

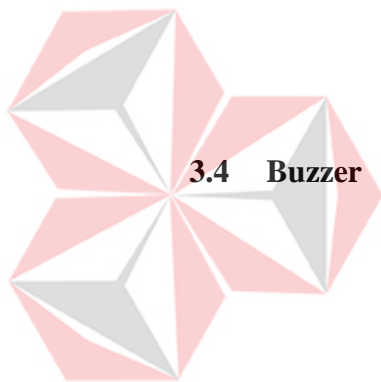


Gambar 3.4 Pin-pin pada LCD 16x2

Keterangan :

1. **GND** : catu daya 0Vdc
2. **VCC** : catu daya positif
3. **Constrate** : untuk kontras tulisan pada LCD
4. **RS** atau **Register Select** :

- High : untuk mengirim data
 - Low : untuk mengirim instruksi
5. **R/W** atau **Read/Write**
- High : mengirim data
 - Low : mengirim instruksi
 - Disambungkan dengan LOW untuk pengiriman data ke layar
6. **E (enable)** : untuk mengontrol ke LCD ketika bernilai LOW, LCD tidak dapat diakses
7. **D0 – D7** = Data Bus 0 – 7
8. **Backlight +** : disambungkan ke VCC untuk menyalakan lampu latar
9. **Backlight –** : disambungkan ke GND untuk menyalakan lampu latar



UNIVERSITAS
Dinamika



Gambar 3.5 Buzzer

Buzzer Listrik adalah sebuah komponen elektronika yang dapat mengubah sinyal listrik menjadi getaran suara. Pada umumnya, Buzzer yang merupakan sebuah perangkat audio ini sering digunakan pada rangkaian anti-maling, Alarm pada Jam Tangan, Bel Rumah, peringatan mundur pada Truk dan perangkat peringatan bahaya lainnya.

Jenis Buzzer yang sering ditemukan dan digunakan adalah Buzzer yang berjenis Piezoelectric, hal ini dikarenakan Buzzer Piezoelectric memiliki berbagai kelebihan seperti lebih murah, relatif lebih ringan dan lebih mudah dalam menggabungkannya ke Rangkaian Elektronika lainnya. Buzzer yang termasuk dalam keluarga Transduser ini juga sering disebut dengan Beeper.

Jika dibandingkan dengan Speaker, Piezo Buzzer relatif lebih mudah untuk digerakan. Sebagai contoh, Piezo Buzzer dapat digerakan hanya dengan menggunakan output langsung dari sebuah IC TTL, hal ini sangat berbeda dengan Speaker yang harus menggunakan penguat khusus untuk menggerakan Speaker agar mendapatkan intensitas suara yang dapat didengar oleh manusia.

Piezo Buzzer dapat bekerja dengan baik dalam menghasilkan frekuensi di kisaran 1 – 5 kHz hingga 100 kHz untuk aplikasi Ultrasound. Tegangan Operasional Piezoelectric Buzzer yang umum biasanya berkisar diantara 3Volt hingga 12 Volt.

3.5 Tombol Push-On



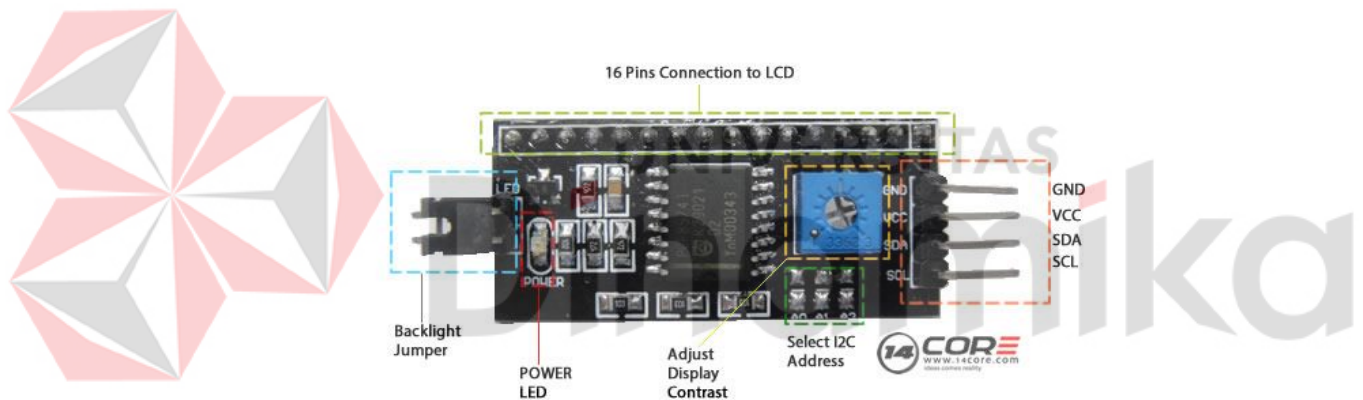
Gambar 3.6 Tombol Push-On

Push button switch (saklar tombol tekan) adalah perangkat / saklar sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan

sistem kerja tekan unlock (tidak mengunci). Sistem kerja unlock disini berarti saklar akan bekerja sebagai device penghubung atau pemutus aliran arus listrik saat tombol ditekan, dan saat tombol tidak ditekan (dilepas), maka saklar akan kembali pada kondisi normal.

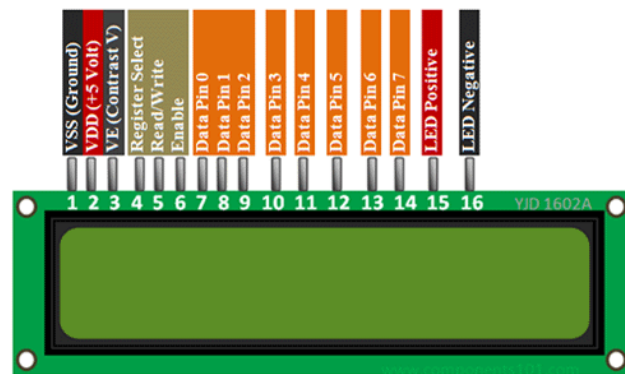
Sebagai device penghubung atau pemutus, push button switch hanya memiliki 2 kondisi, yaitu On dan Off (1 dan 0). Istilah On dan Off ini menjadi sangat penting karena semua perangkat listrik yang memerlukan sumber energi listrik pasti membutuhkan kondisi On dan Off.

3.6 Modul I2C LCD



Gambar 3.7 Pin-pin modul I2C

Pada umumnya LCD Karakter dikendalikan dengan menggunakan metode paralel, metode ini banyak menggunakan pin dari mikrokontroler, enam sampai tujuh pin yang digunakan, maka dari itu untuk meminimalisir penggunaan pin digunakan modul I2C untuk mengendalikan LCD karakter 16 x 2, hanya menggunakan dua pin, yaitu SDA dan SCL.



Gambar 3.8 Pin LCD Karakter 16 x 2

3.7 Arduino IDE

Arduino IDE (Integrated Development Environment) adalah software yang di gunakan untuk memprogram di arduino, dengan kata lain Arduino IDE sebagai media untuk memprogram board Arduino. Arduino IDE bisa di download secara gratis di website resmi Arduino IDE.

Arduino IDE ini berguna sebagai text editor untuk membuat, mengedit, dan juga mevalidasi kode program. bisa juga digunakan untuk meng-upload ke board Arduino. Kode program yang digunakan pada Arduino disebut dengan istilah Arduino “sketch” atau disebut juga source code arduino, dengan ekstensi file source code .ino

3.7.1 Bagian – bagian pada Arduino IDE

Editor Programming pada umumnya memiliki fitur untuk cut / paste dan untuk find / replace teks, demikian juga pada Arduino IDE. Pada bagian keterangan aplikasi memberikan pesan balik saat menyimpan dan mengekspor serta sebagai tempat menampilkan kesalahan. Konsol log menampilkan teks log dari aktifitas

Arduino IDE, termasuk pesan kesalahan yang lengkap dan informasi lainnya. Pojok kanan bawah menampilkan port serial yang di gunakan. Tombol toolbar terdapat ikon tombol pintas untuk memverifikasi dan meng-upload program, membuat, membuka, dan menyimpan sketch, dan membuka monitor serial.



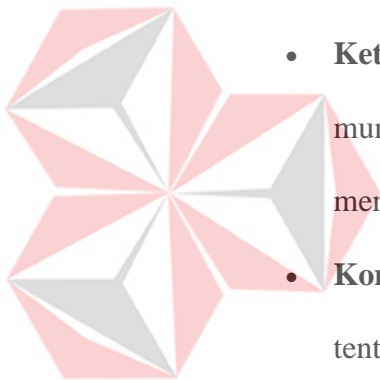
Gambar 3.9 Interface dan bagian-bagian Arduino IDE

Keterangan :

- **Verify** pada versi sebelumnya dikenal dengan istilah Compile. Sebelum aplikasi di-upload ke board Arduino, biasanya untuk memverifikasi terlebih dahulu sketch yang dibuat. Jika ada kesalahan pada sketch, nanti akan muncul error. Proses Verify / Compile mengubah sketch ke binary code untuk di-upload ke mikrokontroler.
- **Upload** tombol ini berfungsi untuk mengupload sketch ke board Arduino. Walaupun kita tidak mengklik tombol verify, maka sketch

akan di-compile, kemudian langsung diupload ke board. Berbeda dengan tombol verify yang hanya berfungsi untuk memverifikasi source code saja.

- **New Sketch** Membuka window dan membuat sketch baru.
- **Open Sketch** Membuka sketch yang sudah pernah dibuat. Sketch yang dibuat dengan IDE Arduino akan disimpan dengan ekstensi file .ino
- **Save Sketch** menyimpan sketch, tapi tidak disertai dengan mengcompile.
- **Serial Monitor** Membuka interface untuk komunikasi serial, nanti akan kita diskusikan lebih lanjut pada bagian selanjutnya.
- **Keterangan Aplikasi** pesan-pesan yang dilakukan aplikasi akan muncul di sini, misal *Compiling* dan *Done Uploading* ketika kita mengcompile dan mengupload sketch ke board Arduino
- **Konsol log** Pesan-pesan yang dikerjakan aplikasi dan pesan-pesan tentang sketch akan muncul pada bagian ini. Misal, ketika aplikasi mengcompile atau ketika ada kesalahan pada sketch yang kita buat, maka informasi error dan baris akan diinformasikan di bagian ini.
- **Baris Sketch** bagian ini akan menunjukkan posisi baris kursor yang sedang aktif pada sketch.
- **Informasi Board dan Port** Bagian ini menginformasikan port yang dipakai oleh board Arduino.



3.7.2 Sketch Arduino IDE

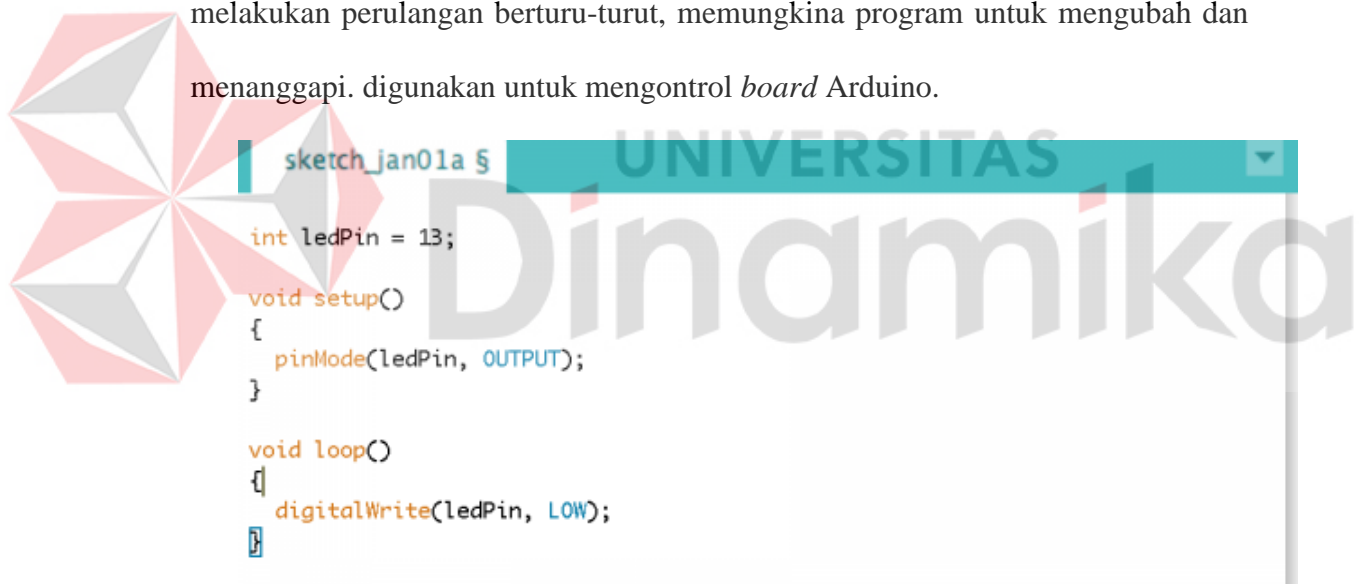
Struktur kode pada arduino yaitu berisi fungsi `setup()` dan `loop()`.

- **Setup()**

fungsi ini dipanggil pertama kali ketika menjalankan sketch. digunakan sebagai tempat inisialisasi *variable*, *pin mode*, penggunaan *library* dan lainnya. fungsi ini dijalankan sekali ketika *board* dinyalakan atau di reset.

- **loop()**

Setelah membuat fungsi `setup()` sebagai tempat inisialisasi variabel dan menetapkan nilai maka selanjutnya fungsi `loop()` seperti namanya fungsi ini akan melakukan perulangan berturut-turut, memungkinkan program untuk mengubah dan menanggapi. digunakan untuk mengontrol *board* Arduino.



Gambar 3.10 Struktur pada Sketch Arduino IDE

3.8 EEPROM

EEPROM(Electrically Erasable Programmable Read-Only Memory) adalah sebuah memory yang tidak terhapus walau pun tanpa di aliri listrik. EEPROM ini memiliki alamat dan jumlah memory. Data yang disimpan tidak akan terhapus

walaupun tanpa di aliri listrik lagi. Berbeda dengan RAM, tanpa di aliri listrik, atau Arduino padam, maka memory kembali menjadi kosong.

Setiap data yang ingin disimpan pada blok *memory* memiliki alamat, Biasanya Arduino pada satu alamat hanya mampu menyimpan memory sebanyak 8 bit, jika dalam desimal hanya dapat menyimpan data sebesar 255 dalam satu alamat. Cara mengakses EEPROM pada Arduino IDE dapat dilihat pada Tabel 3.1 berikut.

Tabel 3.1 Cara mengakses EEPROM pada Arduino IDE

Program	Keterangan
<code>#include <EEPROM.h></code>	Library untuk EEPROM untuk Arduino IDE
<code>EEPROM.write(address,data);</code>	Menulis data kedalam EEPROM. Parameter “address” untuk alamat EEPROM, “data” untuk tempat data yang disimpan kedalam EEPROM.
<code>EEPROM.read(address);</code>	Membaca data yang telah tersimpan didalam EEPROM. Parameter “address” untuk memanggil data yang tersimpan pada alamat tertentu.

BAB IV

DESKRIPSI PEKERJAAN

Pada bab empat, menjelaskan mengenai deskripsi pekerjaan yang meliputi Analisis Sistem, Spesifikasi Alat, Skematik Alat, Desain Alat, Pembuatan Program, Uji Coba Alat dan Program, serta Hasil Alat.

4.1 Analisis Sistem

Berdasarkan hasil wawancara dan pengkajian yang dilakukan pada saat Kerja Praktik Program Kemitraan Masyarakat, terdapat beberapa masalah pada proses pembelajaran di TK Wijaya, yaitu:

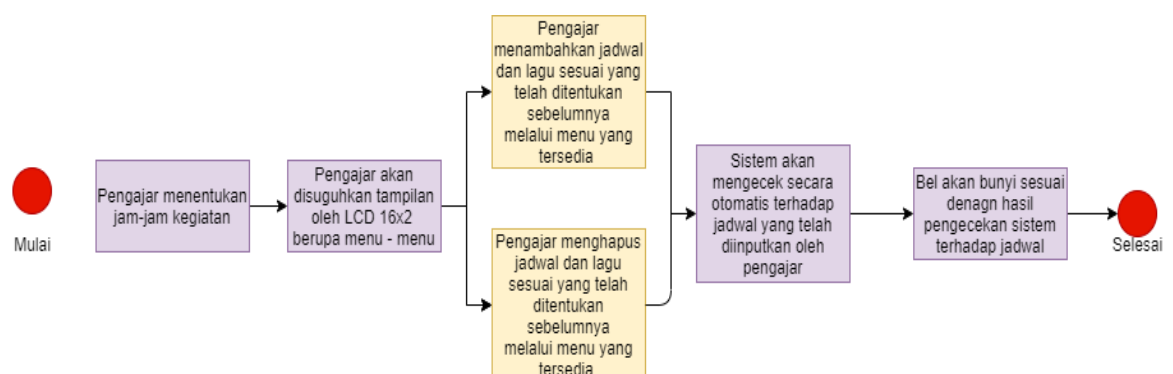
1. Tidak ada pengingat waktu, sehingga pengajar terkadang lupa untuk memberhentikan pembelajaran ketika jadwal sudah berganti.
2. Untuk mengatur para murid di TK Wijaya dibutuhkan suara lebih lantang dikarenakan rentang umur mereka (murid TK Wijaya) masih anak-anak sehingga masih suka bermain dan bercanda.
3. Para pengajar tidak hanya membutuhkan sistem alarm yang bisa berjalan secara otomatis, tetapi juga memiliki bunyi bel yang dapat mengenalkan para murid di TK Wijaya terhadap lagu-lagu yang mengedukasi.

Atas dasar tersebut dibuatlah sistem bel sekolah yang berjalan secara otomatis dalam hal pengecekan jadwal dan membunyikan bel serta pengaturan jadwal yang fleksibel, ditunjang dengan lagu yang sesuai dijadikan pengingat waktu kegiatan bagi anak-anak ditingkat TK (Taman Kanak-kanak). Dengan memasukan jadwal

dan lagu bel yang diinginkan melalui tombol – tombol pada alat Bel Sekolah Otomatis dengan panduan berupa tampilan yang tersedia pada LCD 16x2, maka para pengajar tidak perlu lagi memperhatikan jam dinding maupun media waktu lainnya untuk mengatur jadwal kegiatan di TK Wijaya. Untuk membangun sistem diatas dibutuhkan program yang nantinya diupload kedalam Arduino Uno, hal ini dibutuhkan karena semua sistem diatur oleh Mikrokontroler ini. Mikrokontroler nantinya akan mengatur semua sistem meliputi bagaimana cara mengolah data yang dihasilkan oleh RTC DS3201, menampilkan data pada LCD 16x2, menampilkan tampilan menu yang mudah dipahami, memanipulasi frekuensi buzzer untuk menghasilkan ringtone lagu, meyimpan data jadwal dan lagu kedalam memori EEPROM, menghapus data jadwal dan lagu kedalam memori EEPROM. Maka dari itu terciptalah sebuah sistem bel sekolah yang berjalan secara otomatis dilengkapi bel berupa ringtone lagu yang mendukung para murid di TK Wijaya.

4.2 Diagram Alur Proses Sistem Kendali Bel Sekolah

Adapun berdasarkan Analisis Sistem dan Permasalahan yang ada, terbentuklah Diagram Alur ini, untuk menggambarkan Alur Proses Sistem Kendali Bel Sekolah sebagai jawaban dari permasalahan yang ada.



Gambar 4.1 Diagram alur Sistem Kendali Bel Sekolah

4.3 Alur Kerja Sistem Kendali Bel Otomatis

1. Konsep dasar dari alat ini bel akan menyala ketika kondisi jadwal yang telah diinputkan terpenuhi.
2. Untuk menginputkan jadwal melalui menu - menu yang ditampilkan LCD 16x2.
3. Alur untuk manambah jadwal beserta lagu sebagai tanda bahwa kondisi jadwal telah terpenuhi melalui menu *Alarm* → *Tambah Jadwal*
4. Alur untuk menghapus jadwal beserta lagu apabila tidak diperlukan melalui menu *Alarm* → *Hapus Jadwal*
5. Setelah user menambahkan atau menghapus jadwal beserta lagu maka data akan disimpan kedalam memori mikrokontroler yang bernama EEPROM.
6. Jadwal yang telah diinputkan user dapat dilihat melalui menu *Alarm* → *Lihat Jadwal*.
7. Karena kelebihan EEPROM yang telah disebutkan pada BAB 3 maka data jadwal dan lagu yang telah diinputkan oleh user tidak akan hilang meskipun mikrokontroler kehilangan daya.

4.4 Spesifikasi Alat

Berdasarkan perancangan sistem diatas, maka untuk menjawab permasalahan tersebut, dirancanglah sebuah alat PERAPITU (Perangkat Alarm Peningat Waktu) yang berjalan sesuai dengan diagram alur Gambar 4.1.

4.4.1 PERAPITU

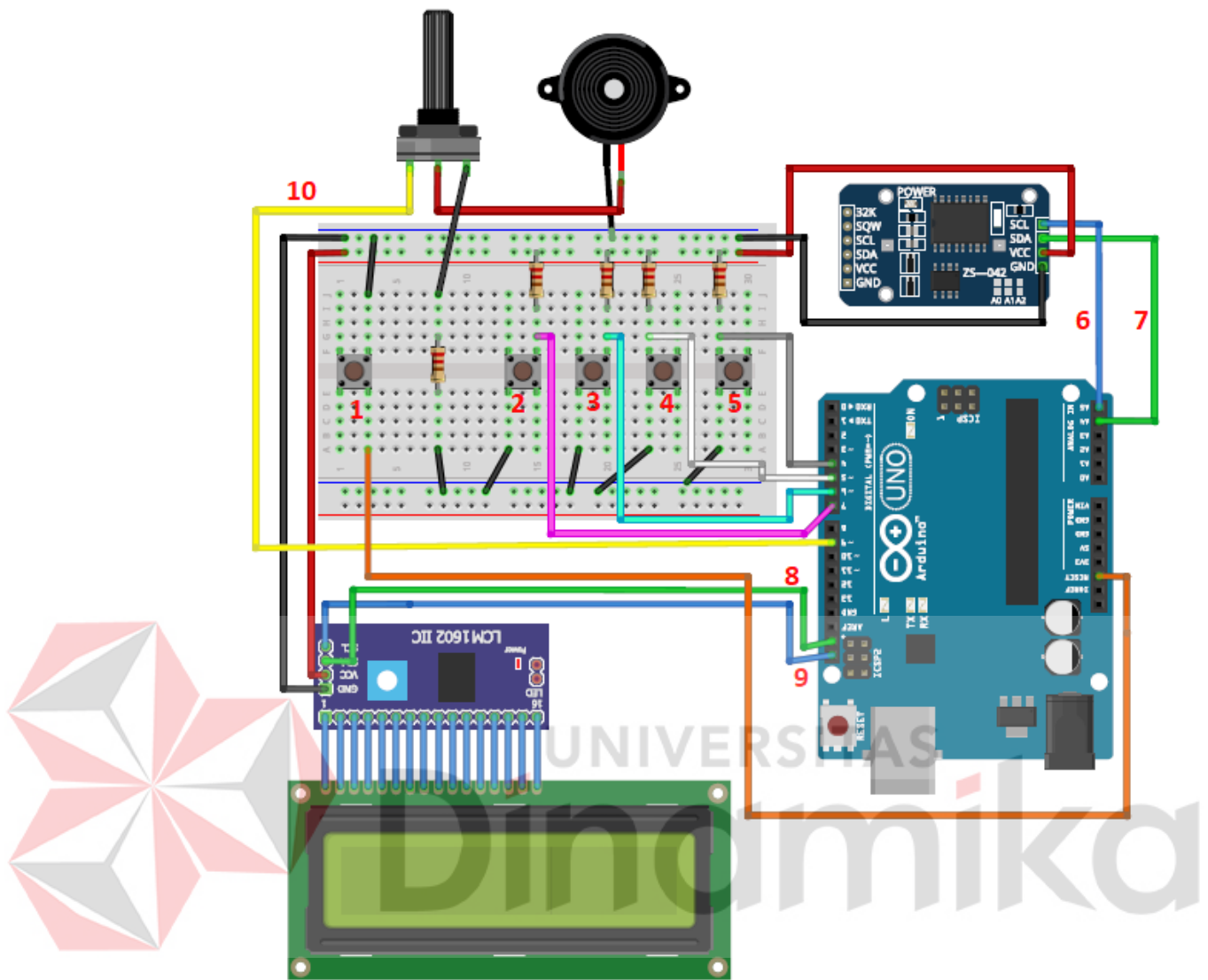
PERAPITU dirancang untuk menjadi bel sekolah otomatis di TK WIJAYA serta dilengkapi dengan ringtone lagu anak-anak yang mengedukasi, maka dari itu alat ini menggunakan spesifikasi seperti pada tabel dibawah ini.

Tabel 4.1 Spesifikasi alat PERAPITU

Spesifikasi	Keterangan
Mikrokontroler	Arduino Uno
Data Waktu	Modul RTC DS3231
Pengatur Suara Buzzer	Potensiometer
Antarmuka Pengguna	LCD Karakter 16 x 2 + i2C
Tombol	Push On
Catu Daya	Adaptor 12V 1A

4.5 Skematik Alat

Adapun skematik alat yang dibuat berdasarkan spesifikasi alat diatas, maka skematik rangkaian listrik PERAPITU dapat dilihat pada Gambar 4.2.

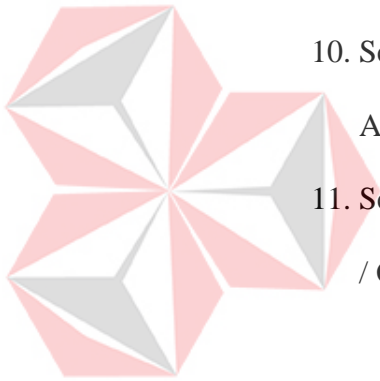


Gambar 4.2 Rangkaian listrik PERAPITU

Pada skematik rangkaian listrik Alat PERAPITU seperti gambar 4.2 setiap komponen terhubung dengan pin Arduino Uno, berikut pinout dari skematik rangkaian Alat PERAPITU :

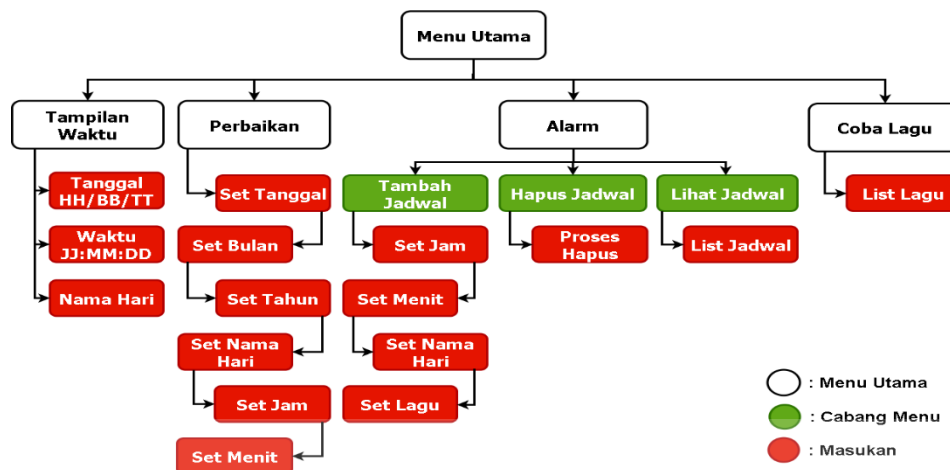
1. Button Reset terhubung dengan pin reset arduino.
2. Button Proceed / Ok terhubung dengan pin digital 7 arduino dengan rangkaian *Pull-up*.

3. Button Back terhubung dengan pin digital 6 arduino dengan rangkaian *Pull-up*.
4. Button Atas terhubung dengan pin digital 5 arduino dengan rangkaian *Pull-up*.
5. Button Bawah terhubung dengan pin digital 4 arduino dengan rangkaian *Pull-up*.
6. Port SCL RTC DS3231 terhubung dengan pin analog A5 Arduino.
7. Port SDA RTC DS3231 terhubung dengan pin analog A4 Arduino.
8. Pin SDA I2C Lcd 16 x 2 terhubung dengan pin analog A4 Arduino.
9. Pin SCL I2C Lcd 16 x 2 terhubung dengan pin analog A5 Arduino.
10. Setiap port dengan kabel berwarna merah terhubung dengan pin 5v
Arduino
11. Setiap port dengan kabel berwarna hitam terhubung dengan pin GND
/ Ground Arduino



4.6 Skematik Tampilan Menu

Adapun skematik tampilan menu pada sistem yang dibuat pada alat PERAPITU berdasarkan kebutuhan dari para pengajar di TK WIJAYA dapat dilihat pada Gambar 4.3.



Gambar 4.3 Skematik tampilan menu pada sistem PERAPITU

Skema tampilan menu pada Gambar 4.3 merupakan urutan bagaimana pengguna mengoperasikan PERAPITU. Pengguna dapat mengakses fitur-fitur yang disediakan oleh PERAPITU dengan panduan menu yang ditampilkan oleh LCD 16x2. Berikut urutan untuk mengakses fitur – fitur PERAPITU melalui tampilan menu.

- Untuk melihat waktu saat ini ke menu utama **Tampilan Waktu**
- Untuk memperbaiki waktu pada menu utama **Tampilan Waktu**, ke menu utama **Perbaikan**.
- Untuk menambah jadwal dan lagu alarm ke menu utama **Alarm** → **Tambah Jadwal**
- Untuk menghapus jadwal dan lagu alarm ke menu utama **Alarm** → **Hapus Jadwal**

- Untuk melihat jadwal dan lagu alarm yang sudah diinputkan oleh pengguna ke menu utama **Alarm → Lihat Jadwal**
- Untuk mengetahui bagaimana bunyi dari lagu-lagu / ringtone yang tersedia pada alat PERAPITU ke menu utama **Coba Lagu**

4.7 Penjelasan Program

Adapun Penjelasan program yang meliputi seluruh sistem pada alat PERAPITU, terdiri atas beberapa inisialisasi dan function yaitu :

4.7.1 Inisialisasi

Program inisialisasi untuk mendeklarasikan library, variabel, tipe data yang akan digunakan.

```
#include <DS3231.h>
#include <EEPROM.h>
#include <Wire.h>/////library i2c
#include <LiquidCrystal_I2C.h>/////library LCD
#include "lagu.h"
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3,
POSITIVE);
DS3231 rtc(SDA, SCL);
Time waktu;
String HARI [7] =
{"Senin","Selasa","Rabu","Kamis","Jumat","Sabtu","Ming
gu"};
String BULAN [12] =
{"Januari","Februari","Maret","April","Mei","Juni","Ju
li","Agustus","September","Oktober","November","Desemb
er"};
String LAGU [12] = {"Kasih Ibu","Anak Gembala","Potong
Bebek","Cicak di Dinding","Balonku","Bunyi
Hujan","Burung Kakatua","Tiup
Lilin","Sayonara","Guruku Tersayang","Satu Dua","Mari
Makan"};
byte up=7, down=6, back=5, utama=4;
int Buzzer = 9;
int jam,menit,Tanggal,Bulan,Hari,Lagu;
int status_jam,status_menit,status_hari,status_lagu;
int Tahun = 20;
int hari ;
```

```

int menu =1;
int submenu = 0;
int menualarm=1;
int menu_jadwalalarm=1;
int menu_hapusalarm=1;
int mode;
int setjam;
int setmenit;
int daflagu;
int cnt_hapusjadwal;
int cnt_jadwal;
int tambah_hapus;
int jm,mnt,hr;
int status_simpan;
int jmhapus,mnthapus,hrhapus,lguhapus;
int nilai;
int b,bb,LL;
int lagu;
int data_Jam;
int data_Menit;
int data_Detik;
int data_Tanggal;
int data_Bulan;
int data_Tahun;
int data_Hari;

```

Inisialisasi diatas merupakan sekumpulan library dan variabel yang bersifat global, sehingga dapat digunakan oleh seluruh *function* yang ada didalam program.

4.7.2 void pinmode ()

Function void pinmode () digunakan untuk mengatur mode pada pin Arduino.

```

void pinmode() {
  pinMode(A0,INPUT);
  pinMode(up, INPUT_PULLUP);
  pinMode(down, INPUT_PULLUP);
  pinMode(back, INPUT_PULLUP);
}

```

```
pinMode(utama, INPUT_PULLUP);
}
```

Function dalam void pinmode merupakan untuk tombol operasi pada alat PERAPITU. Untuk function diatas pin analog A0 pada Arduino diset sebagai Input secara langsung dengan menyebutkan nama asli dari pin analog arduino yaitu A0. Sedangkan pada pinMode lainnya diset menggunakan variabel global berikut.

```
byte up=7, down=6, back=5, utama=4;
```

4.7.3 void setup()

Adapun function void setup() akan dijalankan oleh program hanya sekali setelah mikrokontroler mendapatkan daya.

```
void setup() {
  Serial.begin(9600);
  lcd.begin(16,2);
  rtc.begin();
  pinMode(Buzzer, OUTPUT);
  booting();
  updatemenu();
}
```

4.7.4 void getwaktunormal ()

Adapun function void getwaktunormal () bekerja sebagai pengambil data waktu sekarang yang dihasilkan oleh RTC DS3231.

```
void getwaktunormal()
{
  waktu=rtc.getTime();
  data_Jam = waktu.hour;
  data_Menit = waktu.min;
```

```

data_Detik = waktu.sec;
data_Tanggal = waktu.date;
data_Bulan = waktu.mon;
data_Tahun = waktu.year;
data_Hari = waktu.dow;
}

```

Variabel “waktu” mengakses data RTC DS3231 dengan *syntax* “rtc.getTime()” sehingga variabel lainnya dapat mengakses data jam, menit, detik, tanggal, bulan tahun, dan hari dari RTCDS3231 melalui variabel “waktu”.

4.7.5 void buzzerr ()

Adapun *function* void buzzerr () digunakan untuk pemanggilan pengecekan terhadap jadwal yang sudah diinputkan oleh user.

```

void buzzerr() {
    for (int y=1; y<=EEPROM.read(730);y++)
    {
        jm = EEPROM.read(0+y);
        mnt = EEPROM.read(80+y);
        hr = EEPROM.read(160+y);
        if(data_Hari-1 == hr && data_Jam == jm &&
        data_Menit == mnt)
        {
            LL= EEPROM.read(240+y);
            if(LL == 0)
            {
                kasihibu();
            }
            if(LL == 1)
            {
                anakgembala();
            }
            if(LL == 2)
            {
                potongbebek();
            }
            if(LL == 3)

```



```

        {
            cicakdidinding();
        }
        if(LL == 4)
        {
            balonku();
        }
        if(LL == 5)
        {
            bunyihujan();
        }
        if(LL == 6)
        {
            burungkakaktua();
        }
        if(LL == 7)
        {
            tiuplilin();
        }
        if(LL == 8)
        {
            sayonara();
        }
        if(LL == 9)
        {
            gurukutersayang();
        }
        if(LL == 10)
        {
            satuduatigaempat();
        }
        if(LL == 11)
        {
            makanjanganbersuara();
        }
    }
    if(data_Hari-1 != hr || data_Jam != jm ||
data_Menit != mnt)
    {
        digitalWrite(Buzzer,LOW);
    }
}

```

Ketika kondisi jadwal terpenuhi terhadap waktu sekarang maka bel akan berbunyi sesuai dengan lagu / ringtone yang telah diinputkan user. Tetapi jika kondisi jadwal belum terpenuhi terhadap waktu sekarang maka program akan melakukan pengecekan lagi.

4.7.6 void loop ()

Adapun *function* void loop() sebagai program utama digunakan untuk menjalankan program yang ada didalam *function* secara terus menerus sampai mikrokontroler tidak mendapatkan daya. Function ini terdiri dari beberapa percabangan untuk memanggil *function* lainnya dan mengecek kondisi inputan nilai tombol untuk mengubah tampilan menu pada LCD 16x2.

Berikut beberapa function yang dipanggil oleh void loop () secara terus menerus sampai mikrokontroler tidak mendapatkan daya.

```

//////////////////////////////////Buzzer//////////////////////////////////
getwaktunormal();
buzzerr();
//////////////////////////////////

```

Sedangkan program untuk mengecek inputan tombol dan merubah tampilan menu pada LCD 16x2 adalah sebagai berikut. Perubahan tampilan menu berdasarkan kondisi *statement* tombol apakah *LOW* atau *HIGH* dan perubahan nilai dari variabel “submenu”.

```

//////////////////////////////////Maintenance//////////////////////////////////

    if(digitalRead(utama) == LOW && menu == 2 && submenu
    == 0)
    {

```

```

submenu = 2;
Tanggal = 0;
Bulan = 0;
Tahun = 20;
Hari = 0;
jam = 0;
menit = 0;
bb = 0;
lcd.clear();
aturwaktu();
//updatemenu();
while(digitalRead(utama) == LOW);
}
if (submenu == 2)
{
    aturwaktu();
}
if(digitalRead(back) == LOW && submenu == 2)
{
    submenu = 0;
    lcd.clear();
    updatemenu();
    while(digitalRead(back) == LOW);
}
if(digitalRead(utama) == LOW && submenu == 2)
{
    bb = bb+1;
    if (bb == 2 || bb == 4)
    {
        lcd.clear();
    }
    aturwaktu();
    while(digitalRead(utama) == LOW);
}
////////////////////////////////////
////////////////////////////////////

////////////////////////////////////Alarm////////////////////////////////////
////////////////////////////////////
    if(digitalRead(utama) == LOW && menu == 3 && submenu
== 0)
    {
        submenu =1;
        lcd.clear();
        menualarm1();
        delay(100);
        while(digitalRead(utama) == LOW);
    }

```

```

}

if (submenu == 1)
{
    menualarm1();
}

if(digitalRead(utama) == LOW && menualarm == 1 &&
submenu == 1)
{
    submenu = 3;
    jam = 0;
    menit = 0;
    hari = 0;
    lagu = 0;
    b = 0;
    lcd.clear();
    aturalarm();
    delay(100);
    while(digitalRead(utama) == LOW);
}

if(submenu == 3)
{
    aturalarm();
}

if(digitalRead(back) == LOW && submenu == 3)
{
    submenu = 1;
    lcd.clear();
    menualarm1();
    while(digitalRead(back) == LOW);
}

if(digitalRead(utama) == LOW && submenu == 3)
{
    b = b+1;
    if (b == 2 || b == 3)
    {
        lcd.clear();
    }
    aturalarm();
    while(digitalRead(utama) == LOW);
}
////////////////////

```

4.7.7 void daftarlagu ()

Adapun *function* void daftarlagu berfungsi untuk menampilkan dan memainkan daftar lagu yang tersedia pada alat PERAPITU ketika dipanggil oleh *function* void loop().

```
void daftarlagu(){
  if(daflagu < 1)
  {
    daflagu = 1;
    lcd.setCursor(0,0);
    lcd.print(daflagu);
    lcd.print(".");
    lcd.setCursor(2,0);
    lcd.print("Play >>");
    lcd.setCursor(0,1);
    lcd.print(LAGU[daflagu - 1]);
    while(digitalRead(utama) == LOW);
  }
  if(daflagu >=13)
  {
    daflagu = 12;
  }
  if(daflagu < 10)
  {
    lcd.setCursor(0,0);
    lcd.print(daflagu);
    lcd.print(".");
    lcd.setCursor(2,0);
    lcd.print("Play >>");
    lcd.setCursor(0,1);
    lcd.print(LAGU[daflagu - 1]);
  }
  if(daflagu >= 10)
  {
    lcd.setCursor(0,0);
    lcd.print(daflagu);
    lcd.print(".");
    lcd.setCursor(3,0);
    lcd.print("Play >>");
    lcd.setCursor(0,1);
    lcd.print(LAGU[daflagu - 1]);
  }
}
```

```

    }
    if(digitalRead(utama)==LOW    &&    submenu==7    &&
daflagu == 1)
    {
        kasihibu();
    }
    if(digitalRead(utama)==LOW    &&    submenu==7    &&
daflagu == 2)
    {
        anakgembala();
    }
    if(digitalRead(utama)==LOW    &&    submenu==7    &&
daflagu == 3)
    {
        potongbebek();
    }
    if(digitalRead(utama)==LOW    &&    submenu==7    &&
daflagu == 4)
    {
        cicakdidinding();
    }
    if(digitalRead(utama)==LOW    &&    submenu==7    &&
daflagu == 5)
    {
        balonku();
    }
    if(digitalRead(utama)==LOW    &&    submenu==7    &&
daflagu == 6)
    {
        bunyihujan();
    }
    if(digitalRead(utama)==LOW    &&    submenu==7    &&
daflagu == 7)
    {
        burungkakaktua();
    }
    if(digitalRead(utama)==LOW    &&    submenu==7    &&
daflagu == 8)
    {
        tiuplilin();
    }
    if(digitalRead(utama)==LOW    &&    submenu==7    &&
daflagu == 9)
    {
        sayonara();
    }
}

```

```

    if(digitalRead(utama)==LOW    &&    submenu==7    &&
daflagu == 10)
    {
        gurukutersayang();
    }
    if(digitalRead(utama)==LOW    &&    submenu==7    &&
daflagu == 11)
    {
        satuduatigaempat();
    }
    if(digitalRead(utama)==LOW    &&    submenu==7    &&
daflagu == 12)
    {
        makanjanganbersuara();
    }
}

```

4.7.8 void updatemenu ()

Adapun *function* void updatemenu () menampilkan tampilan menu utama dari sistem pada alat PERAPITU ketika dipanggil oleh *function* void loop(). Tampilan utama terdiri dari **Tampilan waktu** sekarang, **Perbaikan, Alarm,** dan **Coba Lagu.**

```

void updatemenu(){
    switch(menu) {
        case 0:
            menu =1;
            waktunormal();
            break;
        case 1:
            //lcd.clear();
            waktunormal();
            break;
        case 2:
            //lcd.clear();
            lcd.print(">Perbaikan");
            lcd.setCursor(0,1);
            lcd.print(" Alarm");
            break;
        case 3:
            //lcd.clear();
            lcd.print(" Perbaikan");
    }
}

```

```

        lcd.setCursor(0,1);
        lcd.print(">Alarm");
        break;
    case 4:
        lcd.setCursor(0,0);
        lcd.print(">Daftar Lagu");
        break;
    case 5:
        menu = 4;
        lcd.setCursor(0,0);
        lcd.print(">Daftar Lagu");
        break;
    }
}

```

Penggunaan struktur program *switchcase* dalam *function* void *daftarlagu()* tidak memiliki tujuan tertentu, melainkan hanya variasi. Program ini bisa juga dieksekusi dengan struktur *if else*.

4.7.9 void menualarm1 ()

Adapun *function* void *menualarm ()* menampilkan menu-menu setelah masuk menu utama **Alarm**. Struktur program pada *function* ini juga menggunakan *switch case*.

```

void menualarm1()
{
    switch(menualarm) {
        case 0:
            menualarm=1;
            break;
        case 1:
            //lcd.clear();
            lcd.setCursor(0,0);
            lcd.print(">Tambah Jadwal");
            lcd.setCursor(0,1);
            lcd.print(" Hapus Jadwal");
            break;
        case 2:

```

```

        //lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(" Tambah Jadwal");
        lcd.setCursor(0,1);
        lcd.print(">Hapus Jadwal");
        break;
    case 3:
        //lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(">Jadwal Alarm");
        break;
    case 4:
        //lcd.clear();
        menualarm =3;
        break;
    }
}

```

4.7.10 void digit ()

Adapun *function* void digit () untuk menambahkan karakter “0” didepan angka data jam, menit, dan detik dari RTC yang hanya bernilai satu digit.

Function ini dibuat agar tampilan waktu RTC di LCD sama seperti jam digital pada umumnya.

```

void digit (int angka){
    if (angka < 10){
        lcd.print("0");
        lcd.print(angka);
    }
    else
        lcd.print(angka);
}

```

4.7.11 Penyimpanan data jadwal ke EEPROM

Pada sistem kendali bel otomatis diperlukan cara menyimpan data yang telah diinputkan user menjadi sebuah jadwal. Adapun cara menyimpan data jadwal pada sistem ini dengan cara memanggil *function* tertentu kemudian menyimpannya kedalam EEPROM. Berikut beberapa *function* tertentu yang dimaksud. Berikut cara menyimpan data jadwal kedalam EEPROM :

```
void simpanjam(int JAM)
{
    for (int i=1;i<=80;i++)
    {
        status_simpan = EEPROM.read(400+i);
        if(status_simpan == NULL)
        {
            EEPROM.write (i,JAM );
            EEPROM.write (400+i,1);
            break;
        }
    }
}

void simpanmenit(int MENIT)
{
    for (int j=81;j<=160;j++)
    {
        status_simpan = EEPROM.read(400+j);
        if(status_simpan == NULL)
        {
            //jm_mnt_hr[j] = MENIT;
            //indexjam_mnt_hr[j] = j;
            EEPROM.write (j, MENIT);
            EEPROM.write (400+j,1);

            break;
        }
    }
}

void simpanhari(int HARI)
{
    for (int k=161;k<=240;k++)
    {
        status_simpan = EEPROM.read(400+k);
```

```

        if (status_simpan == NULL)
        {
            EEPROM.write (k, HARI);
            EEPROM.write (400+k,1);
            break;
        }
    }
}
void simpanlagu(int lgu)
{
    for (int L=241;L<=320;L++)
    {
        status_simpan = EEPROM.read(400+L);
        if (status_simpan == NULL)
        {
            EEPROM.write (L, lgu);
            EEPROM.write (400+L,1);
            break;
        }
    }
}

```

Sebagai contoh, jika variabel “status_simpan” bernilai Null maka nilai “int JAM” akan disimpan oleh EEPROM dengan alamat nilai dari variabel “i”. Kemudian untuk menandai bahwa pada alamat EEPROM tersebut telah terpakai oleh data jam, maka dibuatlah status penanda seperti saat akan menyimpan data jam yaitu dengan cara menyimpan nilai 1 pada alamat EEPROM 400 ditambah variabel “i”.

```
EEPROM.write (400+i,1);
```

Adapun pada *function* void simpanjam () terdapat parameter “int JAM” sebagai wadah nilai dari *function* yang memanggilnya yaitu void aturalarm.

```

void aturalarm()
{
    if(b == 0)
    {

```

```

    lcd.setCursor(0,0);
    lcd.print("Set Jam:");
    jamalarm();
}

if(b == 1)
{
    lcd.setCursor(0,1);
    lcd.print("Set Menit:");
    menitalarm();
}

if(b == 2)
{
    lcd.setCursor(0,0);
    lcd.print("Set Hari:");
    harialarm();
}

if(b == 3)
{
    lcd.setCursor(0,0);
    lcd.print("Set Lagu:");
    lagualarm();
}

if(b == 4)
{
    simpanjam(jam);
    simpanmenit(menit);
    simpanhari(hari);
    simpanlagu(lagu);
    apply();
    cnt_jadwal = EEPROM.read (730);
    EEPROM.write (730,cnt_jadwal +1);
    submenu = 1;
    menualarm1();
}
}

```

4.7.12 Menghapus data alarm pada EEPROM

Adapun cara menghapus data jadwal pada sistem ini dengan cara memanggil *function* void `hapusjadwal ()`. Pada *function* ini terdapat algoritma

agar data jadwal yang telah diinputkan tetap urut didalam sistem maupun tampilan meskipun terjadi penghapusan.

```
void hapusjadwal(int parm_hapus)
{
    //lcd.print ("Tekan tombol")
    EEPROM.write(0+parm_hapus,0);
    EEPROM.write(80+parm_hapus,0);
    EEPROM.write(160+parm_hapus,0);
    EEPROM.write(240+parm_hapus,0);
    cnt_hapusjadwal = EEPROM.read(730);
    EEPROM.write(400+cnt_hapusjadwal,0);
    EEPROM.write(480+cnt_hapusjadwal,0);
    EEPROM.write(560+cnt_hapusjadwal,0);
    EEPROM.write(640+cnt_hapusjadwal,0);
    for (parm_hapus; parm_hapus <= cnt_hapusjadwal;
    parm_hapus++)
    {
        jmhapus = EEPROM.read (0+parm_hapus+1);
        mnthapus = EEPROM.read (80+parm_hapus+1);
        hrhapus = EEPROM.read (160+parm_hapus+1);
        lguhapus = EEPROM.read (240+parm_hapus+1);

        status_jam      =      EEPROM.read
(400+cnt_hapusjadwal+1);
        status_menit    =      EEPROM.read
(480+cnt_hapusjadwal+1);
        status_hari      =      EEPROM.read
(560+cnt_hapusjadwal+1);
        status_lagu      =      EEPROM.read
(640+cnt_hapusjadwal+1);

        if(status_jam == 0 &&status_menit == 0 &&
status_hari == 0 && status_lagu == 0)
        {
            EEPROM.write (0+parm_hapus,jmhapus);
            EEPROM.write (80+parm_hapus,mnthapus);
            EEPROM.write (160+parm_hapus,hrhapus);
            EEPROM.write (240+parm_hapus,lguhapus);
        }

        if(status_jam == 1 &&status_menit == 1 &&
status_hari == 1 && status_lagu == 1)
        {
            EEPROM.write (0+parm_hapus,jmhapus);
```

```

EEPROM.write (80+parm_hapus,mnthapus);
EEPROM.write (160+parm_hapus,hrhapus);
EEPROM.write (240+parm_hapus,lguhapus);

EEPROM.write (400+parm_hapus,status_jam);
EEPROM.write (480+parm_hapus,status_menit);
EEPROM.write (560+parm_hapus,status_hari);
EEPROM.write (640+parm_hapus,status_lagu);
    }
}
EEPROM.write (730,cnt_hapusjadwal-1);
apply();
}

```

Berikut contoh implementasi dari algoritma diatas pada Tabel 4.2 ketika terjadi penghapusan data jadwal. Terdapat 3 jadwal :

Tabel 4.2 Urutan data pada sistem dan tampilan

Urutan	Data	Status jadwal
1	Jam =06, menit=30	1
2	Jam =07, menit=00	1
3	Jam =09, menit=45	1

Setelah terjadi penghapusan pada data jadwal urutan ke-2 maka sistem dan tampilan akan menjadi seperti Tabel 4.3.

Tabel 4.3 Sistem dan tampilan setelah terjadi penghapusan

Urutan	Data	Status jadwal
1	Jam =06, menit=30	1
2	Jam =09, menit=45	1

3	Tidak ada jadwal	0
---	------------------	---

4.7.13 Program lagu

Adapun program lagu dibuat sebagai dasar dalam hal memanipulasi outputan dari buzzer sehingga menjadi lagu atau ringtone. Program lagu terdiri dari deklarasi dan beberapa *function*.

A. Deklarasi

```
#define B4 494.00//5not
#define C4 262.00//5not
#define D4 294.00//5not
#define E4 330.00//5not
#define F4 349.00//5not
#define G4 392.00//5not
#define B4 494.00//5not
#define D8 4699.00//2not
#define F6 1397.00//4not
#define G6 1568.00//5not
#define LA6 1760.00//6not
#define B6 1976.00//7not
#define C7 2093.00//1
#define Db7 2217.46//2not
#define D7 2349.32//2
#define Eb7 2489.02///3not
#define E7 2637.02//3
#define F7 2793.83//4
#define Gb7 2959.96///5not
#define G7 3135.96//5
#define Ab7 3322.44
#define LA7 3520.01///6
#define Bb7 3729.31
#define B7 3951.07///7
#define C8 4186.01//1not
#define GS6 1661.00//5 coret
#define CS6 1109.00//1 coret
#define CS8 4435.00//1not coret
#define FS6 1480.00//4 coret
#define AS6 1865.00// 6 coret
```

Deklarasi diatas merupakan manipulasi frekuensi outputan buzzer menjadi note *solmisasi*.

B. void PlayTone ()

```
void PlayTone(unsigned int note, unsigned long
delaynote, unsigned int dutycycle)
{
    unsigned long dutyon = delaynote * dutycycle / 100;
    unsigned long dutyoff = delaynote - dutyon;
    tone(9, note); delay(dutyon);
    noTone(9); delay(dutyoff);
}
```

Function void PlayTone () memiliki tiga parameter. Penjelasan dari ketiga parameter *function* void PlayTone () dapat dilihat pada Tabel 4.4.

Tabel 4.4 Penjelasan parameter function void PlayTone ()

Parameter	Keterangan
unsigned int note	Sebagai wadah daripada variabel note <i>solmisasi</i> ketika terjadi pemanggilan function void PlayTone ()
unsigned long delaynote	Sebagai wadah daripada variabel “mydelay” . Parameter ini berfungsi sebagai pengatur lamanya note <i>solmisasi</i> dibunyikan ketika terjadi pemanggilan function void PlayTone ()
unsigned int dutycycle	Sebagai wadah daripada variabel “myduty” ketika terjadi pemanggilan <i>function</i> void PlayTone () . Parameter ini berfungsi sebagai pengatur lamanya note <i>solmisasi</i> dibunyikan terhadap satuan periode variabel “delaynote”

C. Fuction lagu

Adapun *function* lagu terdiri dari deklarasi variabel “mydelay”, “myduty” dan pemanggilan *function* void PlayTone(). Terdapat 12 *function* lagu karena memang daftar lagu yang dimiliki oleh sistem berjumlah 12 judul yaitu :

1. Balonku
2. Kasih Ibu
3. Anak Gembala
4. Cicak di Dinding
5. Bunyi Hujan
6. Burung Kakatua
7. Tiup lilinnya
8. Potong Bebek
9. Sayonara
10. Satu Dua Tiga Empat
11. Guruku Tersayang
12. Makan Jangan Bersuara

Karena pola dari semua *function* lagu diatas sama, maka berikut cuplikan program *function* dari salah satu judul lagu diatas yaitu void cicakdidinding ().

```
void cicakdidinding()
{
    ///Cicak Di Dinding
    unsigned long mydelay = 500;
    unsigned int myduty = 80;
    PlayTone(G7, mydelay, myduty);
    PlayTone(E7, mydelay, myduty);
}
```

```

PlayTone(G7, mydelay, myduty);
PlayTone(E7, mydelay, myduty);
PlayTone(E7, mydelay, myduty);
PlayTone(F7, mydelay, myduty);
PlayTone(G7, mydelay*2, myduty);
PlayTone(F7, mydelay, myduty);
PlayTone(D7, mydelay, myduty);
PlayTone(F7, mydelay, myduty);
PlayTone(LA7, mydelay, myduty);
PlayTone(G7, mydelay, myduty);
PlayTone(F7, mydelay, myduty);
PlayTone(E7, mydelay*2, myduty);
PlayTone(LA7, mydelay, myduty);
PlayTone(F7, mydelay, myduty);
PlayTone(LA7, mydelay, myduty);
PlayTone(F7, mydelay, myduty);
PlayTone(LA7, mydelay, myduty);
PlayTone(B7, mydelay, myduty);
PlayTone(C8, mydelay*2, myduty);
delay(300);
PlayTone(C8, mydelay*2, myduty);
PlayTone(E7, mydelay, myduty);
PlayTone(G7, mydelay, myduty);
PlayTone(F7, mydelay, myduty);
PlayTone(D7, mydelay, myduty);
PlayTone(C7, mydelay*2, myduty);
}

```

4.8 Hasil Alat

Adapun hasil alat dari Sistem Kendali Bel Sekolah yang bernama PERAPITU, terbagi atas dua, yaitu : hasil kemasan dan hasil sistem.

4.8.1 Hasil Kemasan

Adapun hasil dari Alat PERAPITU pada Gambar 4.4 menggunakan kotak berbahan akrilik warna hitam berdimensi 12,5cm x 8,5cm x 5,5cm.



Gambar 4.4 Kemasan alat PERAPITU

4.8.2 Hasil Sistem

Adapun hasil sistem PERAPITU dapat dilihat dari tampilan LCD 16x2 saat pengguna mengoperasikan alat. Berikut tampilan – tampilan dari alat PERAPITU saat pengoperasian :

A. Tampilan awal PERAPITU



Gambar 4.5 Tampilan awal PERAPITU

Tampilan alat pada Gambar 4.5 menunjukkan waktu sekarang.

Tampilan ini merupakan menu utama dari sistem.

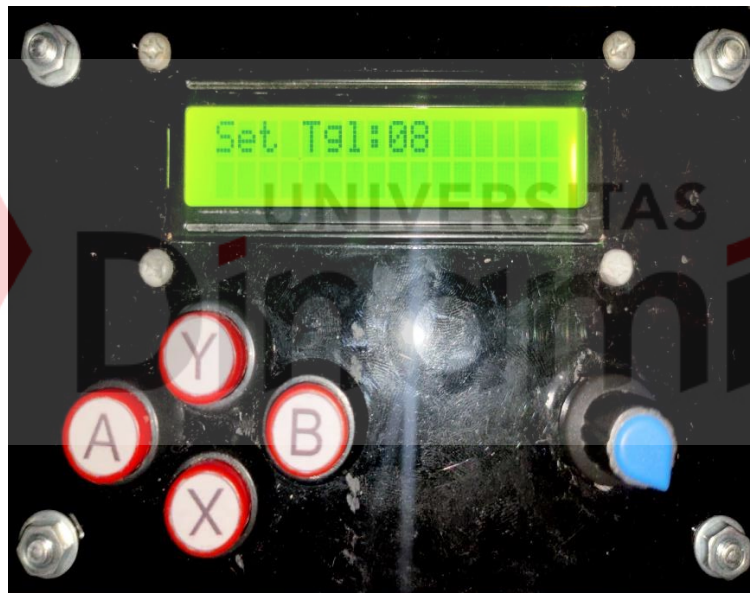
B. Tampilan menu Perbaikan

Pada tampilan menu Perbaikan terdapat langkah-langkah berupa gambar untuk memperbaiki tampilan menu utama pada Gambar 4.5.

Berikut gambar-gambar pada tampilan menu utama perbaikan



Gambar 4.6 Tampilan Perbaikan pada menu utama sistem



Gambar 4.7 Tampilan setelah masuk ke menu Perbaikan untuk set tanggal



Gambar 4.8 Tampilan setelah masuk ke menu Perbaikan untuk set bulan



Gambar 4.9 Tampilan setelah masuk ke menu Perbaikan untuk set tahun



Gambar 4.10 Tampilan setelah masuk ke menu Perbaikan untuk set hari



Gambar 4.11 Tampilan setelah masuk ke menu Perbaikan untuk set tanggal



Gambar 4.12 Tampilan setelah masuk ke menu Perbaikan untuk set menit

C. Tampilan menu Alarm

Pada tampilan menu Alarm terdapat langkah-langkah berupa gambar alat untuk menambah dan menghapus jadwal. Berikut gambar-gambar pada tampilan Tambah Jadwal.



Gambar 4.13 Tampilan menu Alarm pada menu utama



Gambar 4.14 Tampilan setelah masuk ke menu Alarm



Gambar 4.15 Tampilan setelah masuk ke menu Tambah Jadwal untuk set jam



Gambar 4.16 Tampilan setelah masuk ke menu Tambah Jadwal untuk set menit



Gambar 4.17 Tampilan setelah masuk ke menu Tambah Jadwal untuk set hari



Gambar 4.18 Tampilan setelah masuk ke menu Tambah Jadwal untuk set lagu

Selain menambah jadwal, sistem dari alat PERAPITU juga dapat menghapus jadwal yang telah diinputkan. Berikut tampilan-tampilan pada LCD untuk menghapus jadwal.



Gambar 4.19 Tampilan setelah masuk ke menu utama Alarm untuk menghapus jadwal



Gambar 4.20 Tampilan jadwal yang telah ditambahkan oleh pengguna



Gambar 4.21 Tampilan jadwal yang telah dihapus oleh sistem

BAB V

PENUTUP

Pada bab lima ini akan dibahas mengenai kesimpulan dan saran dari Sistem Kendali Sekolah Untuk Program Pengabdian Masyarakat.

5.1 Kesimpulan

Kesimpulan pada pembuatan Sistem Kendali Bel Sekolah berbasis *RTC DS3231* menggunakan *mikrokontroler Arduino Uno* adalah sistem tersebut menggunakan satu alat, yaitu PERAPITU (Perangkat Alarm Pengingat Waktu) , dimana alat ini telah diterima dan dipergunakan oleh pengajar di TK WIJAYA Penjaringan Sari Surabaya. Sistem tersebut telah menjawab kebutuhan dan permasalahan yang ada, yaitu sebagai berikut:

1. Sistem memudahkan pengajar di TK WIJAYA Penjaringan Sari Surabaya untuk menyusun jadwal kegiatan beserta lagu sebagai bel.
2. Sistem telah membantu pengajar dalam mengatur murid-murid ketika jam pergantian jadwal tanpa harus berteriak.
3. Sistem meminimalisir kelalaian pengajar untuk mengingatkan ketika jam kegiatan telah berganti.

5.2 Saran

Adapun saran dari penulis adalah sistem dapat dikembangkan lebih baik lagi, seperti penambahan fitur *IoT (Internet of Thing)* sehingga pengoperasian alat bisa dilakukan kapanpun dan dimanapun, serta penambahan dari segi antarmuka sistem

terhadap pengguna, serta rangkaian elektronika yang dibuat, serta *bug-bug* lain pada program yang mungkin tidak diketahui.



UNIVERSITAS
Dinamika

DAFTAR PUSTAKA

- Arduino. (2020, november 03). *tone()*. Retrieved from arduino.cc:
<https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/>
- Bhagman. (2017, january 14). *Tone Library*. Retrieved from github.com:
<https://github.com/bhagman/Tone#ugly-details>
- ELEKTRONIKA, L. (2017, maret 26). *CARA PROGRAM LCD KARAKTER 16x2 MENGGUNAKAN ARDUINO DAN SIMULASI PROTEUS*. Retrieved from
labelektronika.com: <http://www.labelektronika.com/2017/03/cara-program-lcd-karakter-16x2-Arduno-dan-Proteus.html>
- Faudin, A. (2017, july 23). *Mengenal, Apa itu Arduino ?* Retrieved from
nyebarilmu.com: <https://www.nyebarilmu.com/mengenal-apa-itu-arduino-uno/>
- Sitepu, J. (2018, Juli 5). *Tutorial Cara Menggunakan EEPROM Arduino Tanpa Library, Lengkap dengan Program nya*. Retrieved from mikroavr.com:
<https://mikroavr.com/program-eprom-arduino/>
- Suprianto. (2015, oktober 30). *PENGERTIAN PUSH BUTTON SWITCH (SAKLAR TOMBOL TEKAN)*. Retrieved from blog.unnes.ac.id:
<http://blog.unnes.ac.id/antosupri/pengertian-push-button-switch-saklar-tombol-tekan/>

```

    jmhapus = EEPROM.read (0+parm_hapus+1);
    mnthapus = EEPROM.read (80+parm_hapus+1);
    hrhapus = EEPROM.read (160+parm_hapus+1);
    lguhapus = EEPROM.read (240+parm_hapus+1);

    status_jam = EEPROM.read (400+cnt_hapusjadwal+1);
    status_menit = EEPROM.read (480+cnt_hapusjadwal+1);
    status_hari = EEPROM.read (560+cnt_hapusjadwal+1);
    status_lagu = EEPROM.read (640+cnt_hapusjadwal+1);

    if(status_jam == 0 &&status_menit == 0 && status_hari
    == 0 && status_lagu == 0)
    {
        EEPROM.write (0+parm_hapus,jmhapus);
        EEPROM.write (80+parm_hapus,mnthapus);
        EEPROM.write (160+parm_hapus,hrhapus);
        EEPROM.write (240+parm_hapus,lguhapus);
    }

    if(status_jam == 1 &&status_menit == 1 && status_hari
    == 1 && status_lagu == 1)
    {
        EEPROM.write (0+parm_hapus,jmhapus);
        EEPROM.write (80+parm_hapus,mnthapus);
        EEPROM.write (160+parm_hapus,hrhapus);
        EEPROM.write (240+parm_hapus,lguhapus);

        EEPROM.write (400+parm_hapus,status_jam);
        EEPROM.write (480+parm_hapus,status_menit);
        EEPROM.write (560+parm_hapus,status_hari);
        EEPROM.write (640+parm_hapus,status_lagu);
    }
}
EEPROM.write (730,cnt_hapusjadwal-1);
apply();
}

```